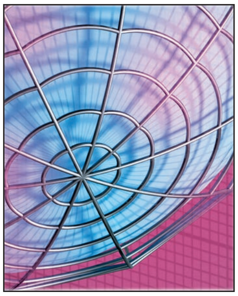

Delivered Deliverables

The State of the Semantic Web, Part I

Danny Ayers • Talis

In this installment of Webscience, I'll look at some of the tangible products coming out of the Semantic Web initiative. In a presentation entitled, "The State of the Semantic Web," Ivan Herman, Semantic Web lead at the W3C (www.w3.org/People/Ivan/CorePresentations/State_of_SW/), described the progress that has been made in regards to specifications, tools, and applications. Ivan and I were both members of the W3C's Education and Outreach group, in which the question of progress was ever present. Ivan's material is informative, and I don't question his analysis. But it was assembled by just one person, so I felt it would be open to further exploration.

I asked the Semantic Web Interest Group (via the semantic-web@w3.org mailing list) for personal opinions on the Semantic Web's present state. Various people responded on- and off-list, and a few even blogged on the question. This yielded more material than I can fit in a single column, so for this installment, I'll stick to facts that are easy to point to, moving on to a more opinion-based discussion in a later issue. What constitutes the Semantic Web itself is open to debate, so I'll focus on aspects that clearly fall under the W3C's initiative, starting with relevant specifications.

Semantic Web Specifications

The first published specification that was unarguably for a Semantic Web technology was "Resource Description Framework (RDF) — Model and Syntax," which appeared as a working draft in 1997 before it became a recommendation in 1999. But it wasn't until 2004 that a reasonably solid suite of core specifications was published for RDF as well as the Web Ontology Language (OWL). Only in the past year or two have we seen a comparable flurry of specification work, this time building on the 2004 foundations to fill in requirements from various communities. (You can find the W3C's Semantic Web specifications via www.w3.org/2001/sw/.)

HTTP and URIs

The Semantic Web is an extension of the traditional Web, and hence uses URIs to identify resources and HTTP to access those resources. Even this level has seen progress from a Semantic Web perspective. The question of how HTTP should deal with identifiers for things that aren't on the Web in the traditional sense (people or cars, for example, rather than documents) was finally addressed in 2007 with a Technical Architecture Group (TAG) finding — essentially, that the HTTP server should provide a redirect to a document describing the resource.

RDF and RDFS

RDF provides the underlying model for other Semantic Web technologies. Like the Web, it's essentially a node and arc-graph model based on URIs, but it usually expresses data as a set of three-part statements, lending itself to treatment using relational techniques. RDF Schema (RDFS) provides a basic language for creating vocabularies we can use to actually describe resources.

Although RDFS specifications are well established, usage patterns have changed over the years, with a few standouts. Developers often avoid the official RDF syntax, RDF/XML, in favor of Turtle (described later), which requires human legibility. RDF reification aimed to make statements about statements, but it doesn't really do this in the manner expected, so today most practitioners use named graphs (which identify a set of statements with a URI) instead. Finally, the Semantic Web community recognizes RDF containers (`rdf:Bag`, `rdf:Seq`, `rdf:Alt`) as se-

1089-7801/09/$25.00 © 2009 IEEE

mantically troublesome, and prefers RDF collections (`rdf:List`).

OWL

OWL appeared at the confluence of RDFS and work on formal logic, specifically *description logics*. It provides terms that enhance RDFS's ability to create vocabularies and lets them exist as formal ontologies with associated inference capabilities.

SKOS

The Simple Knowledge Organization System (SKOS) is, strictly speaking, a vocabulary built on RDFS and OWL, with specifications still in development. However, it shares similar application areas as those other languages. SKOS lets us express classification systems such as taxonomies and thesauri in the RDF model when RDFS and OWL's logical strictures (used directly) might be too strong. It offers a straightforward migration path from existing knowledge organization systems to Semantic Web technologies.

SPARQL

Before we could implement RDF stores as databases comparable to traditional relational databases, we needed a missing piece of the puzzle: a query language. The W3C resolved this in January 2008 with the SPARQL Protocol and RDF Query Language recommendation. The ability to find and project data of interest in a store using a standard declarative language liberates developers from having to hard-code against a given store implementation's API. SPARQL's syntax is relatively intuitive and similar enough to the Structured Query Language (SQL) that developers familiar with traditional databases will have little difficulty seeing how it works. SPARQL has limited features compared to something like SQL, but the SPARQL working group felt it was better to publish the language in this form as soon as possible. A subsequent version will likely appear after a year or two and add any other features that this version's deployment in the wild demonstrates are required.

One little gem SPARQL brings to the table is a description of named graphs, circumventing the need for a separate specification.

