There is probably little argument that the past decade has brought the “big bang” in the amount of online information available for processing by humans and machines. Two of the trends that it spurred (among many others) are: first, there has been a move to more flexible and fluid (semi-structured) models than the traditional centralized relational databases that stored most of the electronic data before; second, today there is simply too much information available to be processed by humans, and we really need help from machines. On today’s Web, however, most of the information is still for human consumption in one way or another.

Both of these trends are reflected in the vision of the Semantic Web, a form of Web content that will be processed by machines with ontologies as its backbone. Tim Berners-Lee, James Hendler, and Ora Lassila described the “grand vision” for the Semantic Web in a *Scientific American* article in 2001:¹ Ordinary Web users instruct their personal agents to talk to one another, as well as to a number of other integrated online agents—for example, to find doctors that are covered by their insurance; to schedule their doctor appointments to satisfy both constraints from the doctor’s office and their own personal calendars; to request prescription refills, ensuring no

NATALYA NOY
STANFORD UNIVERSITY

ORDER from Chaos

Will ontologies help you structure
your semi-structured data?

ORDER from Chaos

harmful drug interactions; and so on. For this scenario to be possible, the agents need to share not only the terms—such as appointment, prescription, time of the day, and insurance—but also the meaning of these terms. For example, they need to understand that the time constraints are all in the same time zone (or to translate between time zones), to know that the term *plans accepted* in the knowledge base of one doctor's agent means the same as *health insurance* for the patient's agent (and not *insurance*, which refers to *car insurance*), and to realize it is related to the term *do not accept* for another doctor, which contains a list of excluded plans.

Such seamless conversation between software agents that were not initially designed to work together is the Holy Grail of Semantic Web research. Regardless of whether this Holy Grail is ever completely discovered (or invented), as with any “grand challenge,” this vision drives cutting-edge research, attracting researchers from artificial intelligence, databases, information integration, data mining, natural language processing, user interfaces, social networks, and many other fields. Simply constructing pieces and components of the Holy Grail is by itself a fruitful and worthwhile endeavor that will produce many useful discoveries along the way.

To make this sort of seamless interaction between software agents possible, the agents must share semantics, or meaning, of the notions that they operate with. These semantics are expressed in ontologies, which contain the explicit definitions of terms used by the agents. These definitions are represented in languages where each construct has a formal explicit meaning that can be unambiguously interpreted by humans and machines. While there are many definitions of an ontology,² the common thread is that an ontology is some formal description of a domain, intended for sharing among applications, expressed in a language that can be used for reasoning.

Since the underlying goal of ontology development is to create artifacts that different applications can share, there is an emphasis on creating common ontologies that can then be extended to more specific domains and appli-

cations. If these extensions refer to the same top-level ontology, the problem of integrating them can be greatly alleviated. Furthermore, ontologies are developed for use with reasoning engines that can infer new facts from ontology definitions that were not put in explicitly. Hence, the semantics of ontology languages themselves are explicit and expressed in some formal language such as first-order logic.

In the past few years, the ontology has become a well-recognized substrate for research in informatics and computer science.³ The word *ontology* now appears on almost 3 million Web pages; the Swoogle crawler (<http://swoogle.umbc.edu>) indexes more than 300,000 ontologies and knowledge bases on the Web. For our purposes, we define ontology to be an enumeration of the concepts—and the relationships among those concepts—that characterize an application area. Ontologies provide an explicit framework for talking about some reality—a domain of discourse—and offer an inspectable, editable, and reusable structure for describing the area of interest. Ontologies have become central to the construction of intelligent decision-support systems, simulation systems, data-integration systems, information-retrieval systems, and natural-language systems. W3C has developed RDF (Resource Description Framework), RDF Schema,⁴ and OWL (Web Ontology Language),⁵ standard languages for representing ontologies on the Semantic Web.

Although the idea of sharing formal descriptions of domains through ontologies is central to the Semantic Web, don't assume that everyone will subscribe to only one or a small number of ontologies. A common misconception about the Semantic Web (and the reason many dismiss it outright) is that it relies on everyone sharing the same ontologies. Rather, the Semantic Web provides two advantages: first, formally specified ontology languages that make semantics expressed in these languages explicit and unambiguous and therefore more amenable to automatic processing and integration; second, the infrastructure to use ontologies on the Web, extend them, reuse them, have them cross-reference concepts in other ontologies, and so on. For example, if I am setting up my own online store, I can reuse ontologies for billing, shipping, and inventory, and extend and customize them for my own store. If someone else reuses the same inventory ontology, we could be part of the same portal that searches through both our inventories. Furthermore,

we will know exactly what we mean by the term *address* (whether it is shipping or billing address), and that *number in stock* is the number of items, not the number of crates of those items.

