Semantic Web Road map

Summary

In [reference], Tim Berners-Lee outlines the high level architecture of the next generation WWW: the Semantic Web. He also discusses some specific issues such as proof, query, and trust.

In current web, web information is only designed for human beings and can not be understood by machines. However, the Semantic Web aims to express web information in a machine-readable way, enabling computers and humans to work collaboratively.

The high level architecture he proposes consists of a basic assertion model, a schema layer and a logic layer.

The assertion model employs Resource Description Framework (RDF) as a general model being able to map any prospective application. This model only consists of assertions and quotations, which lays a common framework for the Semantic Web. The schema layer introduces RDF schema to validate and constraint the way RDF assertions are used. A simple application of this layer is to convert a document in one RDF schema into another one. However, to keep concerns separated, the schema layer is designed to be limited and it does not use any logical expressions. The logic layer, on the other hand, aims to understand the semantic of a schema language in logical terms. This layer uses predicate logic to express inference rules and is powerful to be applied to any applications. However, it does not tell one where to find these rules. For instance, a program which finds a version 2 document wants to find some rules to convert it into a version 1 document. An alternative to adding pointers to these rules to each document is searching third part indices for connections between two schemas.

To avoid computational infeasibility, the Semantic web does not try to generate proofs but to validate proofs given constrained rules, for example, in access control to a web site. The proof here means a chain of assertions and reasoning rules with pointers to all the supporting material.

A query engine is required to query the Semantic Web to get assertions as answers to sorts of questions. A query language, which can be written in RDF logic, should express query types succinctly and certain constrained queries, as SQL does. A specific query engine can only search a finite level of depth in a specified subset of the Web.

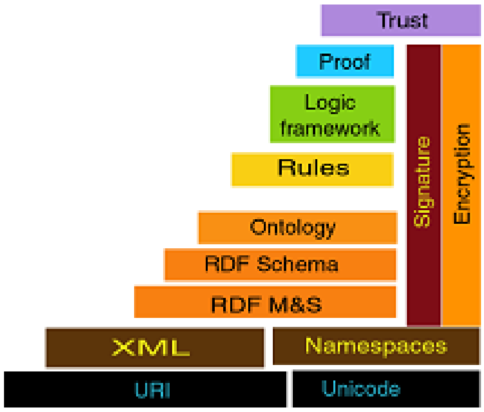
Taking trust into account, an extended logical model is required to include keys with which assertions have been signed. Proof validation and digital signature will work together to provide a signature verification system. Documents will be parsed not just

into trees of assertions, but into trees of assertions about who has signed what assertions.

Search engine in semantic web would index RDF objects rather than words. This can give probably correct answer without too many inappropriate answers as today's search engines do. However, a major problem is the significant explosion of the possibilities that appear in constructing correct answers. A solution is to combine a reasoning engine with a search engine. Besides, problems in real life may be solved in a few steps and may have constraints, which may not lead to a computational explosion. Further Research is demanded in addressing this problem.

Discussions

Actually, this paper describes the following diagram in general. The author defines the framework for Semantic Web and makes some suggestions for further research.



However, this paper does not provide too many details in understanding this diagram. Further readings, especially about rules, proof and security, are required. Besides, this paper has numerous grammar errors and misspelling words. I wonder why they are not fixed before being put online.

Some questions appear in my mind and some of them are answered by reading other papers or by my understanding.

(1) Can we annotate any page?
Don't count on that we can annotate the whole internet, which would result in computation infeasibility. But we can annotate domain-specific web pages.

(2) Annotate the content of the page or just the page?
URI can represent all of resources in the world, including web page, video, audio and a person.

(3) How to annotate a web page?
Who can create the ontology? How to verify them?

Since it is domain dependent, expertise can annotate a specific domain given well designed ontology tools. Digital signature and public key mechanisms can be used to authenticate an expert to enable access to ontology.

(4) Can we automatically annotate page?
Quite hard, but there is research on it.

(5) How to search the Semantic web ? vs. google?
Is there any hyperlink in semantic web?
How to browse a semantic web page?

The Semantic web makes computer understand the content of a web page. I think people can view the web in a similar way as before. But the web page contains additional annotations for machines to process. These annotations will be indexed by the semantic google in a centralized way or decentralized way. The semantic web can be regarded as a large database scaling as large as the Internet. Some semantic web browsers have been developed [2].

