THE

CIO's Guide to Semantics v3

By Dave McComb, President
Semantic Arts, Inc.

Published May 27, 2010
By SemanticUniverse.com

Contents

A Semantic Arts White Paper
11 Old Town Square, Suite 250
Fort Collins Colorado 80524
+1 (970) 490-2224
www.semanticarts.com

A Note from the Author

I'm pleased to offer this third edition of "The CIO's Guide to Semantics." This white paper includes a summary of the platforms and standards in today's semantic technology stack, as well as a summary of five areas where you can expect to see semantic technologies applied in the years to come. I hope you'll take the time to download it, read it and share it with colleagues who are interested in the role semantic tech may play in their companies.

If you are considering the adoption of semantic technologies, or if you're already putting them to use, I recommend that you join me at SemTech 2010, the semantic technology conference, held in San Francisco June 21-25. As in years past, SemTech will give you the opportunity to network with the leading technologists, entrepreneurs and researchers in the field. I serve as the co-chair of the conference, so guess I'm a little biased, but I think we've put together our best event ever. You still have time to register.

It's going to be a great event. Hope to meet you there.

Dave McComb
President, Semantic Arts

Contents

Introduction

Semantics is recognized as a hot topic in information systems and is entering into a phase of strong growth.

Even given its growing financial importance, the impact of semantics on information systems will be far greater than dollar figures suggest, as it's the application of semantics that will finally solve some problems that have dogged information systems the last several decades.

Consider this:

- Already hundreds of major corporations such as Boeing, NASA, and TVA employ semantic-based technologies to directly improve the effectiveness of their information systems.

- Between 35 and 65% of the $300 billion dollars being spent per year on systems integration is attributable to resolving semantic mismatches between systems.

- Almost all of our newest promising technologies such as Web Services, XML, Business Rules and Business Intelligence will depend on semantics for the success of their implementation.

What are Semantics and Semantic Technology?

Semantics is the study of meaning. It's as old as the ancient Greeks. For most of us it was a deadly dull sub-discipline of philosophy, to be avoided. But it turns out that we can't avoid it. We are drowning in a sea of data which occasionally is generously referred to as "information." But the truth is that almost all of it must be interpreted by humans to be of any use. The growth and availability of data and, therefore, our need to consider it in decision-making and planning is growing exponentially, and our systems, rather than helping with this, are for the most part contributing to the problem.

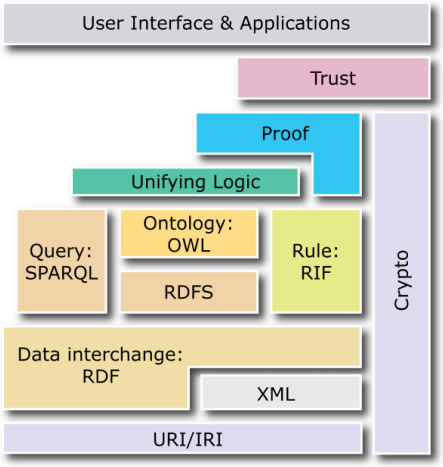
Semantic technologies include software standards and methodologies that are aimed at providing more explicit meaning for the information that's at our disposal. This takes different forms depending on where in the information cycle the semantic technology is applied and which area of the problem it is addressing; as we'll get into later in this paper, there are some commonalities between the technologies but also many differences.

The Semantic Web

In the early nineties Sir Tim Berners-Lee unleashed the World Wide Web in the form we are familiar with, and not long after that he realized that there was a limit to the effectiveness of the Web. While billions of documents could be linked and indexed, they relied on human interpretation to do anything with them. In the mid-nineties he began an initiative to promote research and standards under the banner of "the Semantic Web." After an incredible amount of research and development, the Semantic Web community has coalesced around three key standards: RDF, RDFS and OWL. These standards are the Darwinian survivors of intense competition among dozens of competing standards and approaches, and at that level, represent some of the best current thinking on this topic. Perhaps more importantly, the adoption by the W3C sends a clear message to vendors and consumers that in the future, all products and technologies in this space will, in order to be compatible, have to embrace these standards.

(See semantic web "Layercake" diagram next page)

Note that RDF and RDFS have become W3C standards, as has OWL at the Ontology vocabulary and Logic levels.



