# Creating Semantically Integrated Communities on the World Wide Web[†]

**Invited Talk**
**Semantic Web Workshop**

Co-located with WWW 2002
Honolulu, HI
May 7 2002

**Michael Uschold* and Michael Gruninger****

*The Boeing Company
PO Box 3707 MS 7L-40
Seattle, WA 98124 USA
michael.f.uschold@boeing.com

**Institute for Systems Research
University of Maryland
College Park, MD 20742
gruning@cme.nist.gov

**Abstract**

In this paper, we address the question: *How can we create a network of semantically integrated communities on the World Wide Web?* We first clarify some confusion about what "semantics" means and introduce a semantic continuum ranging from the kind of semantics that exist on the Web today to a rich semantic infrastructure on the Semantic Web of the future. We clarify what is meant by "semantic integration" introducing and defining a "gold standard" whereby two agents that have never met before can successfully exchange information. We acknowledge that this gold standard will only be reachable in limited circumstances, and that a variety of approaches will be needed to achieve successful agent interaction in practical situations on the semantic Web. Towards this end, we introduce several architectures for achieving semantic integration. Each are defined and compared on the basis of how the following questions are answered. Who and when are semantic mappings created between agent ontologies? Is the architecture point to point between each agents, or mediated through another ontology? What is the nature of agreements among the agents? We conclude by making some predictions and recommendations on how the semantic Web will evolve in the coming years.

## 1 A Semantic Continuum

If the Semantic Web becomes a reality, it will emerge from the creation of a network of semantically integrated communities on the World Wide Web. Before we consider semantic integration, we first consider the nature of semantics, as understood by a variety of different people and communities. There seems to be broad agreement that the core idea underlying the "Semantic Web" is *machine-usable content*. Beyond that, agreement falls off rapidly. In the context of the Semantic Web, ask 5 people the following questions, and you might get dozens of different answers.

1. *What* do we mean by semantics?
2. *Where* are the semantics?
3. *When* and *Why* do we need semantics?
4. *Who* is responsible for specifying semantics and negotiating semantic agreements?
5. *How* do we cope with pervasive semantic heterogeneity on the Web?

[†] The major content of this paper is drawn from a much larger report [Gruninger & Uschold 2002] on ontologies and semantic integration.

To better understand the idea of semantics in the Semantic Web context, we introduce a semantic continuum ranging from the kind of semantics that exist on the Web today to a rich semantic infrastructure on the Semantic Web of the future (see figure 1) [Uschold 2002]. At the far left of this continuum, the meaning of Web content is not specified anywhere—rather, it is derived from human consensus and common usage of terms. At the other end, we have formal semantics intended for machine-processing.

Good examples of Web software at the left end of the continuum are today's so-called shopping agents that troll the Web to find the best prices for travel fares, books, and many other products. These programs are highly useful despite there being no explicit semantics for terms like "price", "destination", or "author" that the machines (i.e. agent software programs) have access to. Instead, humans hardwire the semantics of these terms so that the programs do the right thing. Collectively, shopping agents may be thought of as a degenerate case of the semantic Web because (1) they are a clear example of *machines using Web content* to do useful tasks *and* (2) there is no explicit semantics.

A hypothetical example of a Web application at the far right of the semantic continuum is illustrated in figure 2. From a formal specification of the semantics of a term, in conjunction with a pointer to a shared ontology, the agent is able to dynamically discover something about the meaning of the term, "FUEL PUMP", that it had never encountered before. This ability can be used to retrieve relevant documents and is easy to implement using today's technology.