One of the main challenges of semi-structured data is to move away from the fairly rigid and centrally controlled database schemas to the more fluid, flexible, and decentralized model. Similarly, on the Semantic Web, there is no central control. Just as anyone can put up his or her own page on the Web and anyone can point to it (and say either good or bad things about it), on the Semantic Web anyone can post an ontology, and anyone can reuse and extend any ontology that is appropriate to his or her task. Naturally, this model brings up issues of finding the right ontologies, evaluating them, trusting the sources they come from, and so on. We discuss these issues later in the article.

PROBLEMS THE SEMANTIC WEB DOES AND DOESN'T SOLVE

How are ontologies and the Semantic Web different from other forms of structured and semi-structured data, from database schemas to XML? Perhaps one of the main differences lies in their explicit formalization. If we make more of our assumptions explicit and able to be processed by machines, automatically or semi-automatically integrating the data will be easier. Here is another way to look at this: ontology languages have formal semantics, which makes building software agents that process them much easier, in the sense that their behavior is much more predictable (assuming they follow the specified explicit semantics—but at least there is something to follow).⁶

This explicit machine-processable description of meaning is the key difference between XML and ontology languages such as RDF and OWL: In XML, some of the semantics are implicit, encoded in the order and nesting of components in the XML document; in ontology languages, the semantics are explicit, having underlying axioms and formal descriptions. To represent an order of sentences in RDF, you need to use an explicit structure, such as an RDF list, to specify the order, saying explicitly which one comes first, which one comes next, and which one is last, rather than simply positioning things in the order that you want in the serialized document. Conversely, the order in which statements appear in an RDF document is irrelevant, as an underlying structure is a graph, and it is the links between the elements that are important. In fact, it is perfectly legal for an RDF parser to read in an RDF document and write it out, with triples appearing in a completely different order in the docu-

ment—the represented model will still be exactly the same.

So, what does the Semantic Web bring to the table today, and what is it likely to bring in the near future? One of the key hypotheses behind the Semantic Web is that by making lots of ontologies—explicit and machine-processable descriptions of domain models—available on the Web, we can encourage people to reuse and extend them.

The Semantic Web infrastructure encourages and supports publication of ontologies. Hopefully, this infrastructure will encourage agents to reuse existing ontologies rather than create new ones. When two agents use the same ontology, semantic interoperability between them is greatly facilitated: they share the same vocabulary and the understanding of what classes, properties, and individuals in that vocabulary mean, and how they relate to one another. Easy access to and availability of ontologies in standard formats that the Semantic Web infrastructure provides should enable and facilitate such reuse.

In addition to providing the infrastructure and social pressure to share domain models, recent developments in Semantic Web standards provide other key components that facilitate semantic interoperability: standard languages and technical means to support semantic agreement.

W3C recommendations for RDF, RDF Schema, and OWL established a set of standard XML-based ontology languages for the first time. While formal ontologies have existed, they all used different languages, underlying knowledge models, and formats. Having a set of standard languages for ontology interchange and reuse should also facilitate and encourage reuse of ontologies.

Semantic Web languages such as RDF and OWL provide the technical means to facilitate interoperability: the use of namespace URIs (uniform resource identifiers) in RDF and OWL, and imports in OWL, enable specific unambiguous references from concepts in one ontology to concepts in another, thus enabling reuse of ontologies and their components across the Web. For example, a user developing an application that deals with wines can declare specific wine individuals to be instances of the Wine class defined in a shared ontology elsewhere. Then, anyone searching for an instance of that Wine class on the Web will get this user's wine as a result to the query. Furthermore, language constructs in OWL allow you to relate explicitly meanings of terms in different ontologies. For example, if my ontology does not have a concept of Wine, but rather I have the concepts Red Wine and White Wine, I can explicitly declare that the union of

ORDER from Chaos

these two concepts from my ontology is equivalent to the class Wine in this other, widely shared ontology.

The Semantic Web, RDF, and OWL are not by themselves the answer for seamless interoperability. Agents need to share and reuse ontologies published in RDF and OWL rather than choose their own, to reuse them correctly and consistently, or to create correspondences between terms in different ontologies if they choose to reuse different ones. More specifically, several hurdles need to be overcome to make interoperability truly seamless:

Incorrect or inconsistent reuse of ontologies. Reusing ontologies is hard, just as reusing software code is. The Semantic Web makes it likely that people will reuse (portions of) ontologies incorrectly or inconsistently. Semantic interoperability, however, will be facilitated only to the extent that people reference and reuse public ontologies in ways that are consistent with their original intended use. For example, if an agent uses the Dublin Core property creator (<http://purl.org/dc/elements/1.1/creator>) to represent anything other than the creator of the document, interoperating with others that use the property correctly will be a problem.