Semantic Web "Layercake" Diagram

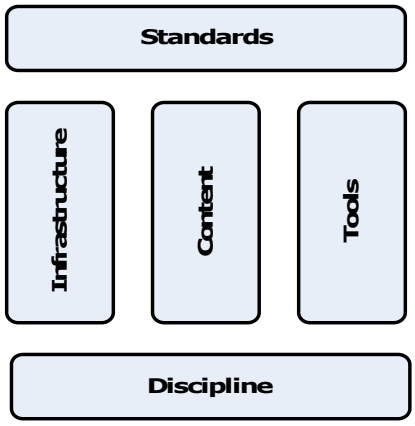
The Semantic Enterprise

Enterprises have been engaging in various semantically-based activities for decades. Indeed, conceptual data modeling is a form of informal Semantic Modeling. We believe that in the short term it is corporations and other large organizations that have the most to gain from the application of semantic technologies. We also believe that just as the corporations adopted Internet-based technology on an internal-only basis for many years in the form of "intranets," something very similar is about to happen and is already happening at some leading enterprises: the application of Semantic Web-based technologies to their internal systems.

As we suggested, products based on Semantic Web standards are not the only approach to solving an enterprise's semantic problems. Indeed, there are many proprietary approaches that certainly work as well or better, and companies should be encouraged to adopt any of these technologies where a clear benefit exists. At the same time, companies, especially those interested in pursuing open systems and loosely coupled architectures, should remain aware of the development of products and technologies embracing the Semantic Web standard technologies.

A Semantic Framework

The Semantic domain covers a lot of ground. This illustration and the text that follows should help provide a framework for the field. The framework consists of the following:



Discipline

The underlying discipline of semantics is a study of meaning and how we apply it in our systems. It is a pervasive undercurrent that expresses itself throughout the rest of the stack. Most of us have had very little formal study in semantics primarily because we are reasonably good at applying semantics informally. However, the quality of our designs and products can be improved drastically merely from paying attention to and studying the way we encapsulate and express meaning in our systems. Some of the key areas within this discipline include business vocabulary, taxonomy, ontologies, the theory of categories, prototype theory, dynamic classification and Description Logics.

In addition to a body of knowledge there are systematic ways to apply that knowledge: methods and methodologies. We are still early in the mainstream adoption of Semantic-based technologies, and as such there are not widespread and well vetted methodologies in place. There are many in development, and we should expect a few schools to become dominant in the not too distant future.

Content

Content is at the center of this framework. The content in a semantically enabled system is either an ontology, which is a body of knowledge formally expressed, or it is data coded to correspond with the ontology. Toward the end of this paper we'll describe how an ontology adds value to a system. For now, think of it first as a definition of terms. Think of it secondly as a formal, machine process-able way of describing relationships and constraints between terms in a way that a system could find and use similarities and differences in its processing.

With normal, unstructured content, often all we have is text. With text, we can perform keyword lookups but not much more. Presentation markup such as HTML isn't much help in understanding the content. XML taunts us with clues, supposedly meaningful tags such as <claim> or <destination>, but there is still little a system can do with these tags. In a semantically enabled system, the tags refer to well defined concepts, and the system can parse the formal definition of the concept and use that to combine the information with other potentially related data.

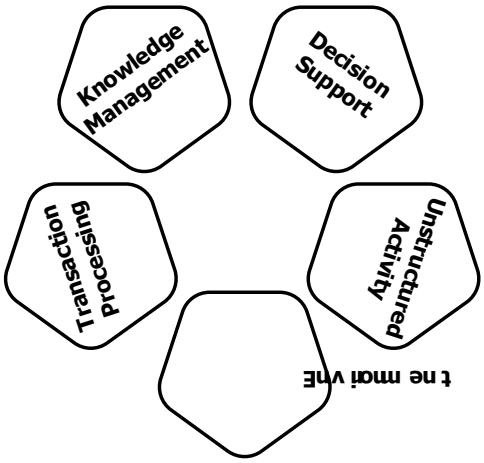
Tools

By tools we mean those that would be used by developers to aid in the development of a semantically aware system. These typically are not part of the runtime environment when the developed system is in use. These tools range from ontology editors, such as Protégé, which allow a knowledge worker or developer to build a valid ontology in a standard expression, such as RDF and OWL, and also include tools that are used to build maps between messages in a semantic broker environment.