Recently, RDF Site Summary (RSS) are more and more popular. RSS is a lightweight multipurpose extensible metadata description and syndication format [3]. An RSS summary, at a minimum, is a document describing a "channel" consisting of URL-retrievable items. Each item consists of a title, link, and brief description. If you like checking news headlines, tracking eBay auctions, or keeping up with the latest blogs, an RSS reader [4] can keep you up to date when they have been updated. You can also custom the way you like to view them in terms of format, place, date and so on.

(6) How to query? What will the user interface be?
If you want to query a triple store, u have to know the ontology used for that store? Koz these terms will appear in your query. That means unlike google, u can not use any terms to query a repository. Right?

Since ontology is the knowledge shared within virtual organizations, it is better to know the ontology before querying. Besides, personalized query can be made. In other words, the system can select corresponding ontology according to the configuration a user makes. For instance, if a user is a chemist, then he provides this information to the system. When he requests a query, his query can be processed and reformed to RDQL with Chemistry ontology by the system. He does not need to know what the Chemistry ontology is.

(7) Trust
Obviously, we can not trust all the RDF statements received from the semantic web. Web of trust should be made [5]. You should know who you trust and how much you trust them. A solution is to use digital signature to ensure the triples were actually said

by whom they claimed to be said by. The verification of one's public key can be made through the Public Key Infrastructure (PKI). A Key Free way is proposed in [6].

(8) How to deal with inconsistent data?
This is an arguable question that people often ask. If a statement A contradict another statement B in the semantic web, then which one to believe? Or you can deduce anything?

Tim Berners-Lee discusses this point in [7]. He states that any real Semantic Web system will not work by believing anything it reads on the web but by checking the source of any trusted information (provenance?). The trust can be verified by digital signature as mentioned above. While checking the source means the system would searches a global index for documents giving an answer to a query. The system needs to know whether the given answer can be derived directly or indirectly from sources it has set up trust on. If two contradicting statements can be derived from the sources signed with the same key, then we can know that signed with the key, anything can be deduced. Then the key is broken and should not be trusted any more.

Besides, time-varying information can also result in contradiction. This can be solved by adding time stamp to statements. After the stamp expires, the statements can be trusted.

However, it seems that the first case makes an assumption that there should be a well-formed system, in which all the assertions are assumed to be correct and reflect the knowledge in the real world. But it is quite hard, isn't it? (There is not only answer to many questions in the real world.)

Anyway, further research will be conducted to address this issue and the semantic web has to tolerate many kinds of inconsistency.

(9) Computational feasibility?
A major problem of the Semantic Web is the significant explosion of the possibilities that need to be traced and reasoned to get answers. Since there are numerous RDF statements in triple stores, the number of possibilities may be astronomical, which makes computation inefficiently or even infeasibly.

[8] discusses this issue. The goal of the semantic web is to express real life. Many things in real life, real questions are not efficiently computable. There are two solutions to this: The classical solution is to constrain the language of expression so that all queries terminate in finite time. The web like solution is to allow the expression of facts and rules in an overall language which is sufficiently flexible and powerful to express real life.

As it turns out, it is true of course that there is a problem that you can follow links forever in the Web. And on the Semantic Web an inference engine will not necessarily terminate. However, one the Web there are many subsystems such as many websites where life is very ordered and predictable, and searches give definitive results and there are no dangling links.

In summary, the Semantic Web is still being developed and many issues needs to be further addressed. To achieve the ultimate goal of being a world wide web is long way to go, current research focus on specific domains, such as Chemistr, Biology, to try to limit the scope of problem set.

Reference

[1] Semantic Web Road map www.w3.org/DesignIssues/Semantic.html

[2] How to Make a Semantic Web Browser
WWW 2004, May 17–22, 2004, New York, New York, USA.ACM 1-58113-844-X/04/0005.

[3] RSS 1.0 specification http://web.resource.org/rss/1.0/spec

[4] Guru News Ticker & RSS Reader http://www.cmpsoft.com/products_ticker.asp

[5] The Semantic Web (for Web Developers)
http://logicerror.com/semanticWeb-webdev

[6] Does Google Show How the Semantic Web Could Replace Public Key Infrastructure?
www.w3.org/2002/03/**key**-free-trust.html

[7] Tim Berners-Lee. Inconsistent data---Semantic Web design issues
www.w3.org/DesignIssues/Inconsistent.html

[8] Tim Berners-Lee. What the Semantic Web can represent---Semantic Web design issues. www.w3.org/DesignIssues/RDFnot.html