The Knowledge Engineering Review, Vol. 17:1, 87–91. © 2002, Cambridge University Press
DOI: 10.1017/S0269888902000346 Printed in the United Kingdom

A semantic continuum on the semantic web*

MIKE USCHOLD

The Boeing Company, PO Box 3707, m/s 7L-40, Seattle, WA USA; e-mail: michael.f.uschold@boeing.com

1 Introduction

In the coming years, the Web is expected to evolve from a structure containing information resources that have little or no explicit semantics to a structure having a rich semantic infrastructure. The key defining feature that is intended to distinguish the future Semantic Web from today's Web is that the *content of the Web will be usable by machines* (i.e. software agents). Meaning needs to be communicated between agents who advertise and/or require the ability to perform tasks on the Web. Agents also need to determine the meaning of passive (i.e. non-agent) information resources on the web to perform these tasks.

There is a great deal of excitement these days about this so-called "Semantic Web". There is a feature article in Scientific American (Berners-Lee *et al.*, 2001), there is a major DARPA program devoted to it (DAML, 2001) and the World Wide Web Consortium (W3C) has embraced it as a formal activity (W3C, 2001). There is much research activity devoted to developing a supporting technological infrastructure. However it remains a vision, not a reality.

The goal of this paper is to clarify what is meant by the term "semantics" as it applies to the Semantic Web. In particular, I focus attention on the question of where exactly the semantics are found and how they are used. I identify a kind of semantic continuum ranging from the kind of semantics that exists on the Web today to a rich semantic infrastructure on the Web of the future. I believe that progress in developing the Semantic Web will take place by moving along this continuum.

2 A semantic continuum

We ask three questions that give rise to a continuum of situations. At one extreme there are no semantics at all, except what is in the minds of the people who use the terms. At the other extreme, we have formal and explicit semantics that are fully automated.

1. Are the semantics explicit or implicit?
2. Are the semantics expressed informally, or formally?
3. Are the semantics intended for automated processing?

I describe four somewhat arbitrary points along this continuum – there are many cases that are not clear cut and thus arguably may fall somewhere in between.

2.1 Implicit semantics

In the simplest case, the semantics are *implicit* only. Meaning is conveyed based on a shared understanding derived from human consensus. A common example of this case is the typical use of

* The content of this paper was first presented as an invited talk called "Where is the semantics in the Semantic Web" (Uschold, 2001b).

I am indebted to Dieter Fensel for claiming that one benefit of XML is its ability to carry semantic information. My dispute of this claim gave rise to a fruitful e-mail exchange that inspired this paper. I am also grateful for many productive discussions with Rob Jasper, Peter Clark, John Thompson, Michael Gruninger, Frank van Harmlen and Anita Tyler.

XML tags, such as *price*, *address* or *delivery date*. Nowhere in the XML document, nor anywhere else, does it say what these tags mean (Cover, 1998). However, if there is an implicit shared consensus about what the terms mean, then people can embed these implicit semantics in screen-scrapers and wrappers. Online travel agents and booksellers routinely do this to find the best deals. The disadvantage of implicit semantics is that they are rife with ambiguity. People often do disagree about the meaning of a term. For example, prices come in different currencies.

2.2 Informal semantics

At the next point on the continuum, the semantics are explicit and are expressed in an *informal* manner, often a text specification document. Until the natural language processing problem is solved, only humans can make direct use of informally expressed semantics. Examples of informal semantics that are expressed in text specification documents are (1) the meaning of tags in HTML, for example, <h2>, which means second level header; (2) the meaning of the `subClassOf` relationship in the RDF Schema language (W3C, 1999); and (3) the meaning of expressions in programming languages, such as Java.

Typically, the semantics expressed in informal documents are embedded (i.e. hardwired) by humans in working software. Compiler writers use language definition specifications to write compilers.

The main disadvantage of informal semantics is that there is still much scope for ambiguity. This decreases one's confidence that two different implementations (say of RDF Schema or Java) will be consistent and compatible. Each Java implementation is different in sometimes subtle ways. Users may notice 'features' and start depending on them. This can result in problems if interoperability is required, or if implementations change.

2.3 Formal semantics for human processing

In this case, we have explicit semantics expressed in a formal language. However, they are intended for human processing only. There are, in effect, comments, but expressed in a formal language. Two of many examples of this are:

1. Modal logic used to define the semantics of performatives such as *inform*, and *request* in agent communication languages (ACL) (Smith *et al*., 1998). Humans use the formal definitions to understand, evaluate and compare alternative ACLs, and/or to implement agent software systems that support these notions.
2. Many axioms in many ontologies. For example, many of the definitions in the Enterprise Ontology (Uschold *et al*., 1998) were created without the expectation that they would be automatically processed. The purpose was to help communicate the intended meaning to people.

