

Tim Berners-Lee

Date: September 1998. Last modified: \$Date: 1998/09/17 20:10:41 \$

Status: . Editing status: Comments please. An parenthetical discussion to the [Web Architecture at 50,000 feet](#). and the [Semantic Web roadmap](#).

[Up to Design Issues](#)

Parenthetically, so as not to disturb the flow of what a semantic web *is*,...what it is not, and how other data models map into directed labelled graphs.

What the Semantic Web can represent

There are many other data models which RDF's Directed Labelled Graph (DLG) model compares closely with, and maps onto. This page is written with the intention of enumerating the similarity and differences between the models, to indicate how the mapping might be done and what extra information must be added in the process.

Where the other models are related to previous unmet promises of computer science, now passed into folk law as unsolvable problems, they suggest a fear that the goal of a Semantic Web is inappropriate.

One consistent difference between the Semantic Web and many data models for programming languages is the "closed world assumption".

A Semantic Web is not Artificial Intelligence

The concept of machine-understandable documents does not imply some magical artificial intelligence which allows machines to comprehend human mumblings. It only indicates a machine's ability to solve a well-defined problem by performing well-defined operations on existing well-defined data. Instead of asking machines to understand people's language, it involves asking people to make the extra effort

Even though it simple to define, RDF at the level with the power of a semantic web will be complete language, capable of expressing paradox and tautology, and in which it will be possible to phrase questions whose answers would to a machine require a search of the entire web and an unimaginable amount of time to resolve. This should not deter us from making the language complete. Each mechanical RDF application will use a schema to restrict its use of RDF to a deliberately limited language. However, when links are made between the RDF webs, the result will be an expression of a huge amount of information. It is clear that because the Semantic Web must be able to include all kinds of data to represent the world, tha the language itself must be completely expressive

A semantic Web will not require every application to use expressions of arbitrary complexity

Even though the language itself allows expressions of arbitrary complexity and computability, applications which generate RDF will in practice be limited to generating simple expressions such as access control lists, privacy preferences, and search criteria. This does not mean that where a "not" is needed, it should not be drawn from a standard vocabulary so than any RDF engine will be able to recognise it as a "not".

(more)

A semantic Web will not require proof generation to be useful: proof validation

will be enough.

The first uses, such as access control on web sites, involve validation of a previously prepared proof, not a requirement to answer an arbitrary question, find the path the construct a valid proof. It is well known that to search for and generate a proof for an arbitrary question is typically an intractable process for many real world problems, and RDF does not require this (unsolvable) problem to be solved to be useful.

A semantic web is not an exact rerun of a previous failed experiment

Other concerns at this point are raised about the relationship to Knowledge representation systems: has this not been tried before with projects such as [KIF](#) and [cyc](#)? The answer is yes, it has, more or less, and such systems have been developed a long way. They should feed the semantic Web with design experience and the Semantic Web may provide a source of data for reasoning engines developed in similar projects.

Many KR systems had a problem merging or interrelating two separate knowledge bases, as the model was that any concept had one and only one place in a tree of knowledge. They therefore did not scale, or pass the test of independent invention. [see [evolvability](#)]. The RDF world, by contrast is designed for this in mind, and the retrospective documentation of relationships between originally independent concepts.

Knowledge Representation goes Global

Knowledge representation is a field which is currently seems to have the reputation of being initially interesting, but which did not seem to shake the world to the extent that some of its proponents hoped. It made sense but was of limited use on a small scale, but never made it to the large scale. This is exactly the state which the hypertext field was in before the Web. Each field had made certain centralist assumptions -- if not in the philosophy, then in the implementations, which prevented them from spreading globally. But each field was based on fundamentally sound ideas about the representation of knowledge. The Semantic Web is what we will get if we perform the same globalization process to Knowledge Representation that the Web initially did to Hypertext. We remove the centralized concepts of absolute truth, total knowledge, and total provability, and see what we can do with limited knowledge.

