Editor: Charles Petrie • petrie@stanford.edu

Semantic Web and Semantic Web Services

Father and Son or Indivisible Twins?

Martin Hepp • *Digital Enterprise Research Institute (DERI), University of Innsbruck*

The Semantic Web is, without a doubt, gaining momentum in both industry and academia. The recent International Semantic Web Conference (ISWC) attracted more than 500 researchers; major vendors including IBM, Oracle, and Software AG have released or announced products; and the forthcoming Semantic Technology Conference in San Jose, California, is poised to be an impressive showcase for executives and venture capitalists on the business potential of semantic technologies. Unfortunately, Semantic Web services — annotating computational functionality rather than data — are underrepresented on the agenda, at least if we take the number of scientific publications about Semantic Web services as a proxy. Indeed, they're widely regarded as the "ugly stepchildren" while most Semantic Web researchers dedicate their attention to annotating Web content stored in static documents or database-driven applications.

In the January/February 2006 installment of Peer to Peer, Rob McCool proposed a very lightweight approach to making the Semantic Web a reality — mainly by adding some extra tags to existing Web content.[1] Although that might work for a small part of the Web, annotating existing Web data won't make the original Semantic Web vision a reality. Instead, evidence shows that Semantic Web services (SWS) frameworks are mandatory components of the Semantic Web, primarily because entities are more willing to expose functionality than data in business settings.

Revisiting Semantic Web Myths

Many assume that we can realize the Semantic Web by gradually augmenting existing data (mainly HTML and XHTML) via ontological annotations derived from today's human-readable Web content. Next-generation Web search engines should then be able to use this machine-readable metadata to improve precision and recall, and intelligent applications would be empowered to extract and recombine information found on the Web. This mindset, however, is flawed because it's based on several myths.

The Needle-in-the-Haystack Assumption

First, the common assumption that "everything is on the Web, but we just can't find what we need" is not true. In a recent representative sample of Web content in the Austrian tourism domain, we collected striking evidence that the amount of information on Web resources was insufficient to find and rank accommodations — at least, if we use the complete set of registered accommodations as the reference.[2] Only 7 percent of vendor-operated sites offered room-availability information, which is the most important fact when searching for a suitable offer; even among tourism portals that support availability checks and booking from the technical side, only 21 percent of the accommodations give availability data. The remaining 79 percent require a potential guest to either call or communicate via email to check availability. In other important information categories such as room features, star rating, or available technical equipment, we found similarly weak coverage. At least half the sites covered only 7 of 16 typically relevant categories in sufficient detail for decision-making. In other words, even perfect annotation of existing Web content would fail to make the Semantic Web a reality in this arena. Although tourism is just one small application domain, researchers have naturally identified it as an ideal showcase because of its information heterogeneity, market fragmentation, and rather complex discovery and matchmaking tasks, including substitution and composition — all of which are limitations that Semantic Web technologies promise to overcome.[3,4]

The Business Web Is Not Stateless

Persistent information publication is a core Web design principle. A fully compliant Web application shouldn't change its internal state in response to an http read access of an available

 1089-7801/06/$20.00 © 2006 IEEE Published by the IEEE Computer Society

resource, but many Web applications ignore this constraint. In fact, modern e-business systems often wouldn't work unless they did (when I buy the Mona Lisa on eBay, for example, it should be gone, and the former offer should be no longer visible for others).

In the business world, almost nothing is stateless. Competition for scarce resources is a core paradigm in a market economy, and concurrency conflicts naturally occur with operations on the information space that represents this economy. If 10 people search for a flight from Boston to Los Angeles, the eleventh person's request is affected by the airline's knowledge that demand is high enough for it to offer the remaining seat without discounting the rate. The whole airline industry relies on yield-management systems that do just such computations, and you can bet that your click stream through a Web shop that uses dynamic pricing will affect the final offer. Some online shops take into account your IP address, location, and even the time of day when determining what products and prices to display.

In other words, a request for price and availability isn't a mathematical function like

```
f(goal, preferences) ->
matching_offers[],
```

because we can't assume that two requests with identical goals and preferences will return the same set of offers. In such a scenario, any data-centric annotation will fail because the data — even if identified by a unique, session-ID-like uniform resource identifier (URI) — expires soon after it's published. We're used to assuming that offers consist of discrete alternatives and stable list prices. A price, however, isn't a static property of a product but rather a context-bound result of interactions between market participants, and a wealth of economics research exists on how asymmetric information distribution affects the price of goods.