Formal semantics for human processing can go a long way to eliminating ambiguity, but because there is still a human in the loop, there is ample scope for errors.

2.4 Formal semantics for computer processing

Finally, there is the possibility of explicit, formally specified semantics that are intended for automatic machine processing. The idea is that agents on the Semantic Web can encounter new terms and interpret what they mean and thus how to use them. See Figure 1 for a simple example of this, which is possible with today's technology. Formal ontologies play a central role. In general, the way this is supposed to happen is that each agent and each information resource has an ontology that formally declares what its terms mean. If Agent 1 sends a message to Agent 2, then along with this message is a pointer to the ontology that Agent 1 is using. The theory is that Agent 2 can look in Agent 1's ontology to see what the terms mean, the message is successfully communicated, the service is performed to specification, and all live happily ever after. The holy grail is for this to happen despite

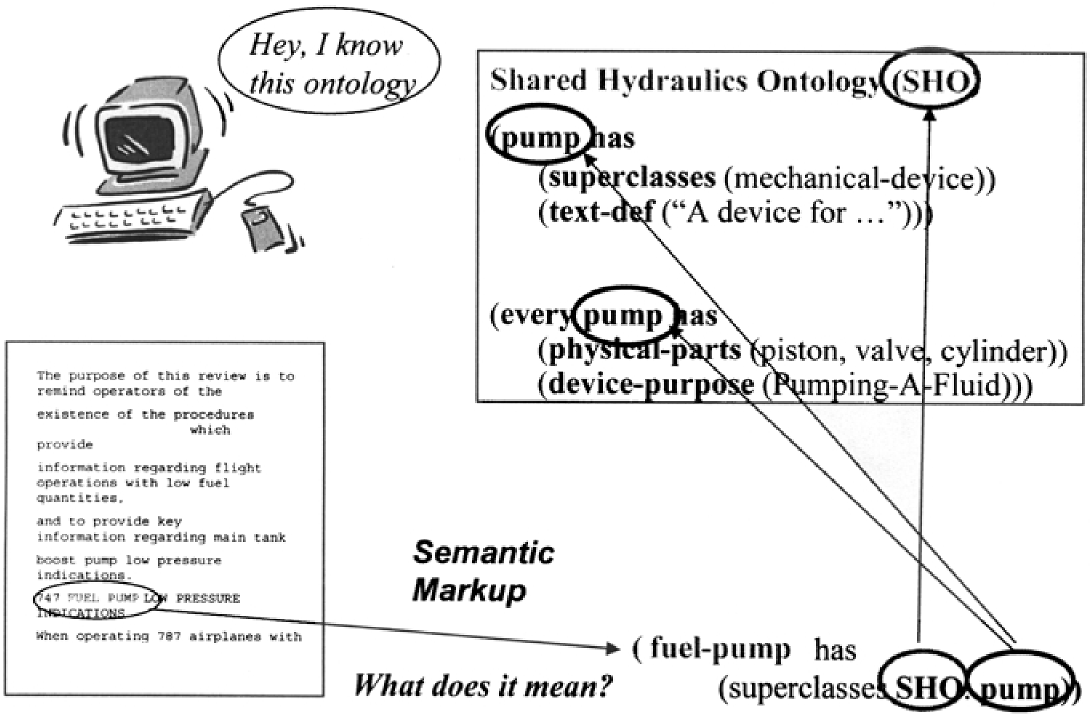

Figure 1 Automated processing of formal semantics. An agent is searching for information on mechanical devices, as defined in a public ontology (SHO). A document contains the term 'fuel-pump,' which the agent has never encountered. Semantic markup reveals that it is a kind of 'pump,' which is in turn defined in SHO as a kind of mechanical device. The agent infers that the document is relevant.

pervasive semantic heterogeneity on the Web (Uschold, 2001a). Much of the effort devoted towards building formal ontologies for the Semantic Web is directed towards this end.

3 Building the Semantic Web

According to its most commonly quoted defining feature, machine-accessible Web content, I argue that to some extent the Semantic Web already exists. Travel and bookseller agents automatically access Web pages looking for good deals. In light of the discussion above, we can see why this works. I claim that the more the following things are true:

- there is widespread agreement about the meaning of a term, and the syntax for expressing it;
- all the software is built by humans who correctly embed the agreed meaning of the term; and
- all the databases and Web pages and applications use the term in the agreed way,

then the less it is necessary to:

- have an explicit formal declarative semantics of the term that the machine can process to interpret the meaning of that term.