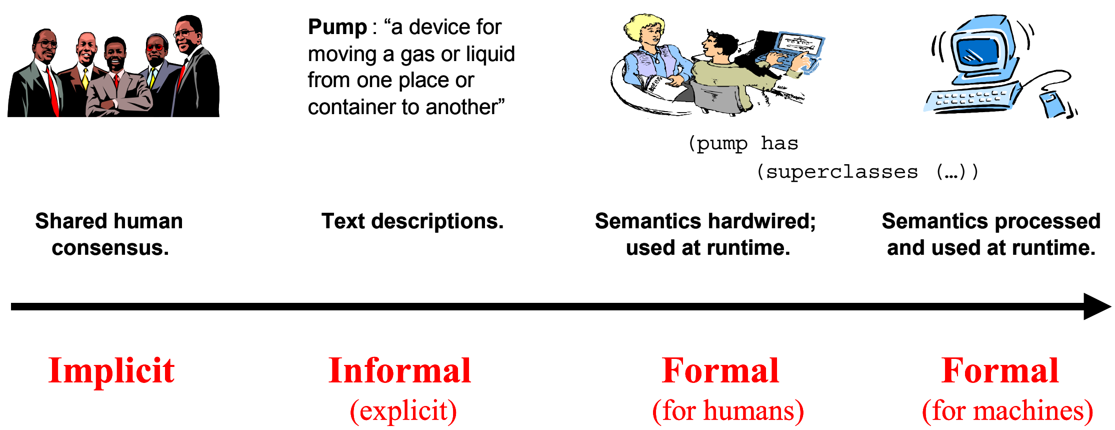


**Figure 1: Semantic Continuum** —*Semantics may be implicit, existing only in the minds of the humans who communicate and build Web applications. They may also be explicit and informal, or they may be formal. The further we move along the continuum, the less ambiguity there is and the more likely it is to have robust correctly functioning Web applications. For implicit and informal semantics, there is no alternative to hardwiring the semantics into Web application software. In the case of formal semantics, hardwiring remains an option, in which case the formal semantics serve the important role in reducing ambiguity in specifying Web application behavior, compared to implicit or informal semantics. There is also the new possibility of using automated inference to process the semantics at runtime. This would allow for much more robust Web applications, in which agents automatically learn something about the meaning of terms at runtime.*

This simple example hints at how we might create a network of semantically integrated communities. The key idea is to have formal shared semantic repositories that are available to and processed by a community of software agents. There are many other important ideas that are required to support the creation and maintenance of semantically integrated communities on a future semantic Web. Next we consider the idea of semantic integration in general, and examine a variety of architectures for achieving it.

## 2 What do we mean by Semantic Integration?

Given the lack of agreement on what "semantics" means, it is no surprise that there is also little agreement on what it means to be "semantically integrated", and even less on how this should be achieved. Informally, two agents are semantically integrated if they can successfully communicate with each other. We introduce a gold standard that we call "*complete semantic integration*". The idea is that agents have never met before. The only access to the meaning of another agent's terms is via the axioms of that agent's ontology. That is, there is no access to what the human agent designer had in mind. This contrasts, for example with the situation today where people markup their documents referring to terms from the Dublin Core [Weible & Miller 2000]. For a software agent to know what to do with Dublin Core terms, the semantics of those terms needs to be hardwired by humans into the code. The humans have access to the informal specification of Dublin Core terms.

Successful exchange of information means that the agents understand each other and there is guaranteed accuracy. Below we elaborate on what we mean by "understanding" and how we might provide guarantees.

We introduce the idea of a *verified ontology* that is analogous to a verified software program. We can never prove that a software program does exactly what the human intended. However, if we introduce a formal specification for the program, it is possible in principle to prove that the program faithfully implements the specification[‡] [Astesiano et al 1999; Srinivas & Jullig 1995]. We can never prove that the formal specification expresses what the human intended. Nevertheless the added step of having a formal specification forces discipline on the requirements specification designer and enables precise communication between designer, customer and software engineer. Overall, this increases the likelihood that the software does what was intended.

