A Database Perspective on the Semantic Web: A Brief Commentary

Frank Manola
fmanola@acm.org

1 The Semantic Web as a Database

The Semantic Web [15] is an exciting evolution of the current World Wide Web, and there is much activity surrounding it. Much of what has been published about it has emphasized the potential of what might be accomplished using improved semantic descriptions of Web resources, based on ontologies and logic-based processing. However, the emphasis on logic, agent-based processing, and so on may obscure for the database community the fact that much of the essence of the Semantic Web comes from its similarity to a database. To back this up, consider the following statements by Tim Berners-Lee, the "inventor" of the World Wide Web, about his vision of the Semantic Web:

> "The Semantic Web is a vision: the idea of having data on the web defined and linked in a way that it can be used by machines not just for display purposes, but for automation, integration and reuse of data across various applications." [15]

> "The Semantic Web is a web of data, in some ways like a global database." and "Leaving aside the artificial intelligence problem of training machines to behave like people, the Semantic Web approach instead develops languages for expressing information in a machine processable form." [3]

> "The semantic web data model is very directly connected with the model of relational databases ... 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." [4]

These statements indicate that a significant focus of the Semantic Web involves simply making Web data more machine-processable, *and more like a database*. Of course, if programs are to process this data correctly, they must process it in a manner consistent with its semantics. However, *the Semantic Web vision doesn't really care how the programs acquire this capability.* For example, this capability may exist because the data uses a vocabulary (and associated semantics) that has been agreed on within a user community, and the programs are written to use that vocabulary, just as with most databases now.

This isn't to say that having more semantics explicitly described in metadata (such as ontologies), using richer languages capable of associating more "meaning" with the data, won't be of tremendous

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

importance. Obviously such explicitly-specified semantic information will be increasingly important in the Web environment, given its scale, number of data sources, and the autonomy and heterogeneity of these sources. For example, such information will be crucial in such areas as integrating and making sense of information from heterogeneous sources, optimizations of various kinds, advanced types of concurrency control, and so on. However, lots of very effective Semantic Web applications (and certainly lots of near-term ones) will not necessarily involve extensive use of machine-processable semantic information (and only a relatively small amount of the semantics of terms can actually be captured in a formal ontology [10], although even a small amount can be extremely useful in practice). As someone said on the xml-dev email group, "Semantic Web my behind! I just want to order a pizza, not have mozzarella explained to me!" It will be a tremendous job to migrate the current Web to the point where all of the data is machine-processable, and where all of the current tools we have available in database systems are available, even without machine-processable semantics. Hence, it will be important not to neglect technologies in areas other than those directly involving machine-processable semantic information, and to integrate those technologies more thoroughly into the Web environment.

So, at its core, the Semantic Web is about making the Web more like a distributed database system, making it more like a *heterogeneous* distributed database system, giving it a richer conceptual model to better describe that data, and adding additional deductive capabilities on top to make better use of all that data. These are all lines of development that have been, and still are, currently active in the database community.

Of course, moving structured database concepts into the Web environment involves some differences as well as similarities. For example, like the current Web, all Semantic Web data would be considered as part of *one big database*, unlike current database systems, in which related data is created as physically (and logically) separate "databases."

Because all this development happens in *one big database* (the Web), rather than in lots of separate ones, different individuals and user communities will use parts of this big database for different purposes, and hence different parts of this big database will use different (or overlapping terms), and will often contain inconsistent (and sometimes untrustworthy) data (just the way the Web does now, using less-structured data). Of course, the hope is that gradually the Semantic Web will migrate toward increasingly-large portions of that Web database using more consistent vocabularies and data. And technologies pioneered in work on heterogeneous database integration will be used to enable further data integration. But even without that, the Semantic Web is still a database, and people can do useful things with such data, in many cases by applying database-like operations to large collections of it, including data from different sources (and this without a single global ontology, as some commentators claim is necessary.)

In fact, people and organizations are getting real use out of some of the Semantic Web technologies now, as illustrated by some of the applications described in the RDF Primer [9].

Moreover, the structured data making up the Semantic Web will be *added* to the existing Web, not placed in an entirely separate space. This creates all kinds of possibilities for making use of both the easily-human-interpretable data on the current Web, and the easily-machine-processable data added to it on the Semantic Web, in novel combinations. This together with the fact that, since this big database is the Web, it can (security and privacy concerns permitting) be both accessed by and added to by anyone.