Five Major Areas where Semantics will be Applied

In this section, we will divide the issues that an enterprise deals with into five major areas. The way Semantic Technologies will be applied in each of these areas varies greatly.

The five areas are:



Transaction Processing

Transactional systems such as payroll inventory management, ERP, and the like, are the bedrock of corporate information systems. Increasingly, semantic technology will be used to resolve the perennial problems that have been difficult to deal with over the course of the last several decades. The main issues with transactional corporate systems have been integration and interoperability, largely because of the great number of applications that the typical enterprise has implemented. Most enterprises of any size have dozens or, more typically, hundreds and occasionally thousands of separately developed and implemented application systems in-house. Many of the systems contain information or transactions that must be shared, however, each system was developed with its own concepts and local semantics. That is, each application has thousands of entities and attributes each with names and definitions and each with subtly different scope and validation and constraints, such that when information is sent from one application to another there are very often unexpected surprises in the receipt.

Corporations typically spend 35 to 65% of their budgets on integration and interoperation of these applications. A large percentage of those expenditures have semantic mismatches as their basis. The main technologies since the last decade to address these concerns have included EAI (Enterprise Application Integration), Service Oriented Architecture and Web Services. Each of these has been implemented in a somewhat ad hoc fashion in the past. Current products and initiatives will allow these approaches to be implemented far more predictably and reliably by relying on semantics.

The EAI approach was meant to replace hand-coded point-to-point interfaces with a hub, where communication is sent from one application to a central point to be distributed to the applications that need it. Certainly this has enjoyed some success; it is a multi-billion-dollar industry and many companies have enjoyed significant improvement by adopting this approach. However, sending a message to a central hub does not resolve differences in meaning between the sender and receiver. Luckily, it does provide a single point where these differences *can be* discovered and reconciled. Enter semantic brokers. A semantic broker is an adjunct or sometimes a replacement for an EAI hub, where the messages are first mapped to a shared ontology. This shared ontology provides a single definitive meaning for each component of the transaction which can then be retranslated back to any destination which has similarly mapped to the shared meaning.

One other thing about the semantics of transactional systems is that transactional systems manage what's been called “structured” information. And the nice thing about structured information is that once we have divined what the fields “mean,” we can rely, at least to a certain extent, on editing, validation, and other mechanisms that have been put in place in the transactional systems to assure that we are getting consistent meaning in those systems. It is spotty, but it’s better than nothing. We'll see as we go to some of the next areas of application that this structured rigor does not exist or has been changed.

Knowledge Management

Knowledge Management typically deals with relatively limited volumes of information that must be interpreted by experts. Very often this involves documents, regulations, laws, policies, and the like; and involves interviewing experts. This process is a very human and knowledge engineering-intensive task, where the intent is to organize

information that has been previously captured and collected and put it into a form where it can be scanned, queried, looked up, or applied at its point of need by someone who may not be nearly as expert in the domain. This area encompasses Business Rules, Content Management and Document Management. Artificial Intelligence was a big part of this domain for many years. The business rules approach is primarily finding the rules in such unstructured information, although occasionally business rules also aim to extract similar knowledge from legacy transaction systems. The central idea is that a knowledge engineer, especially coupled with a domain expert, can read great volumes of documentation and organize that documentation in a way that a system could look up what law, rule, regulation, or bit of corporate lore is important at the point of execution, when another member of the organization could use that knowledge if they had it available. The semantic angle here is twofold. The first is that until we have some semantic consistency and rigor in this process, the product of the knowledge engineer's work will be idiosyncratic. That is, each knowledge engineer will tag things with keywords or terms or whatever they feel are the important ways to index this information at the time they researched it. Furthermore, two terms which are closely related but have different keywords have no way easy way of being cross-associated. So one of the first things to do in an area like this is to develop a rich ontology, which not only will put some structure to the terms to be used, but also indicate which terms are related to other terms.