The Semantic Web and Entity-Relationship models

Is the RDF model an entity-relationship mode? Yes and no. It is great as a basis for ER-modelling, but because RDF is used for other things as well, RDF is more general. RDF is a model of entities (nodes) and relationships. If you are used to the "ER" modelling system for data, then the RDF model is basically an opening of the ER model to work on the Web. In typical ER model involved entity types, and for each entity type there are a set of relationships (slots in the typical ER diagram). The RDF model is the same, except that relationships are first class objects: they are identified by a URI, and so anyone can make one. Furthermore, the set of slots of an object is not defined when the class of an object is defined. The Web works though anyone being (technically) allowed to say anything about anything. This means that a relationship between two objects may be stored apart from any other information about the two objects. This is different from object-oriented systems often used to implement ER models, which generally assume that information about an object is stored in an object: the definition of the class of an object defines the storage implied for its properties.

For example, one person may define a vehicle as having a number of wheels and a weight and a length, but not foresee a color. This will not stop another person making the assertion that a given car is red, using the color vocabulary from elsewhere.

Apart from this simple but significant change, many concepts involved in the ER modelling take across directly onto the Semantic Web model.

The Semantic Web and Relational Databases

The semantic web data model is very directly connected with the model of relational databases. A relational database consists of tables, which consists of rows, or records. Each record consists of a set of fields. The record is nothing but the content of its fields, just as an RDF node is nothing but the connections: the property values. The mapping is very direct

- a record is an RDF node;
- the field (column) name is RDF propertyType; and
- the record field (table cell) is a value.

Indeed, one of the main driving forces for the Semantic web, has always been the expression, on the Web, of the vast amount of relational database information in a way that can be processed by machines.

RDF's serialization format -- its syntax in XML -- is a very suitable format for expressing relational database information.

Relational database systems, manage RDF data, but in a specialized way. In a table, there are many records with the same set of properties. An individual cell (which corresponds to an RDF property) is not often thought of on its own. SQL queries can join tables and extract data from tables, and the result is generally a table. So, the practical use for which RDB software is used typically optimized for soing operations with a small number of tables some of which may have a large number of elements.

RDB systems have datatypes at the atomic (unstructured) level, as RDF and XML will/do. Combination rules tend in RDBs to be loosely enforced, in that a query can join tables by any comlumns which match by datatype -- without any check on the semantics. You could for example create a list of houses that have the same number as rooms as an employee's shoe size, for every employee, even though the sense of that would be questionable.

The Semantic Web is not designed just as a new data model - it is specifically appropriate to the linking of data of many different models. One of the great things it will allow is to add information relating different databases on the Web, to allow sophisticated operations to be performed across them.

RDF is not an Inference system

I am not proposing any FPOC or HOL inference engine. I just note that HOL allows integration of multiple systems which use different inference engines spanning the range from from SQL to AI. For example, a simple HOL would allow any SHOE rules, data and results expressed, and a proof found by a SHOE engine to be verified by anyone.

Surely all first-order or higher-order predicate calculus based systems (such as KIF) have failed historically to have wide impact?

The same was true of hypertext systems between 1970 and 1990, ie before the Web. Indeed, the same objection was raised to the Web, and the same reasons apply for pressing on with the dream.

The problem with all such systems was that they were conceptually or physically centralized. They required link global consistency.

Guess what? KIF is very centralized in its approach to organizing knowledge (the cyc ontology for example suggests that everyone agree on the same terms for common english words, which RDF does not) and it does not promote its concepts to being first class web objects, ie it doesn't use URIs to identify them. To webize KIF or KR in general is, in many ways, the same as to webize hypertext in many ways. Replace identifiers with URIs. Remove any requirement for global consistency. Put in a significant effort into getting critical mass. Sit back.

Surely, many things expressible in FOPC are not efficiently computable?

Dead right. The goal of the semantic web is to express real life. Many things in real life, real questions which we will face are not efficiently computable. There are two solutions to this: The classical (pre-web) solution is to constrain the language of expression so that all queries terminate in finite time. The weblike solution is to allow the expression of facts and rules in an overall language which is sufficiently flexible and powerful to express real life. Create subsets for the web in which specific constraints give you specific computational properties. An analogy is with the human-information systems which existed before the web. Most forced one to keep one's data in a hierarchy (sometimes of fixed depth or a matrix (often with a specific number of dimensions)). This gave consistency properties within the information system. I bet DARPA has many of these systems and still does. The only way they could be integrated was to express them in terms of a much more powerful language - global hypertext. Hypertext did not have any of these reassuring properties. People were frightened about getting lost in it. You could follow links forever. 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, on 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. But there is a HUGE advantage from exposing all this information in a way that allows it to be unified with all the other systems, ordered and unordered.