To deal with some of the differences involved in using database ideas in the Web environment, and in mixing database-like data with the current Web, the Semantic Web employs variants on certain database ideas. For example, from a data modeling perspective, the Semantic Web is based on the Resource Description Framework (RDF) [14], which is effectively a form of highly-normalized relational model. In this model, URIs (Uniform Resource Identifiers [2]) identify the things people want to talk about (termed "resources"). URIs can be thought of as a superset of the URLs (Uniform Resource

Locators) used in web browsers. As with URLs, different persons or organizations can independently create URIs, and use them to identify things. However, unlike URLs, URIs are not limited to identifying things that have network locations, or use other computer access mechanisms. In fact, a URI can be created to refer to anything anyone wants to talk about, including

- network-accessible things, such as an electronic document, an image, a service (e.g., "today's weather report for Los Angeles"), or a collection of other resources.
- things that are not network-accessible, such as human beings, corporations, and bound books in a library.
- abstract concepts that don't physically exist, like the concept of a "creator" property.

Given URIs to identify resources, people can then write property/value pairs describing those resources in the form of triples of **subject** (resource), **property**, and **object** (resource or literal value). This is the basic idea behind not only RDF, but also languages layered on it such as the ontology languages DAML+OIL [7] and OWL [12].

Because URIs are unique throughout the Web, one person's use of a term can be distinguished from another person's use of the same term, through the use of distinct URIs e.g.,

- *http://mysite.example.com/vocabulary/creator* **vs** *http://yoursite.example.com/vocabulary/creator*.

In addition, one person can add additional descriptive information about a resource described by another person, by using the appropriate URI as its key, and (effectively) joins can be used to aggregate this information.

The basics of this type of model have had a long history in the database community, in the form of binary relations with surrogates (the URIs) as keys, since RDF triples can also be thought of as rows in binary relations having the form **property (subject, object)**. The pros and cons of this type of model have been long-debated in the database community, although not in the Web context (see, e.g., some of the comments on "semantic" models in textbooks such as [8], and the associated references). In particular, RDF resembles Codd's RM/T model [6] in some respects, particularly since in RDF, schema/ontology information is represented in the same way as instance data, and hence mixed data and metadata queries are easily possible. RDF is also very much like the graph models used in recent work on "model management". However, the RDF / Semantic Web variant involves an important extension, in that it is grounded in the Web: the subjects, objects, *and properties* in RDF statements (in effect, the table and column names, as well as key values) are URIs, and hence have global identity. Moreover, in many designs these URIs can be dereferenced (you can use them as URLs and point a browser at them) to retrieve Web data that describes the resources. The result is an interesting mixture of relational and object-oriented ideas (since the URIs are also a form of object identifier), and of data and metadata. Moreover, compared with XML, RDF really does play the role of the "relational model of the Web", since, for example, it requires that relationships not be recorded implicitly in data structures (such as XML nested elements), but instead must be made explicit.

The variations on relational themes represented by RDF help deal with a number of differences between the Web environment and a typical database environment. For example, in RDF each triple (row in a binary table) represents a single fact, unlike a conventional relational table, in which each row represents a set of facts. In a conventional database (and in certain Web applications too), it is expected that a relatively small, fixed set of attributes will be used in describing a particular entity. However, on the Web, even though a particular schema might intend to constrain the properties that apply to a particular resource, any number of properties, from multiple sources, might be unpredictably applied to describe a particular resource (often by different people or organizations), and the set of properties might be extended at any time.

Moreover, RDF schemas are *descriptive*, not *prescriptive*. An application may choose to validate instance data against a schema, but this is not mandatory. For example, instance data may be perfectly valid either without some schema-defined properties, or with additional properties not defined in a given schema [9].

Hence, one of the things RDF caters to is the potential for highly irregular or unpredictable amounts of data about a given resource. Where structural regularity is possible (and this is likely in many applications), the RDF from a given source could be stored in a relational (or XML) database (and relational data can easily be interpreted as RDF).

And, of course, the use of URIs as names enables RDF to deal with the problem of distinguishing terms and other resources that might be referred to by many people and organizations in a single, large database.

2 Research Themes

Given the above description of the Semantic Web as a database, it should be clear that there is considerable room for applying numerous database research themes in this new context.

The papers in this issue illustrate a number of these themes, themes that are also illustrated in numerous other papers on the Semantic Web. These include both the application of Semantic Web languages to classic database research areas such as heterogeneous data integration, consideration of relatively new issues that the Semantic Web context makes particularly important, such as that of data provenance, and the use of variations on "semantic" technology in general.