RDFa

A long-running issue surrounding RDF was that it was technically infeasible to embed RDF data directly into XHTML using RDF/XML. Meanwhile, the microformats initiative was gaining popularity, offering a way to express data in HTML based primarily on conventions for values of existing attributes. Wherever possible, microformats follow existing, well-deployed data models. For example, we can use microformats such as hCard and hCalendar (see http://microformats.org) to incorporate personal information (derived from the vCard standard[1]) and calendar information (derived from the iCalendar standard[2]). This approach offers several desirable characteristics — in particular, it follows the DRY (don't repeat yourself) principle. The embedded data is the same as the human-readable information rendered in the browser. With one or two caveats, this data is also extractable as RDF. However, microformats don't offer a generic solution to RDF-in-HTML (the microformat-like eRDF specification does, but the standards bodies somehow overlooked it), and until recently, no clear path existed to creating arbitrary data.

Enter RDFa, which uses standard HTML features in more or less the same DRY manner as microformats but also exploits the XHTML Modularization specification (www.w3.org/TR/xhtml-modularization) to extend HTML with five new attributes: `about`, `property`, `resource`, `datatype`, and `typeof`. This combination lets us express arbitrary RDF data within HTML documents. To an RDFa parser, the HTML document is RDF, whereas to a regular HTML tool, the document is HTML. The specification became a W3C recommendation in October 2008 and is already attracting several publishers (for examples, see the RDFa blog at http://rdfa.info/).

GRDDL

We can (potentially) express a significant proportion of the world's data in XML documents. The Gleaning Resource Descriptions from Dialects of Languages (GRDDL) specification provides a set of processes that will automatically interpret these documents as RDF. Other approaches are possible, but the key at present is applying an XSLT to translate the original XML into an RDF/XML representation. XHTML (such as microformats) uses metadata profiles (www.w3.org/TR/html401/struct/global.html#h-7.4.4.3) to indicate the appropriate transformation. (Dan Connolly, chair of the GRDDL working group, proposed expressing RDF in HTML in a microformat-like manner long before anyone conceived of microformats.)

One Web-oriented feature of GRDDL is that it uses the "follow your nose to find more information" approach. For example, it can use XML namespace documents to indicate the required transformation. When a GRDDL-aware agent encounters an XML document with an http: scheme namespace URI, it can follow its nose and retrieve the namespace document. If that document follows the simple GRDDL conventions to supply the XML-to-RDF/XML transformation, the agent can obtain the RDF corresponding to the original document. This means that once the namespace document is suitably equipped, with one stroke every XML document that uses that namespace is available as RDF, without any modification of the individual documents.

Turtle and N3

The Notation 3 (N3) language began life as Tim Berners-Lee's human-friendly tool for "noodling" with Semantic Web logic and data, along with the cwm engine, in effect a Python implementation of N3. Although N3 itself can seem esoteric, a subset of the N3 syntax maps directly to RDF and makes a considerably more human-friendly notation than RDF/XML. Dave Beckett of Yahoo coined the name Turtle for this subset, and today virtually all RDF toolkits support the serialization. These syntaxes are currently specified only as W3C team submissions, but SPARQL's pattern description is, in effect, Turtle with variables — and Turtle is popular compared to RDF/XML — indicating that they're already fixtures in the Semantic Web interior design.

A common attitude is that RDFS or an OWL ontology have greater significance than any document containing merely RDF instance data.

In the Pipeline

Certain specifications are well on their way to becoming W3C recommendations and are worth noting

POWDER

Designed well over a decade ago, the Platform for Internet Content Selection (PICS; www.w3.org/PICS) specification enabled Web publishers to associate metadata with content. Access control (as applied to children, for example) was the main motivation, although its potential applications were broader. Following input from parties such as the Dublin Core Metadata Initiative, a next-generation PICS was proposed — RDF.

The Protocol for Web Description Resources (POWDER) specifications in many ways return to PICS's roots, reworked firmly on the Semantic Web stack. They allow, for example, site-wide labeling of resources through the (regular expression) matching of strings in site URIs. Like PICS, access control is a significant use case, although the description of resource groups that POWDER enables has a wide range of applications.

OWL 2

As the name might suggest, this is the proposed next version of OWL, with various additional features that the OWL community has requested. At publication time, 12 specification documents were nearing completion.

RIF

RDFS and OWL are very much logic languages, and reasoning across ontology and instance data, at whatever level of sophistication, is a key feature. But quite a chasm exists between them, and much of the logic is found elsewhere, such as the business logic within enterprise knowledge bases. The Rule Interchange Format (RIF) working group aims to devise a common format to enable mapping between rule languages, which should help expose a significant proportion of currently "dark" material on the Semantic Web.

Vocabularies and Ontologies

One feature of the existing Web is that it enables distributed publishing and — similar to open source software — independent development. All HTML document files are considered (more or less) equal according to both the producer and consumer, that is, the HTTP server and client (browser). The Semantic Web inherits this approach, but independent development in this context brings with it certain issues.

A common attitude is that RDFS or an OWL ontology have greater significance than any document containing merely RDF instance data. This attitude is justified even though HTTP treats all such documents on the Web the same. Schemas can offer both a connection to human concepts (through human-language term descriptions and annotations) and a way to glue together disparate sets of instance data using common class and property definitions.