We should not expect a base inference level to include non-decidable computations

I have no expectation of any inference capability in the SW core design. The semantic web does not have HOL inference as a standard. I would expect any SW compliant device to be able to *validate* a HOL proof, but not *generate* one.

If you take a non-HOL-complete language and extend it to HOL, unless you have first defined where you are going (by defining the HOL language and expressing SHOE in it first) you will very likely end up with a rather baroque HOL language.

The FOPC inference model is extremely intolerant of inconsistency [i.e. $P(x)$ & NOT ($P(x)$) \rightarrow Q], the semantic web has to tolerate many kinds of inconsistency.

Tolerance of inconsistency can only be done by fuzzy systems. We need a semantic web which will provide guarantees, and about which one can reason with logic. (A fuzzy system might be good for finding a proof -- but then it should be able to go back and justify each deduction logically to produce a proof in the unifying HOL language which anyone can check) Any real SW system will work not by believing anything it reads on the web but by checking the source of any information. (I wish people would learn to do this on the Web as it is!). So in fact, a rule will allow a system to infer things only from statements of a particular form signed by particular keys. Within such a system, an inconsistency is a serious problem, not something to be worked around. If my bank says my bank balance is \$100 and my computer says it is \$200, then we need to figure out the problem. Same with launching missiles, IMHO. The semantic web model is that a URI

dereferences to a document which parses to a directed labeled graph of statements. The statements can have URIs as parameters, so they can make statements about documents and about other statements. So you can express trust and reason about it, and limit your information to trusted consistent data.

Again, extension to higher order logic makes sense to me, requirement of FOPC inference model seems dangerous.

Most KR systems confuse information with inference rules. When a system stores a rule *a daughter of one's daughter is one's granddaughter* it is typically not just stored as that statement, but in a table of rules to be used by the algorithm at a particular time (for example whenever a parent of a daughter is found). The classification between data and various type of rule is a sort of meta level information which is general not itself expressed in the language. Two systems must be able to interchange the logical meaning of the rule, even when the type of rule may be unknown to each others inference engines. (Of course, the rule expressed in general logic may be recognizable as a rule by another system and absorbed as such.) The example above is logically

$$\forall \alpha, \beta, \gamma (d(\alpha, \beta) \ \& \ d(\beta, \gamma) \Rightarrow gd(\alpha, \gamma))$$

while for example a SHOE-based system and an Algernon-based system may have quite different systems for applying rules at different times.

Conceptual Graphs and the Semantic Web

I have written [a separate set of notes](#) about the relationship between Conceptual Graphs and the Semantic Web.

A few unsorted references - see also other pages in this set.

- [SHOE: simple hypertext ontology extensions](#)

Shoe

References on KR on the Web from Tim Finin:

Here are some relevant papers from the [IJCAI-99 Workshop on Intelligent Information Integration](#). The first is a nice overview...

- [Practical Knowledge Representation for the Web](#), Frank van Harmelen and Dieter Fensel,
- [UML as an Ontology Modelling Language](#), Stephen Cranefield, Martin Purvis,
- [On2broker: Semantic-Based Access to Information Sources at the WWW](#), Dieter Fensel, Jurgen Angele, Stefan Decker, Michael Erdmann, Hans-Peter Schnurr, Steffen Staab, Rudi Studer, Andreas Witt,

and here are some others of possible interest...

Embedding Knowledge in Web Documents, Philippe Martin and Peter Eklund, Eighth International World Wide Web Conference, Toronto, May 11-14, 1999.

Ontobroker: Or How to Enable Intelligent Access to the WWW, Dieter Fensel, Stefan Decker, Michael Erdmann, and Rudi Studer, Eleventh Workshop on Knowledge Acquisition, Modeling and Management, Voyager Inn, Banff, Alberta, Canada, Saturday 18th to Thursday 23rd April, 1998

and if we want a good overview of cyc as a backgrounder

CYC: A Large-Scale Investment in Knowledge Infrastructure Douglas B. Lenat, CACM, 1995. I have a local copy at <http://www.cs.umbc.edu/471/papers/cyc95.pdf>