In fact, considering the Semantic Web as a giant database illustrates that the whole range of database research themes, suitably adapted to the Semantic Web differences, can be usefully explored in this new context. This includes such issues as:

- From a data model perspective, to what extent does the Semantic Web truly justify a new data model (this echoes a corresponding concern within the logic/knowledge representation community at being restricted to RDF's binary relations)? Do earlier arguments about RDF-like data models still apply, and with what force? Are there alternatives to handling global naming and its ramifications, and the need to be able to mix and match terms from multiple vocabularies?
- What is the best way to integrate RDF and relational-style processing with XML? RDF has an XML syntax, but this syntax has received numerous bad reviews. Is there a better syntactic approach?
- The current Web is asymmetric with respect to reading and writing (creating and modifying Web data generally involves different tools than accessing it). What is the best way of reducing this asymmetry? In particular, although some updating protocols exist (like WebDAV), how can we get better transactional behavior for Web data, and what forms of transactions are most suitable?
- Databases have historically emphasized a distinction between logical and physical data structures. In contrast, in the Semantic Web this distinction is currently not as well developed. RDF is typically stored as a graph, rather than using more efficient representations. Can we develop more efficient processing for RDF, and still retain its ability to deal with irregular and varying data structures?

Regarding capturing "semantics" (or conceptual design), people typically have "ontologies" in mind when they design databases. We've been training them to think like this for years in studying database design. However, we haven't had a clean way (or much reason) to capture all that information in the richer models that knowledge representations provide (without using a separate tool often not linked to the database). Now we have better reasons to push this harder. The information systems community needs to adopt ontologies, make them a more explicit part of their data management strategies, and then investigate how to use them in database processing. All the prior work by the database community in richer conceptual languages and deductive databases is relevant here. At the same time, we need to make sure to point to the need to support "industrial scale" management of large amounts of definitions and content that continually change (leveraging, e.g., prior work on versioning and schema modification), and the whole design and implementation process.

Prior work by the database community on integration of heterogeneous data is clearly relevant, and needs to be pushed and extended. A major effort is required to make the process of doing this integration easier. It can't all be automated (e.g., writing "articulation axioms"), or the requirements known ahead of time, so we need to investigate how to make it easier for people to help, both at design time (as before) and *at run time* (we've got instant messaging between people; how about between an agent or query engine and a person, along the lines of "I don't understand exactly what 'sale price' consists of; can you clarify?"). Such an intervention might be quite reasonable in the context of some types of Web applications.

Work on integration also needs to consider how to use the Web (and its technologies) as a potential organizing basis and environment or infrastructure for information sharing (i.e., use of the Web as a *tool* for solving data management problems, not just as the *object* of data management technologies). A key observation here seems to me that part of solving the data integration problem is providing a data integration environment that enables all the pieces to be put together (and, for example, enables people to have access to and potentially reuse metadata and schema designs). The Web is a widely-available shared environment which can make this happen (the development of shared repositories for XML DTDs and schemas is an example of this idea in action).

Finally, a lot more work needs to be done on the various notions of "contexts" [11, 13, 1]. This involves more than just recording and using (e.g., for mediation) metadata about the definitions and assumptions used by different data collections. It also involves being able to integrate information described using languages of different descriptive capabilities, as well as being able to integrate different "reasoning engines" that operate on that information, on the same Web, and have them interoperate. The Web allows people with different reasoning capabilities to interoperate (up to a point!), and this needs to be a goal for our systems as well. The "tradeoff between expressiveness and computability" is frequently cited in discussing the use of deductive machinery in the Semantic Web. We need to be prepared for the fact that people will make different decisions about this tradeoff (e.g., making use of more powerful deductive machinery on relatively limited *subsets* of the Semantic Web, and making use of more efficient machinery on problems of larger scope), and try to cope with all of them.

3 Some Observations

While there is an awful lot of "hype" about the Semantic Web, there is also an awful lot of what might be termed "negative hype" as well; that is, discussion that is equally "hypish", but negative. This is based largely on the use of the word "semantic" in Semantic Web, and the negative visions people sometimes associate with it. (See, e.g., [5] for additional discussion on this subject.) I personally think the Semantic Web has lots of promise (I've been involved with the W3C Semantic Web Activity for the past 3 years, and I don't feel like a stoker on the Titanic!). Of course, I'm less hopeful for some of the more grandiose visions that "Semantic Web" sometimes calls to mind, but we have to wait and see. Part of the problem is people who seem to make the unstated assumption that we have to reason the following way:

1. For the Semantic Web to be really useful and successful, it has to achieve all the grandiose visions

anyone has ever included in a "futures" paper about it.

2. The Semantic Web will never achieve those grandiose visions.
3. Therefore the Semantic Web will never be really useful or successful.