Reuse of existing vocabularies or ontologies enables interoperability for no cost — after all, we're all speaking the same language. A growing selection of well-known vocabularies — although generally designed with a single information domain in mind — contain terms that are reusable across a range of applications (Friend of a Friend [FOAF], Dublin Core, and so on). If, for example you need to model a person within your application, you'll likely find most of the specification work you need in FOAF: simply use the term `foaf:Person` and its associated properties.

Mechanisms exist for locating such existing schema (notably, Semantic Web search engines). Unfortunately, people still commonly duplicate terms found elsewhere. While you're building the data model for an application, it's convenient to invent required terms in a new namespace, and once your application is up and running, you have little motivation to replace new terms with equivalents from existing vocabularies. Although we can achieve less (locally) intrusive interoperability by using RDFS/OWL terms to map from the new terms to existing ones, this might still seem like extra work for no obvious gain. Note that third parties can create and publish such mappings independent of the publishers of either the new terms or the already deployed vocabularies.

Many factors influence vocabulary use, creation, and maintenance. The desire for academic kudos or commercial interests could lead an organization to reinvent so they can claim to have their own vocabulary. Some issues boil down to whether the potential user of a particular vocabulary trusts the vocabulary's publisher. The W3C has long debated whether it would be productive to create or adopt vocabularies and give them an "official" mark of approval (presumably along with a commitment to maintain the material appropriately). One or two near-precedents exist (work surrounding the vCard vocabulary, for instance), but the ongoing consensus seems to be that the cost in terms of implying that this kind of quasi-centralization is desirable outweighs the benefits. The architectural aim is distributed vocabulary development, although developers often overlook the ease with which this is possible (in principle, at least — modeling itself is rarely easy).

We can thus argue that the Semantic Web doesn't need a centralized, official vocabulary agency — in fact, such a thing would be something of an anathema (although repositories that index or cache vocabularies from distributed sources are a different matter). Domain experts and developers should create and publish the vocabularies they need. Attitudes that some centralized agency is necessary persist, but vocabulary work on the Web at large is progressing reasonably well.

Deployment Areas

So, we have numerous specifications and the tools (such as RDF stores, which I'll discuss more in the next column) with which to implement them. But where are they actually deployed?

We can find Semantic Web technologies within traditional industries such as oil and gas, defense, e-government, and financial services. The life sciences are a particularly active sector (the W3C has a Health Care and Life Sciences group), and given RDF's roots in the document metadata world, libraries and related services have unsurprisingly adopted such technologies as well.

One possible indicator of the Semantic Web's state is how it's reflected offline. Researchers have written and presented copious amounts of papers at various conferences. Academia is an obvious avenue for research and publication, but events such as the Semantic Technologies conference series (www.semantic-conference.com) also have the Semantic Web in scope and are oriented toward commercial applications (the first topic area they list is "Industry Trends, Market Outlook, and Business and Investment Opportunities"). Company whitepapers on the Semantic Web are easy to find.

When it comes to books, they range from the academic to hands-on practical development (see http://esw.w3.org/topic/SwBooks). It's hard to imagine the effort required to write *Practical RDF* (O'Reilly) back in 2003, given the need to explain RDF/XML syntax and the language's questionable features, such as reification. On the other hand, although *Semantic Web for the Working Ontologist* (Morgan Kaufmann, 2008) features Turtle syntax, its scope is largely restricted to modeling, which shows how the field has expanded. In terms of outreach to a broader community, the forthcoming *Semantic Web for Dummies* (Wiley) speaks for itself.

Regarding outreach, I'm obliged to mention one tangible result of the W3C Education and Outreach group's activities: a new logo for the Semantic Web initiative, which has found its way onto stickers, t-shirts, and keyrings (see www.w3.org/2007/10/sw-logos.html).

So far, I've covered only a few aspects of the state of the Semantic Web, so I'll return to the topic soon. But as a preliminary conclusion, I'd like to offer the following personal observation. Roughly three years ago, I started compiling a list of notable developments for a blog feature entitled, "This Week's Semantic Web." Unfortunately, I didn't have the time to maintain it for long, but a year or so ago I restarted the effort (http://blogs.talis.com/nodalities/category/this-weeks-semantic-web). Recently, I had to give up again, this time because there were simply too many developments going on to realistically keep track of. We might be seeing only hints of the Semantic Web's future potential, but enough work is going on in the field to suggest that the future's not far off.

References

1. F. Dawson and T. Howes, *vCard MIME Directory Profile*, IETF RFC 2426, Sept. 1998; www.ietf.org/rfc/rfc2426.txt.
2. F. Dawson and D. Stenerson, *Internet Calendaring and Scheduling Core Object Specification (iCalendar)*, IETF RFC 2445, Nov. 1998; www.ietf.org/rfc/rfc2445.txt.

Danny Ayers works for Talis as a developer and community liason for its Semantic Web platform (http://talis.com/platform). His blog is at dannyayers.com. Contact him at danny.ayers.ieee@gmail.com.