The Suggested Upper Merged Ontology: A Large Ontology for the Semantic Web and its Applications

Adam Pease, Ian Niles, John Li
[apease | iniles | jli]@teknowledge.com
Teknowledge

Abstract

In this paper we discuss the development and application of a large formal ontology to the semantic web. The Suggested Upper Merged Ontology (SUMO) (Niles & Pease, 2001) (SUMO, 2002) is a "starter document" in the IEEE Standard Upper Ontology effort. This upper ontology is extremely broad in scope and can serve as a semantic foundation for search, interoperation, and communication on the semantic web.

Introduction

When you give someone new information, he can combine the new fact with his or her existing knowledge and derive additional information. When you tell a computer something in XML, it may be able to infer something else, but only because of some other software it has that's not part of the XML specification. That software might vary with respect to implementation and with respect to the answers it provides and yet still conform to the XML specification.

A new *semantic* markup language has been created called DARPA Agent Markup Language (DAML) (Hendler & McGuiness, 2000), which partially addresses these problems. If you tell a computer something in DAML, a certain set of conclusions are required from any system that conforms to DAML. Systems may be able to provide all sorts of additional services and responses beyond the standard, but a certain set of conclusions will always be required. Thus, a set of DAML statements by itself (and the DAML specification) can allow you to conclude another DAML statement, whereas a set of XML statements by itself (and the XML specification) does not allow you to conclude any other XML statements. XML can generate new data only by drawing on knowledge embedded (but not explicitly stated) in procedural code.

An example should make this clear. "Parenthood is a more general relationship than motherhood." and "Mary is the mother of Bill" together allow a system conforming to DAML to conclude that "Mary is the parent of Bill". Accordingly, if a user poses a query such as "Who are Bill's parents?" to a DAML search system, the system can respond that Mary is one of Bill's parents, even though that fact is not explicitly stated anywhere. More formally, given the statements

```
(motherOf subPropertyOf parentOf)
(Mary motherOf Bill)
```

a DAML-compliant system can conclude

```
(Mary parentOf Bill)
```

based on the logical definition of "subPropertyOf" as given in the DAML specification. On the other hand, nothing in XML would sanction this inference, since XML itself provides no semantics for its tags. One might create a program that assigns the appropriate semantics to the "subPropertyOf" tag, but since that semantics would not be part of the XML specification, applications could be written which conform to the XML specification and yet do not derive the conclusion.

Other web languages such as RDFS go a step further than XML and could support the example just given, but DAML offers a host of standard properties such as equivalence ("childOf" on an English geneology site is the same as "filsDe" on a French site), and it allows one to state that particular properties are unique (e.g. a social security number is associated with only one individual).

It should be easy to see that the additional power provided by DAML is extremely useful across a wide range of domains. For example, in financial aggregation software, one might query about all bank accounts associated with a particular person (whether they are directly owned by, held in trust for, etc). For another example, in logistics software, one might want to ask the rates for shipping to any eastern European city (where no such category has been predefined and only the countries in eastern Europe are listed). Having existing knowledge that can be dynamically applied, rather than predefined procedures, is extremely powerful.

DAML provides the basic infrastructure that allows a machine to make the same sorts of simple inferences that human beings do. It is a critical foundation for a web of information that machines can draw upon.

Large Ontologies

The DAML standard is, however, just the beginning of what is needed. Imagine the presence of thousands of rich ontologies that are expressed DAML, many of which are not linked by equivalence or subsumption relations. Imagine further the presence of powerful agents which use these ontologies. Without linkages among the ontologies, we have a "tower of babel" problem between the various agents that are trying to make use of the ontologies: the agents are saying useful things which are mutually unintelligible. A slightly more harmonious vision is one where the ontologies are linked and translation occurs during each transaction. Since this translation has a significant cost, the pressure to conform to common ontologies will be very powerful over time.

One model is that the development of these ontologies will be "bottom-up", driven by incremental, short-term communication needs. While such a model is certainly possible, we contend that it is far from optimal. In fact, it is likely to experience the difficulty faced in the reuse of expert systems in the late 80's. As system requirements expand and evolve, system often have to be reengineered from scratch in order to remove fundamental assumptions that make the model unsuitable for supporting new requirements.