Finding the right ontologies. To reuse an ontology, one needs to find something to reuse. Users must be able to search through available ontologies to determine which ones, if any, are suitable for their particular tasks. Furthermore, even though internally the Semantic Web is for machines to process, user-facing tools must be available that enable users to browse and edit ontologies, to custom-tailor their tasks, to create different views and perspectives, to extract appropriate subsets, and so on. One such tool platform—Protégé—is presented in the next section.

Using different ontologies. If two agents operating in the same domain choose to use different ontologies, the interoperability problem will still exist even if their ontologies and data are in OWL and RDF. The agents will then need to create mappings between terms in their respective ontologies, either manually or with the help

of tools. While OWL provides some constructs to express the mappings, it does not free users from the necessity of finding and declaring these mappings. Therefore, just as schema mapping and integration are crucial in the database world, the mapping problem is still very much alive in the ontology world.

In summary, to enable semantic interoperability among agents, several key problems must be addressed:

1. Users must be able to search through available ontologies to determine which ones, if any, are suitable for their particular tasks;
2. Tools and techniques must be available that enable custom tailoring of the ontologies being reused to the users' tasks, views and perspectives, and subsets appropriate for the users;
3. Just as schema mapping and integration are crucial in the database world, ontology mapping and integration will remain a serious problem, as there will be ontologies that overlap and cover similar domains, but still bring their unique values to the table.

THE PROTÉGÉ STORY

With ontologies being the key technology in the Semantic Web, ontology tools will be essential to its success. In the Protégé group at Stanford Medical Informatics (<http://protege.stanford.edu>), we have been working for more than two decades on technologies to support effective development and use of ontologies by both knowledge engineers and domain experts. Protégé is an open source ontology editor and knowledge-acquisition system that arguably has become the most widely used ontology editor for the Semantic Web. It has around 30,000 registered users, an active discussion forum, and an annual international users conference.

More important, Protégé is also a platform for developing knowledge-based applications, including those for the Semantic Web. It provides a Java-based API for developing plug-ins, and a thriving community of developers all over the world develop plug-ins for tasks ranging from different ways to visualize ontologies and knowledge bases, to using wizards that help in ontology specification, to invoking different problem-solvers, reasoners, and rule engines with the data in the ontologies, to performing ontology mapping, comparing ontology versions, importing and exporting other ontologies, and so on (<http://protege.stanford.edu/download/plugins.html>). The large user community and popularity of the Protégé system is a testament to the idea that using ontologies in software

systems is gaining popularity rapidly (Protégé gets several hundred new registrations each week).

One of the main goals of Protégé has always been accessibility to domain experts. For example, there are a number of visualization plug-ins (<http://protege.cim3.net/cgi-bin/wiki.pl?ProtegePlug-insLibraryByTopic>) that present concepts and relations as diagrams, or allow users to define new concepts and relations in a knowledge base by drawing a flowchart where nodes and edges are actually complex objects (http://protege.stanford.edu/doc/tutorial/graph_widget/).

As ontologies become more of a commodity, their collaborative development within organizations, and outside single organizations, becomes essential. Protégé supports multiuser ontology development, allowing multiple users to access an ontology server to edit the same ontology simultaneously. While Protégé does support ontology comparison and versioning (<http://protege.stanford.edu/plugins/prompt/prompt.html>), there are now frequent requests to support versioning just as seamlessly as software code (versioning is supported in modern development environments).

Through the experience in the Protégé group, we are finding more interest in the industry to use ontologies to model the IT structure of an organization and its business processes. On the one hand, many companies are realizing that a huge number of various components in their systems need to be described explicitly to understand how they relate to one another and how changes or malfunctions in one component will affect other components. On the other hand, for many companies, it is not the tools themselves that constitute their main intellectual-property value; rather, this value is in the business-process descriptions, in the company's understanding of the domain, and in their methods of performing domain analysis and generating software tools from this analysis. Formal ontologies, through their system-independent descriptions that are somewhat removed from the actual implementation, force the knowledge bearers to think and encode their knowledge in abstract terms that may be more readily used by the tools to generate necessary software.

DaimlerChrysler, for example, a member of the Protégé Industrial Affiliates Program since 2001, has developed a broad spectrum of applications for employing semantic technologies to support an improved engineering process. Their main focus is knowledge representation and management, semantic portals, parametric design, intelligent decision support, configuration, and semantic information integration. DaimlerChrysler Research and

Technology has built a framework, based on Protégé, for developing ontology-based applications customized for the engineering domain. The framework, including both a toolset and an application development methodology, has been successfully used in pilot applications for supporting different product lifecycle phases such as product design, marketing, sales, and service.