We can, of course, make any piece of information a first-order object on the Web by assigning a URI for each query result. Yet, that doesn't free us from providing a means for discovering functionality that can transfer us from state A to state B — for example, a service that returns an offer, identified by a URI, for a given description of a goal — because the results aren't precomputed chunks of data and so can't be published until the respective request has been initiated. Thus, the idea of Triplespace Computing[5] — using persistent publication of triples for machine-to-machine communication — as an alternative communication paradigm to message exchange is orthogonal to the problem of describing data versus describing invokable functionality.

Entities are more willing to expose functionality than data in business settings.

Annotation of Data vs. Annotation of Functionality

Work already exists on annotating dynamic Web content,[6,7] but the fact that results to queries for availability and price aren't a functional value to this input isn't the same as whether a Web site is based on static HTML/XHTML pages or dynamic Web pages (PHP, active server pages, and so on) that are generated on the fly via a background database. Including database content as Semantic Web data isn't the same as including content that must be accessed via business functionality and that's not guaranteed to be repeatable. If the result to a request is valid only in the context of that request (an expiring offer for a flight ticket, for example), annotating the application in a way that makes all internal data appear as if it were static won't help. Also, although we can build wrappers to annotate many Web applications' functionality, annotating the data inside is often impossible because discovery and matchmaking are hidden inside the system. In such scenarios, the only viable solution seems to be to declaratively describe which goal a given function can fulfill, what state is required prior to invoking the function, and how the invocation will affect the state of the world — that is, its post-conditions. This is exactly what Semantic Web services frameworks, such as the Web Service Modeling Ontology (WSMO), OWL-S, or the Semantic Web Services Language (SWSL), offer. The SPARQL protocol,[8] which will provide a standard query interface to Resource Description Framework (RDF) databases, can also be regarded as a simplistic framework for exposing functionality, albeit limited to database queries.

Data annotation is also problematic from a practical perspective: if tools such as Human Language Technology (HLT) can perform it automatically, the question arises whether we should add annotations to data at all, given that we could apply the same HLT at data-consumption time. Manual annotation, on the other hand, is slow, costly, and can become inaccurate if an annotator fails to update it when human-readable content changes. In this sense, annotation violates the "one fact in one place" paradigm, which has contributed so much to data consistency since E.F. Codd introduced it.

The True Complexity of Matchmaking

In imperfect markets, revealing information is an important strategic action. For example, a hotel might

Table 1. Frequency of terms related to "Semantic Web" vs. "Semantic Web services" in Web documents and scholarly works.

Query	Google	Query	IEEE Xplore
"Semantic Web"	15,300,000	('semantic web'<in>metadata)	670
"Semantic Web services"	328,000	('semantic web services'<in>metadata)	65
OWL ontology	808,000	('owl' and 'ontology'<in>metadata)	268
"OWL-S" ontology	68,200	('owl-s' and 'ontology'<in>metadata)	67
"OWL-S" Web services	89,200	('owl-s' and 'web services'<in>metadata)	131
SWSL Web services	12,600	('swsl' and 'web services'<in>metadata)	4
WSMO	108,000	('wsmo'<in>metadata)	4
WSMO Web services	45,600	('wsmo' and 'web services'<in>metadata)	24
WSMO ontology	41,300	('wsmo' and 'ontology'<in>metadata)	13

not want to publicly acknowledge that it has few bookings for a given date because that information would give bargaining power to potential guests. Market participants generally also try to disclose information only to seriously interested customers. In addition, they might quote prices based on inferences about potential guests' willingness to pay. Insurance markets are a typical example of symmetry in discovery: not all possible contracts and rates are available (or even visible) to everyone. Again, the querying party's properties affect the offer set. Among others, IBM's Yigal Hoffner has done a lot of work in this area.[9]

All too often, Semantic Web research regards matchmaking as a query to a static set of available options. If you're not convinced that this view is too simplistic, consider mating as a typical example of symmetry and matchmaking's iterative nature. Mating is symmetric because an individual's availability is visible only if the potential mate meets several criteria, and the visibility of characteristics might equally depend on whether the other party meets specific criteria ("I show that I am rich only if you are beautiful," for example, or higher-order expressions such as "I don't want to be visible for others who want to be visible only for someone who is rich"). Mating is iterative in that we learn about the option space by analyzing our initial query's result set, and might restrict or weaken our requirements and preferences in response. The same pattern is evident throughout the business world: wholesalers' offers are unavailable to consumers, rebates for state employees are hidden from others, and so on. Even the fact that these options exist is often invisible rather than an openly declared precondition.