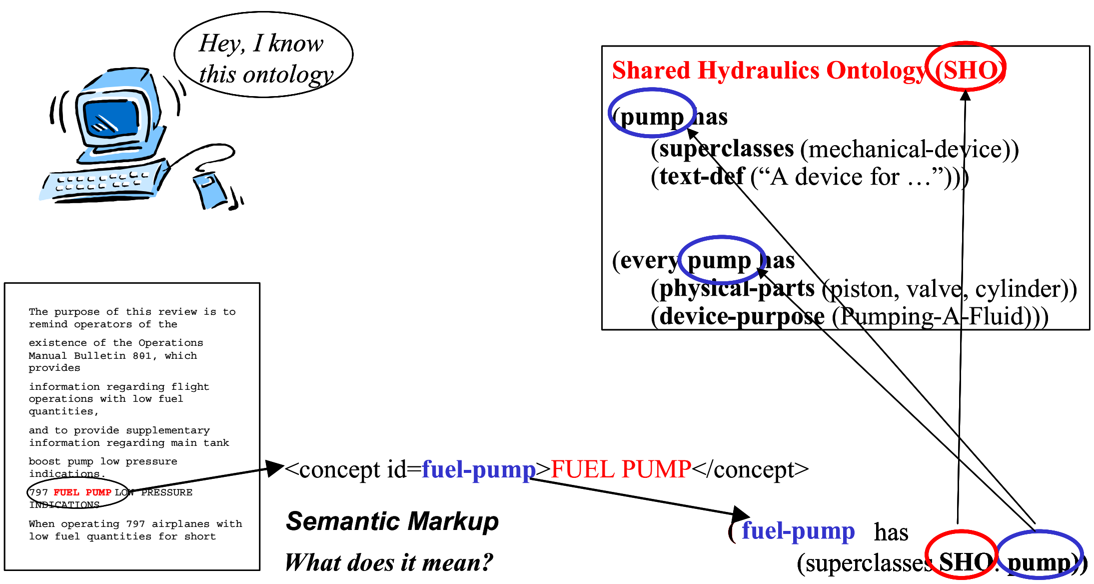


**Figure 2: Formal Semantics for Machine Processing** —*An agent is searching for information about mechanical devices, as defined in a public ontology (SHO). A document contains the term "FUEL PUMP," which the agent has never encountered. Semantic markup reveals that it refers to the concept* `fuel-pump`*, which 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.*

By analogy, we introduce the idea of a formal specification of what the ontology is supposed to represent. These are the *intended models* of the ontology. The actual models are reflected by the set of axioms that

‡ With today's technology, it is also possible in practice, for modest-sized software components.

comprise the ontology. A *verified ontology* is one whose intended models are proven to be the same as the actual models. Under certain circumstances, it is possible to do these proofs [Menzel and Gruninger 2001]. By analogy with software verification, we can never prove that the formal specification of intended models expresses what the human ontology designer intended. However, but this added step forces discipline and enables precise communication between ontology designers and agent designers. Overall, this increases the likelihood that the ontology represents what the ontology designer intended.

We claim that a *verified ontology is necessary for complete semantic integration*. It is not sufficient, however. If one agent's ontology does not include all the concepts in the other agent's ontology, then even if the ontologies are verified, successful communication cannot take place. Complete semantic integration thus requires additionally that the intended models of both agents are the same. Informally, this requirement says that the scope of the ontologies is the same.

If the ontologies are verified, then proving that the intended models are the same is equivalent to proving that the logical theories captured in each ontology are logically equivalent. This is accomplished by hypothesizing and testing mappings between both ontologies. The idea is to verify that all inferences that hold for one agent, also hold when translated into the other ontology. If the inference does not hold, then the mapping is incorrect. Operationally, for one agent to "understand" another agent means that the other agent behaves in the expected way. The idea is that if the agent has the ontology that you think it does, and if you have the correct mapping to that ontology, then it should behave as you expect it to behave. In a formal setting, behavior is exactly aligned with the inferences that are made from the agents ontology. This idea is called the *ontological stance* [Menzel and Gruninger 2001].

In the absence of a procedure for generating and verifying all possible inferences for a given mapping, we will never be able to prove equivalence. However, each inference that does hold increases confidence that the mapping is correct, that the ontologies are logically equivalent and that the agents will "understand" each other.

We propose this as a gold standard for semantic integration. We recognize that these are very strict requirements and that in practice this will only be achievable for relatively few niche applications. In the greater Web, incomplete semantic integration will be the rule. Agents' ontologies cannot be assumed to be verified, nor will there likely be complete overlap in all the agents' ontologies. Next we consider a variety of architectures for achieving semantic integration in a practical context.

# 3 Architectures for Semantic Integration

There are a variety of architectures that may be used to achieve semantic integration. The differences depend on the origins of the semantic mappings, whether there is a mediating ontology, and the nature and degree of the agreements that exist among the anticipated community of interacting agents. Different architectures can be distinguished and compared to one another by considering the following questions:

1. Who is generating the agent to agent semantic mapping?
    a. The *agent designer*.
    b. The *ontology designer*.
    c. The *agents*.
2. When is the mapping between two agents' ontologies created?
    a. Mappings are *pre-defined* before the agents interact.
    b. Mappings are dynamically generated at *agent-interaction time*.
3. What is the topology of the architecture?
    a. Mapping is done *point-to-point* between the agents.
    b. Mapping is *mediated* (e.g. by a neutral ontology).
4. What is the nature of the agreements among the agents?
    a. Agreement is on a single *global ontology* for all interacting agents.
    b. Agreement is on an *interlingua ontology*.
    c. Agreement is on *alignments/mappings* between ontologies.

d. There is *no* a priori agreement.