Another example of a Protégé industrial partner is Exigen Group, based in San Francisco. Exigen uses Protégé interactively to model background knowledge about the business domain. The business model and rules are used to create BPM (business process management) applications. Information extraction techniques are used to

As more ontologies become available, it becomes harder, rather than easier, to find an ontology to reuse.

develop repositories of ontologies and rules from business documents. An OWL reasoner validates the extracted knowledge for the ontology component, and a rule engine validates the rule component. Protégé is employed to correct and augment the knowledge interactively, parts of which can then be fed back as enriched background knowledge. The knowledge base describes interfaces and migration paths among Exigen document archives, thus providing the semantics of document transformations.

THE CHALLENGING ROAD AHEAD

This article has already mentioned a number of challenges facing both the Semantic Web community and any other community that attempts to apply knowledge- and data-intensive solutions on the Web scale. Here are some additional challenges.

Matchmaker, Matchmaker, Find Me a Match. The Semantic Web vision critically depends on the ability of users to reuse existing ontologies (to enable interoperability, among other things), which in turn requires that users are able to find an ontology they need. Different kinds of ontology repositories exist today: some are gen-

ORDER from Chaos

erated by crawling the Web (e.g., <http://swoogle.umbc.edu>), some are curated (e.g., <http://protege.stanford.edu>), and some allow domain experts to add their ontologies to them (e.g., Open Biomedical Ontologies; obo.sourceforge.net). These repositories, for the most part, are just places to store and retrieve ontologies, however, at best enabling simple cross-ontology search. They do not enable their users to evaluate the ontologies in the repository, to search them intelligently, to know how ontologies were used, and what other users in the field think about various aspects of particular ontologies.

As more ontologies become available, it becomes harder, rather than easier, to find an ontology to reuse for a particular application or task. Even today (and the situation will only get worse), it is often easier to develop a new ontology from scratch than to reuse someone else's existing ontology. First, ontologies and other knowledge sources vary widely in quality, coverage, level of detail, and so on. Second, in general, few, if any, objective and computable measures are available to determine the quality of an ontology. Deciding whether an ontology is appropriate for a particular use is a subjective task. We can often agree on what a bad ontology is, but most people would find it hard to agree on a universal "good" ontology: An ontology that is good for one task may not be appropriate for another. Third, while it would be helpful to know how a particular ontology was used and which applications found it appropriate, this information is almost never available today.⁷

It's the Web, Stupid. The knowledge-representation community has always dealt with somewhat small, isolated problems, and knowledge bases haven't had to interoperate a lot. Moving knowledge representation to the Web creates many unique challenges. Scalability is one. Some Protégé users already have ontologies with many tens of thousands of classes.⁸ This is well below the limit for storage. While effective retrieval of concepts and editing can be done seamlessly, and using reasoners for ontologies of this size is possible, it may be pushing the limits of effective use. One of the solutions is effective

modularization of ontologies (see the next point).

Another challenge of the Web is interoperation. No matter what the combination of the "winning"

technologies is at the end, one thing is certain: These technologies will support and significantly facilitate interoperation among different models and content.

It is unlikely that this support will be manifested by an extremely small, well-defined number of standards that everyone will use, thus essentially eliminating the main hurdles to

interoperability. Rather, it is probably a solution that would encourage such reuse and the use of a limited number of different

models in the first place, but will also support, facilitate, and effectively implement mappings and translations among them.

Black Boxes. OWL, the W3C standard ontology language for the Semantic Web, is actually a complex language. Writing ontologies in OWL, even using graphical user interfaces, is a challenging task. We need to develop special-purpose editors with only limited functionality that are easy for people to use and grasp and have many wizards and tools for development. It is probably unrealistic to assume that many noncomputer scientists, without special training, will be able to develop full-fledged ontologies in OWL, with logical expressions, universal and existential restrictions, and so on.

If the Semantic Web vision is to succeed, however, they won't have to. One might argue that most of the ontologies on the Semantic Web would be simple hierarchies with simple properties, rather than ontologies using different types of restrictions, disjointedness, and complex logical expressions with unions and intersections. There is a place for both, and a small number of well-developed and verified ontologies will be necessary. At the same time, people should be able to reuse them as "black boxes" without having to understand much about them. If I reuse a well-established ontology of time, such as OWL-Time (<http://www.isi.edu/~pan/OWL-Time.html>), all I should have to know is that if I say that you order food before you pay for it, using the notion of *before* from the OWL-Time ontology, my system would be able to figure out the temporal relations between the events.