The symmetry and strategic aspects of revealing information are fundamental patterns in business interactions rather than just additional complexity that we can easily abstract from. As a consequence, developing a Semantic Web that requires data to be persistently published to an unknown audience might improve the Web, but it would virtually exclude e-business applications, despite their common use as proof of relevance in numerous papers on the Semantic Web.

No Semantic Web without Services

Exposing functionality in the form of Web services is generally more attractive for market participants than publishing all relevant facts directly on the Web. To turn the Web into the Semantic Web will require a move beyond the data-centric approach of annotating information on Web pages to annotating exposed functionality in Semantic Web services technologies. This will necessitate a substantial shift as the Semantic Web services research community is currently much smaller than the general Semantic Web research community. For example, Google Scholar returns 19,200 scientific documents for a search on "Semantic Web" compared to just 1,820 for "Semantic Web services." Table 1 further amplifies this fact with a comparison of related terms in queries through Google and the IEEE Xplore digital library.

As I mentioned, even perfect annotation of existing Web content would be insufficient to enable the Semantic Web vision, as long as the annotation is limited to persistently published information. The problem isn't just the lack of machine access to Web content but rather the lack of content itself, except as encapsulated in back-end systems or managed portals that expose only well-defined functionality and limited Web access to internal databases. With no reason to assume that the encapsulation of information inside systems will decrease, I believe that the Semantic Web must include annotation of functionality through Semantic Web services technologies such as WSMO, SWSL, or OWL-S. I'm also convinced that it's possible to describe SPARQL endpoints using something like WSMO and thus embed this promising approach into a more generic Semantic Web services framework.

As a first step, Semantic Web

researchers should reconsider some rather naïve assumptions about market participants' willingness to persistently reveal information to a general audience. For example, no sane business will publish its full inventory data to the general public.

As far as Semantic Web services are concerned, we should think about whether fully automated discovery, composition, and orchestration is a realistic scope, or whether more lightweight approaches are appropriate. Semantic Web services can mean a lot more than AI-minded automation of discovery and composition. Perhaps clever human–machine team approaches with mature tooling support will be much more relevant than "magic," fully mechanized solutions that operate under constraints that can hardly be met outside the laboratory regarding the underlying ontologies' consistency and reliability. Quite appealing is that both can likely fit well into a single comprehensive representational framework for exposing and finding functionality on the Web, such as WSMO.

References

1. R. McCool, "Rethinking the Semantic Web, Part 2," *IEEE Internet Computing*, vol. 10, 2006, pp. 93–96.
2. M. Hepp, K. Siorpaes, and D. Bachlechner, "Towards the Semantic Web in E-Tourism: Can Annotation Do the Trick?" work in progress, 2006; available at www.heppnetz.de/publications.
3. H. Werthner and S. Klein, *Information Technology and Tourism: A Challenging Relationship*, Springer, 1999.
4. H. Werthner and F. Ricci, "E-Commerce and Tourism," *Comm. ACM*, vol. 47, 2004, pp. 101–105.
5. R. Krummenacher et al., "WWW or What Is Wrong with Web Services," *Proc. 2005 IEEE European Conf. Web Services* (IEEE ECOWS 05), 2005, pp. 235–243.
6. L. Stojanovic, N. Stojanovic, and R. Volz, "Migrating Data-Intensive Web Sites into the Semantic Web," *Proc. ACM Symp. Applied Computing* (SAC 02), 2002, pp. 1100–1107.
7. H. Song, S. Giri, and F. Ma, "Data Extraction and Annotation for Dynamic Web Pages," *Proc. 2004 IEEE Int'l Conf. e-Technology, e-Commerce, and e-Service* (EEE 04), 2004, pp. 499–502.
8. K.G. Clark, ed., "SPARQL Protocol for RDF," W3C working draft, 14 Sept. 2005; www.w3.org/TR/rdf-sparql-protocol/.
9. Y. Hoffner and S. Field, "Transforming Agreements into Contracts," *Int'l J. Cooperative Information Systems*, vol. 14, 2005, pp. 217–244.

Martin Hepp is a senior researcher at the Digital Enterprise Research Institute (DERI) at the University of Innsbruck, Austria, where he leads the Semantics in Business Information Systems research cluster. He created eClassOWL, the first industry-strength ontology for products and services, and is currently working on using Semantic Web services technology for business process management. Hepp has a Master's degree in business management and business information systems and a PhD in business information systems from the University of Würzburg, Germany. Contact him at mhepp@computer.org; www.heppnetz.de.