Imagine that a software developer has been given the task of creating a software system to manage warehouse inventory and publish the information on the semantic web. The developer begins by defining a class of objects which have a temporal extent (life within the warehouse), a position (such as on a particular shelf or loading dock), and physical characteristics like weight and volume. After the first successful release of the software, management specifies additional system requirements to manage the tasks that workers, shippers and suppliers perform with respect to the warehouse items. These requirements mean that the software and the underlying ontology have to be changed to accommodate tasks, which have a temporal extent but not a position or other physical characteristics. After the new version of the system is deployed, the requirements are modified again to encompass purposes of the company and its partners. Since purposes have neither temporal nor spatial extent, this means that further changes will have to be made to the software and the underlying ontology.

One way to ameliorate this sort of situation is to embrace a standard upper-level ontology. While such an ontology would not obviate the need for changes to accommodate shifting requirements, it can mitigate the degree of change. An upper-level ontology can alert the software designer to fundamental modeling issues and choices, such as the distinction between objects and processes. Since there are literally thousands of such distinctions which may need to be taken into account, it is impossible to suppose that all of them might be anticipated by a smart designer.

Another, equally weighty consideration which motivates a standard upper-level ontology is reuse. With such an ontology, it is possible to apply, at very low cost, information models that have been rigorously designed and extensively tested. As an analogy, there are very few people who are capable of writing a modern operating system kernel. A slightly larger number of people have the skills to write paging systems or device drivers. Many more people write applications that use those services. Few developers would even consider writing their own operating system, and yet every year information models that are every bit as fundamental and as challenging as operating systems are written from scratch.

Thus, there are two key advantages of a standard upper-level ontology. First, such an ontology can more easily accommodate changes in system requirements by anticipating the possible forms of such changes in deep and challenging information structures such as temporal or spatial models. Second, reference to such an ontology means that developers do not have to constantly reinvent the wheel by reimplementing something that could simply be reused.

Suggested Upper Merged Ontology

The Suggested Upper Merged Ontology (SUMO) was created by merging a number of existing upper-level ontologies. These ontologies encompass content created by Sowa (Sowa, 2000), Guarino and his colleagues (Borgo et al, 1996, 1997), Allen (Allen, 1984), and Smith (Smith, 1996), as well as more concrete ontologies from the repositories at Stanford KSL and ITBM-CNR. Currently, the SUMO is divided into 11 sections whose interdependencies are carefully documented. The first section, known as the Structural Ontology, contains definitions for relations that serve as the framework for definining the ontology proper. The second section, known as the Base Ontology, consists of very fundamental ontological notions such as abstract entity and the distinction between objects and processes. The Set/Class Theory section of the SUMO consists of basic set theoretic content. The numeric section provides, among other things, definitions of basic arithmetic functions. The Temporal section builds on Allen's temporal relations (Allen, 1985). The Mereotopology ontology contains a basic axiomatization of part/whole relations, as well as a formalization of holes borrowed from (Casati and Varzi, 1998). The Graph Theory section provides general graph theoretic notions. The Unit of Measure section provides definitions of the SI and other unit systems (Pinto, 2000). The remaining sections of the ontology provide subhierarchies and axioms relating to process types (e.g. 'ChangeOfPossession' and 'Touching'), object types (e.g. 'Book' and 'Fish'), and attribute types (e.g. 'SocialRole' and properties of sensation). As of May, 2002, the SUMO contains roughly 1000 terms and 3700 statements involving those terms.

Aside from all of the inherent advantages that the SUMO has as an upper-level ontology, it is also useful because it is being mapped (Niles, 2002) to WordNet, a huge, structured lexicon of English meanings (Miller et al, 1993). As of May, 2002, roughly 50,000 WordNet synsets have been mapped to the SUMO. The major motivation for mapping SUMO to WordNet is to promote the use of SUMO in natural language understanding applications. By relating WordNet synsets to terms in a formal ontology we believe it will be possible to leverage the extensive semantic content of SUMO for sense disambiguation, anaphora resolution, summary generation, and other core natural language processing tasks. An example should make this clear. Interpreting the sentence "The board approved the pay increase." Requires knowledge beyond what is explicitly stated in the sentence, e.g. that only an agent can participate in an approval action. By creating mappings from the two WordNet senses of board (piece of wood) and board (corporate board) to the SUMO concepts 'Device' and 'Organization', respectively, we can employ the argument type restrictions on SUMO relations such as 'agent' to winnow down the space of possible parses of the sentence. More specifically, in this case we can eliminate the parse involving the "piece of wood" sense of the word "board ".