We outline five architectures that can be used to integrate agents. Each answers the above questions in different ways. The properties of these various architectures are briefly described below and summarized in Table 1.

| **Questions** / **Architecture** | **Who generates the mappings?** | **When define Agent to Agent mapping?** | **Topology** | **Degree of Agreement** |
|---|---|---|---|---|
| **Global ontology** | *no mappings* | *no mappings* | Point-to-point | Agree on Everything |
| **Manual mapping** | Agent designers | Before agents interact. | Point-to-point | No *a priori* agreement |
| **Interlingua ontologies** | Agent designers | Auto-generated at agent interaction time. | Mediated | Agree on Interlingua ontologies |
| **Community ontologies** | Ontology designers | Auto-generated at agent interaction time. | Mediated | Agree on alignment mappings |
| **Ontology Negotiation** | Agents themselves | Auto-generated at agent interaction time. | Point-to-point | No *a priori* agreement |

Table 1: Semantic integration architectures.

**Ontology Negotiation** [Truszkowski & Bailin 2001] **–** In the Ontology Negotiation architecture, the agents themselves generate and test the mappings automatically, at agent-interaction time. There is no mediated ontology, the mappings are point to point between the agents. There are no a priori agreements. To do this reliably an consistently is the Holy Grail of semantic integration.

**Global Ontology—**In this case, we assume that all agents use the same ontology. This approach alleviates the need for mappings entirely. This architecture is severely limited. It is only practical for small communities, or where there is an able and powerful dictator.

**Manual Mapping** ([Obrst 2001], [Fillion *et al*. 1995]) **–** In the case of Manual Mapping, the human agent designers specify the agent to agent mapping between the agent's ontologies prior to their interaction. These mappings are point-to-point between the agents. There is no *a priori* agreement about semantics between the agents. This architecture can be thought of as a fully manual version of ontology negotiation.

**Interlingua** [Ciociou *et al* 2001]**–** In the Interlingua architecture, each agent designer generates a mapping from their agent's ontology to a standard interchange ontology, or interlingua. This is done before the agents interact. The agent to agent semantic mappings are generated dynamically at agent-interaction by executing the pre-specified mappings to and from the interlingua. In this case, the interlingua ontology mediates the mapping between the agent ontologies. The agents that wish to participate in this architecture must agree *a priori* to use the interlingua ontology. This is a partially automated version of ontology negotiation.

**Community Ontologies –** In the Community architecture, we assume the existence of a library of ontologies that has been built by aligning and mapping ontology modules developed by some user community. The ontology designers create the alignments and mappings before agent-interaction time.

Different agent designers use ontologies from this library. When the agents interact, they invoke these pre-specified inter-ontology mappings in order to automatically generate the agent to agent mappings. This architecture uses the various community ontologies as mediating ontologies, rather than a single interlingua ontology. This approach is also a partially automated version of ontology negotiation. This is an elaboration of the idea of agents specifying their semantics by pointing to existing ontologies on the Web [Hendler 2001].

These architectures should be thought of as the building blocks for the semantic Web of the future—not mutually exclusive alternatives. All of these, and perhaps other approaches will evolve and be combined in creative ways. Note also, that these architectures typically can be applied at various points along the semantic continuum. This is especially so for the manual mapping, interlingua, and community approaches. From an integration perspective, the issue is moot for the global ontology architecture. Given the limitations of current technology, the fully automated ontology negotiation architecture is likely to remain relatively informal for the foreseeable future.

# 4 Summary and Conclusions

In this paper, we have addressed the question of what is meant by "semantics" in the semantic Web. It means many thing to many people, as indicated by different points along a semantic continuum ranging from implicit semantics that is derived from human consensus, to formal semantics for machine processing. We defined a gold standard for semantic integration, in which agents that have never met before can successfully exchange meaning. We introduced the idea of a verified ontology which is necessary for complete semantic integration. Finally, we considered a variety of architectures for achieving semantic integration. Each were defined and compared on the basis of the origins of the semantic mappings, the topology of the architecture, and the nature of agreements.