The point is that the machine does not *need* to learn it; the human has already embedded this meaning in the software. The Java compiler already "knows" about Java language statements, just as the Web travel shopbot "knows" how to find the fare for a given trip. If every travel agent site agreed to use a common XML DTD, with the tag <fare>, then travel shopbots could do their tasks. There is no requirement for inference; it is sufficient to embed the meaning of <fare> into the shopbots. To summarise this point, I conjecture that the following is a law of the Semantic Web:

> The more agreement there is, the less it is necessary to have explicit machine-processable semantics

The import of this is that we should direct our attention to identifying those circumstances where it is going to be worth the effort to represent the semantics of terms. Much can be done without it. Most of the proposed applications being talked about in the literature do not emphasise or discuss any need for using inference. There are some exceptions (Decker *et al.*, 1999), but there is much work to be done to meet performance and scale requirements of the World Wide Web. For further discussion of inference on the Semantic Web, see Jasper and Tyler (2001).

We envision that progress in developing the Semantic Web will proceed by moving along the semantic continuum described above. Initially, we need to make a large number of simplifying assumptions (e.g. about homogeneity). Then they should be relaxed, one by one, moving ever closer to the goal of a semantically integrated Web. To my knowledge, there remains a dearth of well-developed use cases for end-user Semantic Web applications that clearly show the need for automated semantics processing. It seems much more technology driven – build it and they will come. There needs to be a better balance between technology push and application-demand pull. A wide variety of challenge problems should be identified. These should be major drivers for choosing which assumptions to relax, and thus in which areas to make progress first.

Arguably, the most challenging problem facing the development of a Semantic Web is to achieve integration in the face of pervasive semantic heterogeneity, e.g. different languages, different terms, different conceptualizations. We have to accept that there will never be global standards on all these things. Therefore, key research areas are semantic mapping, translation and interoperability. However, because the advantages of standards are so great, the goal should be to work towards homogeneity wherever it is possible to obtain it, and to develop mapping and translation capabilities where heterogeneity remains.

Lightweight "ontologies", such as the Yahoo! subject taxonomy, may be the only ontologies to have much commercial impact in the short term. This work is important. At the other end of the spectrum is the development and use of semantically rigorous ontologies and knowledge bases that can support automated inference. Challenge problems in the form of well-developed use cases that require inference should be developed.

References

Berners-Lee, T, Hendler, J and Lassila, O, 2001, "The Semantic Web" *Scientific American* **284**(5) 34–43; `http://www.scientificamerican.com/2001/0501issue/0501berners-lee.html`.

Cover, R, 1998, "XML and semantic transparency" the XML Cover Pages `http://www.oasis-open.org/cover/xmlAndSemantics.html`.

DAML, 2001, see `http://www.daml.org/`.

Hendler, J, 2001, "Agents and the Semantic Web" *IEEE Intelligent Systems* **16**(2) 30–37.

Jasper, R and Tyler, A, 2001, "The role of semantics and inference in the semantic web: a commercial challenge" *Proceedings of the International Semantic Web Working Symposium held at Stanford University*, `http://www.semanticweb.org/SWWS/program/position/soi-jasper.pdf`.

Smith, I, Cohen, P, Bradshaw, J, Greaves, M and Holmback, H, 1998, "Designing conversation policies using joint intention theory" *Proceedings of the Third International Conference on Multi Agent Systems* (ICMAS-98) pp. 269–276.

Decker, S, Erdmann, M, Fensel, D and Studer, R, 1999, "Ontobroker: ontology based access to distributed and semi-structured information" in R Meersman *et al.* (eds.) *Semantic Issues in Multimedia Systems. Proceedings of DS-8* Kluwer Academic Publisher, Boston; see also `http://ontobroker.aifb.uni-karlsruhe.de/index_ob.html`.

Uschold, M, 2001a, "Barriers to effective agent communication" *Proceedings of the Workshop on Ontologies in Agent Systems, 5th International Conference on Autonomous Agents* `http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-52/oas01-uschold.pdf`.

Uschold, M, 2001b, "Where is the semantics on the Semantic Web?" Invited presentation at the *Workshop on Ontologies in Agent Systems, 5th International Conference on Autonomous Agents* `http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-52/oas01-uschold-presentation.pdf`.

Uschold, M, King, M, Moralee, S and Zorgios, Y, 1998, "The enterprise ontology" *Knowledge Engineering Review* 31–89.

W3C, 1999, *Resource Description Framework (RDF) Schema Specification* (ed. D. Brickly and R. Guha) `http://www.w3.org/TR/1998/WD-rdf-schema/`.

W3C, 2001, Semantic Web Activity Statement `http://www.w3.org/2001/sw/Activity`.