A second benefit of the mapping process is that it serves as a check on the coverage of the ontology. As the mappings are created, many gaps in the conceptual space of the SUMO have been identified. In many cases, we have discovered that the most specific concept to which a WordNet synset could be mapped was too broad in meaning and, hence, that a new, more specific SUMO concept needed to be defined. Because of the enormous number of words and meanings represented in WordNet, the number of applications that make use of it, and the intense scrutiny to which it has been put, the lexicon is very well suited as to the role of validating the SUMO.

Domain Specific Ontology

Aside from developing the SUMO and creating the mappings from SUMO to WordNet, we are also creating domain ontologies that are aligned with the SUMO. These domain ontologies inherit the broad conceptual distinctions of the SUMO, and they specify the concepts and axiomatic content of a particular domain, e.g. financial transactions and the Quality of Service (QoS) delivered by distributed computing systems. Because the SUMO provides a rich substratum of reusable content, it is easier and faster to create these domain ontologies than it would be without the SUMO. Furthermore, since these domain ontologies are consistent with the SUMO, even applications that correspond to different domains can interact at a more general semantic level. A final advantage of these domain ontologies is in the feedback they provide to the upper-level ontology. The creation of these ontologies alerts us to areas in the upper ontology that need to be fleshed out. The SUMO-compliant ontologies that have been developed thus far include an ontology of the quality of service of distributed computer services, an ecommerce ontology, and a number of military ontologies covering Air Force and Army concepts.

We anticipate that, over time, the semantic web will evolve communities of practice with various degrees of semantic agreement. While we do not expect that one particular upper-level ontology will be shared by everyone, we do expect that there will be only a handful of such ontologies, and all applications will be compliant with one or another of these ontologies in order to have some degree of mutual interoperability. We also envision that translation services, such as the one described in the next section, will be used to bridge the gap between alternative ontologies. Further, it seems likely that communities will adopt domain specific ontology standards to facilitate the interchange of information. Industry is already moving slowly in this direction with the development of XML-based domain-specific information interchange standards such as OFX (OFX, 2001). Ultimately, we expect there will be a web of ontologies, many shared, some linked to facilitate translation, and many "leaves" representing the highly specific concepts of specialized communities.

Semantic Search

Current text-based search tools often return nothing useful or a bundle of results that can be sorted out only by a human being. An extensive common ontology can radically improve this situation. By marking up documents with the concepts of a common ontology, we can do better than the simple HTML-based keyword search that is the current state of the art. The Agent Semantic Communications Service (ASCS) (Li et al, 2001) allows users to perform high-quality semantic search in the DAML-annotated web environment. ASCS consists of two main components. A Semantic Search Agent (SSA) allows agents to find entities based on the ontologies they share, and a Semantic Translation Service (STS) provides a basis for communication between agents that employ different ontologies. Furthermore, the architecture of ASCS allows any text-based search agent to work with it in such a way that an integrated set of results from both engines can be presented to users.

An example should make clear how ASCS works. First, an application can use an SSA to query very detailed information annotated in DAML somewhere on the web, e.g. personal telephone numbers. The result of this query is very accurate, because the semantic content of the telephone numbers is clearly indicated by concepts from an underlying DAML ontology. Since some pages may employ a different ontology to mark up personal telephone numbers in DAML, the STS creates a mapping between the various ontologies.

ASCS supports several kinds of simple inference that can serve to broaden queries including equivalence, inversion, generalization, and specialization. Equivalence uses the DAML samePropertyAs and sameClassAs relations to restate queries that differ only in form. Generalization and specialization utilize the subPropertyOf and subClassOf relations to find matches on more general or more specific classes and relations. Inversion, for example, allows the query (parentOf Bill Mary) to be reformulated as (childOf Mary Bill) if parentOf and childOf have been specified as inverses of each other.