We anticipate that progress in development of the Semantic Web will take place in various ways. One aspect will be movement along the semantic continuum. As machine processing of formal semantics increases, the amount of hardwiring of semantics into Web applications will decrease. Ongoing efforts in reaching public standards and agreements on protocols and languages will be augmented by the establishment of shared semantic repositories of content ontologies. They will be at various points along the semantic continuum ranging from the highly informal Dublin Core to the very formal Process Specification Language [Menzel and Gruninger 2001]. Whenever it is possible and appropriate, reaching agreements is extremely powerful, in that it eliminates the need for mapping. Where agreements are not forthcoming, there will be ongoing development of technologies for semantic mapping and translation.

We expect that in the medium and long term, a suite of mapping and translation architectures, tools and approaches will become available, each suited for a particular well-understood niche. It is important to approach the semantic integration problem from both theoretical and pragmatic standpoints. Pragmatic approaches are required to meet the ever-growing need for semantic mapping and translation. Ad hoc point solutions with any number of (possibly implicit) assumptions will continue to be created. This needs to be complemented by rigorous formal approaches so that we can understand what if any fundamental limits exist. This in turn, can guide the development of pragmatic solutions. By analogy, theoretical work in knowledge representation revealed that there is a fundamental tradeoff between expressive power and computational efficiency [Levesque & Brachman 1985]. This has had a large impact on the development of practical knowledge representation and inference systems.

Finally, we note that *there is no need for semantics envy*—there is nothing inherently superior about being further along the semantic continuum. In some cases there will be advantages; in other cases there will not. What is good is what works. We believe that in the short and possibly medium term, approaches that do not make use of machine processible semantics are likely to have the most impact on the development of the Semantic Web. They will mainly use informal or lightweight ontologies. There will be very few applications that require and use fully automated processing of complex ontologies with formal semantics. Eventually, there will be sufficiently large niche markets that require to justify this complexity and expense. We believe that even in the long term, these "high-end" semantic Web applications are likely to comprise a relatively small portion of Web applications overall.

## 5 References


[Astesiano et al 1999] E. Astesiano, H.-J. Kreowski and B. Krieg-Bruckner (eds.). *Algebraic Foundations of Systems Specification*. Springer.

[Ciociou *et al*. 2001] Ciocoiu, M., Gruninger M., and Nau, D. Ontologies for integrating engineering applications*, Journal of Computing and Information Science in Engineering,* 1:45-60.

[Fillion *et al*. 1995] Fillion, F., Menzel, C., Blinn, T., Mayer, R., "An Ontology-Based Environment for Enterprise Model Integration", Proceedings of the IJCAI Workshop on Basic Ontological Issues in Knowledge Sharing.

[Gruninger & Uschold 2002] M. Gruninger and M. Uschold; "Ontologies and Semantic Integration," to appear in *Software Agents for the Warfighter*, the first in a series of reports sponsored by the US Government Information Technology Assessment Consortium (ITAC). Edited by Jeff Bradshaw, Institute for Human and Machine Cognition (IHMC), University of West Florida

[Hendler 2001] Hendler, J. "Agents on the Semantic Web," *IEEE Intelligent Systems*, Vol. 16, No. 2, March/April 2001

[Hoare 1969] C.A.R. Hoare: The Axiomatic Basis of Computer Programming, CACM 12, 10, pp. 567-583, Oct. 1969

[Levesque & Brachman 1985] Levesque, H., & Brachman, R. A fundamental tradeoff in knowledge representation and reasoning (revised version). In R. Brachman & H. Levesque Readings in knowledge representation. Los Altos, CA: Morgan Kaufmann.

[Menzel & Gruninger 2001] Menzel, C. and Gruninger M. A formal foundation for process modeling, *Formal Ontology in Information Systems 2001*, Ogunquit Maine.

[Obrst *et al*. 2001] Obrst, L., Wray, R., and Liu, H. Ontology engineering for ecommerce: A real B2B example *Formal Ontology in Information Systems 2001*, Ogunquit Maine.

[Srinivas & Jullig 1995] Srinivas, Y. V. and Jullig, R., *Specware™: Formal Support for Composing Software*, in Proceedings of the Conference of Mathematics of Program Construction, Kloster Irsee, Germany, 1995.

[Truszkowski & Bailin 2001] Truszkowski, W. and Bailin, S. Ontology negotiation between strange agents, *Fifth International Conference on Autonomous Agents*, Montreal.

[Uschold 2002] Uschold, M "Where are the Semantics in the Semantic Web?" AI Magazine, (forthcoming)

[Weible & Miller 2000] Stuart Weibel, Eric Miller An Introduction to Dublin Core October 25, 2000 http://www.xml.com/pub/a/2000/10/25/dublincore/