This strikes me as being a dubious argument, precisely because, whether or not #2 is true, #1 isn't. But some people seem to want to avoid thinking about what a "success criterion" would be for the Semantic Web (or how they would know a Semantic Web if they saw one, for that matter).

Just because someone can't do impossible thing X with all this data doesn't mean someone else can't do useful thing Y with it. And perhaps X isn't impossible after all. Or, if it's impossible on the Web as a whole, it isn't impossible (and is even useful) on some subset of the Web. Most people, after all, aren't interested in the entire Web. They're interested in some particular part of it, containing the data that's important to them. If they can work out how to structure it, and develop a common vocabulary (or even an ontology), that's terrific, and we should be encouraging them. And they'll be mainly using techniques explored first in the database community.

If the Web simply managed to provide an environment in which people could hook up existing databases, work out agreements on the terms, and define (say) synonym relationships between some of the attributes, and this became the basis of lots of useful data interchange, would that be "unsuccessful" because it only involved, say 10% of the data on the Web (what percentage of the world's data is in databases of any kind at the moment, he asked rhetorically)? 10% (or even some lesser number) sure sounds like a success to me!

Finally, I think it's important to emphasize that this is *the Web* we're talking about: the largest distributed system in existence, and the largest distributed database in existence. It's accessed, *and contributed to*, by thousands and thousands of individuals and companies, and it implicitly contains all database data (as well as all the services) that can be accessed through it. This represents a grand challenge for information systems technology if I ever saw one! Even without advanced semantics processing, the Semantic Web is an evolution of this existing Web that will provide a database in which programs can do useful things whose scope is hard to imagine. Even though this evolution will happen to different extents, at different rates, in different parts of the Web, over time, it not only will happen, it is happening right now. So, in my view, when people sometimes refer to the Semantic Web as "a vision", they seriously understate the case: "an emerging reality" would be more accurate.

References

[1] T. Berners-Lee. Conceptual graphs and the semantic web, 2001. http://www.w3.org/DesignIssues/CG.html.

[2] T. Berners-Lee, R. Fielding, and L. Masinter. Uniform Resource Identifiers (URI): Generic Syntax. http://www.isi.edu/in-notes/rfc2396.txt, Aug. 1998.

[3] T. Bernes-Lee. *Semantic Web road map*, Sept. 1998. http://www.w3.org/DesignIssues/Semantic.html.

[4] T. Berners-Lee. *What the Semantic Web can represent*, Sept. 1998. [also appears as "What the Semantic Web is not"] http://www.w3.org/DesignIssues/RDFnot.html.

[5] M. H. Butler. Is the semantic web hype? http://www-uk.hpl.hp.com/people/marbut/isTheSemanticWebHype.pdf.

[6] E. F. Codd. Extending the database relational model to capture more meaning. *ACM Trans. Database Systems*, 4(4), Dec. 1979.

[7] D. Connolly, F. van Harmelen, I. Horrocks, D. M. P. Patel-Schneider, and L. Stein. DAML+OIL (March 2001) Reference Description. W3C, Dec. 2001. http://www.w3.org/TR/daml+oil-reference.

[8] J. Date. *An Introduction to Database Systems*. Addison Wesley Longman, Inc., 2000.

[9] F. Manola and E. Miller (editors). RDF Primer, Oct. 2003. http://www.w3.org/TR/rdf-primer/.

[10] N. Guarino and A. Persidis. Evaluation framework for content standards, 2003. http://ontoweb.aifb.uni-karlsruhe.de/Members/ruben/Deliverable%203.5.

[11] R. V. Guha. *Contexts: A Formalization and Some Applications*. PhD thesis, Stanford University, 1991. http://www-formal.stanford.edu/guha/index.html.

[12] M. Dean, G. Schreiber (Editors); F. van Harmelen, J. Hendler, I. Horrocks, D.L. McGuinness, P.F. Patel-Schneider, L.A. Stein (Authors). OWL Web Ontology Language Reference. World Wide Web Consortium, Aug. 2003. http://www.w3.org/TR/owl-ref/.

[13] J. McCarthy and S. Buvac. Formalizing context (expanded notes). http://www-formal.stanford.edu/jmc/mccarthy-buvac-98/index.html.

[14] W3C (World Wide Web Consortium) RDF Core Working Group page. http://www.w3.org/2001/sw/RDFCore/.

[15] World Wide Web Consortium (W3C). *W3C (World Wide Web Consortium) Semantic Web Activity page.* http://www.w3.org/2001/sw/.