ASCS is structured as a web of distributed agents. Each SSA has a set of STSes that it can call to translate content to and from other ontologies. Each STS knows about a set of SSAs to which it can post queries. Since a given SSA has indexed just a portion of the DAML web, usually with regard to the usage of one or a small number of particular ontologies, there will be a limited number of relationships between SSAs and SSTs. We anticipate that, as the DAML web scales up, we will deploy many of these agents. Similarly, many STSes will be deployed, translating among a handful of ontologies, much in the way that human translators are used to make dynamic connections between speakers of different languages.

We believe that this highly distributed approach will scale well. Our current implementation of ASCS allows configuration of a “server farm” of redundant agents where simultaneous query requests can be dynamically dispatched to the least heavily loaded process or server. Furthermore, the implementation includes a crawling/indexing component much like HTML-based search engines. A process periodically spiders the DAML web, parses each page, and builds an index that can be efficiently searched. The spider is written in Java, and the search and translation agents are implemented for the most part in Prolog.

One challenge in semantic search is that ASCS is not very forgiving of queries which are formulated vaguely or ambiguously. One way to address this issue is to make use of the SUMO/WordNet mappings. By providing tens of thousands of English terms and mapping them to more precise SUMO concepts, we allow the user to be imprecise, and let the mappings handle the burden of a more precise reformulation. There is a risk of course that the reformulation may include a mapping that was not intended by the user, but this problem can be mitigated by providing feedback to the user about the word sense that was chosen and allowing the user to modify his/her choice. It is also possible that the mappings could be used in the context of an automatic sense-disambiguation program.

References

Allen, J. (1984), Towards a general theory of action and time, Artificial Intelligence, 23, 123-154.

Berners-Lee, T., Hendler J., and Lassila, O. (2001), The Semantic Web, Scientific American, May 2001. http://www.scientificamerican.com/2001/0501issue/0501berners-lee.html

Borgo, S., Guarino, N., and Masolo, C. (1997) “An Ontological Theory of Physical Objects,” in L. Ironi (ed.), Proceedings of Eleventh International Workshop on Qualitative Reasoning (QR'97), Cortona (Italia), 3-6 Giugno, 223-231.

Borgo, S., Guarino, N., and Masolo, C. (1996) “A Pointless Theory of Space Based on Strong Connection and Congruence,” in L. C. Aiello and J. Doyle (eds.), Principles of Knowledge Representation and Reasoning (KR96), Morgan Kaufmann.

Casati, R. and Varzi, A. (1995), Holes and Other Superficialities, MIT Press, Cambridge, MA.

Hendler, J., and McGuinness, D., (2000), The DARPA Agent Markup Language, IEEE Intelligent Systems, 15 (6) (November), 67-73. Available: http://www.ksl.stanford.edu/people/dlm/papers/ieee-daml01-abstract.html

Li, J., Pease, A., and Barbee, C. (2001), Performance of Semantic Search, Teknowledge Technical Report, June 15, 2001, see http://projects.teknowledge.com/HPKB/Publications/DAML.ps

Miller, G., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K, Introduction to WordNet: An On-Line Lexical Database, August, 1993. Available: http://www.cogsci.princeton.edu/~wn/papers/

Niles, I. (2002), Mapping an Upper Ontology to WordNet, forthcoming.

Niles, I., and Pease, A., (2001), Toward a Standard Upper Ontology, in *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems* (FOIS-2001). See also http://ontology.teknowledge.com and http://suo.ieee.org

OFX (2001), Open Financial eXchange, web site, http://www.ofx.net/ofx/ab_main.asp

Pinto, S. (2001), Revising and Extending the Units of Measure "Subontology", in *Working Notes of the IJCAI-2001 Workshop on the IEEE Standard Upper Ontology*.

Sowa, J. (2000), Knowledge Representation, Brooks/Cole, Pacific Grove, CA.

Smith, B. (1994) “Fiat Objects,” N. Guarino, L. Vieu and S. Pribbenow (eds.), Parts and Wholes: Conceptual Part-Whole Relations and Formal Mereology, 11th European Conference on Artificial Intelligence, Amsterdam, 8 August 1994, 15–23.

Smith, B. (1996) "Mereotopology: A Theory of Parts and Boundaries", Data and Knowledge Engineering, 20, 287–303.

SUMO (2002), Suggested Upper Merged Ontology, web site, http://ontology.teknowledge.com