Trust Me. Trust—and especially, some computable metric of it—is extremely important in any kind of setting where anyone can post machine-processable data. The problem of trust (or mistrust) is already plaguing the Web. On the Web, information is consumed mostly by

humans, who can often use their background knowledge and intuition to assess the trustworthiness of the source. Our vast background knowledge and experience, which is hard to encode, helps a lot in determining what is good and what is bad, but mechanisms are still needed to help us in these decisions, as we are not experts in every field.

At the other end of the spectrum, databases are designed largely for machine processing, but they are largely centrally controlled, and schemas are usually published (if at all) by respectable companies. A database is not something that an average blogger posts on the Web. In the free-flowing realm of the Semantic Web and other data being posted essentially by anyone to be processed by machines, these machines need to be able to determine how trustworthy the sources are. Reputation systems, certificates, and security measures will all need to be adapted to this realm.

“Just Enough” and Imprecise Answers. Historically, both reasoning in artificial intelligence and querying in databases has been about precise answers to specific questions, with few exceptions. With a much less centrally controlled, loosely coupled collection of resources that come online and go away, have questionable credentials, and use different representations and levels of precision in their representation, we must develop query and reasoning techniques that are flexible enough to deal with such fluid collections. These methods should not require precision in either specification or answers.

Furthermore, in both reasoning in artificial intelligence and query answering in databases, there is a paradigm of getting a complete answer to your question with respect to the data stored in the accessible resources. On the Web, we are used to getting answers that are “good enough” or “as good as possible,” given the time constraints and the resources available, but not necessarily the best. If I find an airfare that seems to be about the lowest that is reasonable for this time of the year, I am happy to stop my search, even if I know that a ticket that is \$10 cheaper is available somewhere on the Web. Likewise on the Semantic Web, a result that we can find quickly and effectively with the resources that are currently available is often good enough and does not have to be perfect.

None of these challenges is insurmountable, and addressing them will produce a lot of interesting advances on the way. Think of this as the history of artificial intelligence in a nutshell: Research has not produced, and may never produce, a machine that is as intelligent as a human being. But think of how many scientific and technical advances we have made while pursuing this

goal! Likewise, whether or not we achieve the Holy Grail of all machines talking to and understanding one another without much, if any, intervention from humans, we will produce many useful tools along the way.

REFERENCES

1. Berners-Lee, T., Hendler, J., and Lassila, O. 2001. The Semantic Web. *Scientific American* 284(5): 34–43.
2. Welty, C. 2003. Ontology research. *AI Magazine* 24(3).
3. McGuinness, D. L. 2001. Ontologies come of age. In *The Semantic Web: Why, What, and How*, ed. D. Fensel, J. Hendler, H. Lieberman, and W. Wahlster. Cambridge: MIT Press.
4. Brickley, D., and Guha, R. V. 1999. Resource description framework (RDF) schema specification. Proposed recommendation, World Wide Web Consortium.
5. Dean, M., Connolly, D., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D. L., Patel-Schneider, P. F., and Stein L. A. 2002. Web ontology language (OWL) reference version 1.0; <http://www.w3.org/tr/owl-guide/>.
6. A detailed comparison of ontologies and databases can be found in Uschold, M., and Grüninger, M. 2004. Ontologies and semantics for seamless connectivity. *SIGMOD Record* 33(3).
7. Noy, N., Guha, R. V., and Musen, M. A. 2005. User ratings of ontologies: who will rate the raters? In AAAI 2005 Spring Symposium on Knowledge Collection from Volunteer Contributors. Stanford, CA.
8. Golbeck, J., Frago, G., Hartel, F., Hendler, J., Parsia, B., and Oberthaler, J. 2003. The National Cancer Institute's thesaurus and ontology. *Journal of Web Semantics* 1(1).

LOVE IT, HATE IT? LET US KNOW

feedback@acmqueue.com or www.acmqueue.com/forums

ACKNOWLEDGMENTS

The author would like to thank Tania Tudorache of DaimlerChrysler and Oleg Bondarenko of Exigen Group for providing information on the use of Protégé and ontologies in their companies. Protégé is a national resource supported by grant LM007885 from the United States National Library of Medicine.

NATALYA NOY is a senior research scientist at Stanford Medical Informatics at Stanford University. She has been involved in research in ontology development for more than 10 years and has been active in the field of the Semantic Web almost since its inception. She received her Ph.D. in computer science from Northeastern University.

© 2005 ACM 1542-7730/05/1000 \$5.00