The second use of semantics in knowledge management or knowledge engineering is if the knowledge engineer uses the same or a compatible ontology to that which is being used in the transactional systems, there is a hope for marrying these two systems. There is a great deal of benefit to be gained in doing this. Imagine transactional systems that had tags for hazardous material in their inventory system. That material would be cross-linked to the knowledge management base of what needs to be done: constraints for transporting, storage, reporting, and the like. This is just one tiny example but I believe you can see the point. Furthermore, we believe that the business rules approach will be greatly accelerated by the adoption of standards in the area of ontology and, in particular, rule expression. Currently, most business rule approaches are proprietary, which has led to some reluctance for companies to invest in these technologies. But as the technologies adopt the open standard interfaces, reluctance to embrace these will naturally diminish.

Decision Support

There's a whole sub area of information systems whose primary purpose is to bring relevant information and analytics together and present them to decision makers to improve the quality of their decisions. This includes data warehousing initiatives as well as business intelligence, statistical information management, dashboards and executive information systems. These systems differ from the other two discussed so far in that, first of all, there often are very large quantities of data to be dealt with at any one point in time. Data warehouses commonly contain multiple terabytes of information and many contain tens and hundreds of terabytes. The other thing important about these systems is that a great deal of effort is spent in the harvesting of data from the transactional systems and its population, summarization, aggregation, and interpolation in the decision package. The primary part of this work, which often goes under the heading of Extract, Transform, and Load (ETL), consumes a great deal of effort in resolving semantic differences as well as the syntactic differences in the data coming from different source systems. Another major problem that has to be resolved with these systems is what's called data cleanup. This is necessary because different source systems place different importance on the reliability and veracity of information they store and, hence, combining data from multiple sources mixes and matches data of varying levels of quality and completeness.

Semantically, we want to do two things in this space. One, by semantically tagging not only the meaning of the information in the source systems but the clues as to the quality and completeness of the information, we can greatly aid the ETL process. The second area where semantics is beginning to help in these systems is at the point of use. Many of the systems by their nature are quite complex. They deal with thousands upon thousands of attributes in warehouses and data stores. Having an ontology-driven workbench or business intelligence dashboard would allow the less trained operators not only to understand the meaning and potential inclusion criteria of specific data elements that they’re reviewing, but also allow them to find similar or related terms that they might want to query on.

Unstructured Activity

In this category, we place a great deal of business activity that has escaped structured representation in the past. This includes a lot of individual interaction of employees with each other and customers and suppliers; it includes correspondence, phone conversations, e-mail, and a great number of tasks that are done on a daily basis. Some tasks are mediated by and result in transactions in the transactional systems and, as such, are currently pretty well captured and structured. However, there are a great many activities that are not being managed at the same level of structure and yet information is being generated and decisions are being based on this information. As you might expect, most unstructured activities create unstructured data.

The semantic approach to this unstructured data differs drastically from the semantic approach to the unstructured data in the knowledge engineering section that we talked about earlier. The knowledge engineering section relies on a top-down method employed by a knowledgeable person examining each document in detail. In the unstructured activity world, we have far too many small bits of miscellaneous activity for people to manually scrub and index and tag them. Instead, we must look to our systems to help in partially automating this process. Here the approach and the products are quite different. In knowledge engineering there's a tendency to proceed in a top-down fashion: What type of information is this? What is this rule or regulation about? That approach relies on first putting things into high-level categories and eventually getting down to specifics. This is very difficult to automate. In the unstructured activity world we rely on finding specific instances that are easy to syntactically parse from the information. So, if we were scanning e-mail, we might look for things that looked like order numbers or product numbers or customer IDs or customer names. These pieces of information are far less ambiguous than general words in English or any other language. Once a parser finds a piece of data like this, there’s an avenue to begin to look for other related information. For instance, if we detect an order number in an e-mail, at a minimum, we can index that e-mail to that order number, such that if anyone was doing subsequent research in a transactional system and wanted to see if there was any correspondence about it, they could find it. But there are other approaches in this area that go considerably beyond that. Armed with the order number, we could interrogate our systems and find out the customer, ship date, any problems with the product, etc., and might be able to build up a small bit of information and even inference from

other information in the e-mail that would now make sense from that context. While the approach to this area is different, proceeding from very specific instances in a bottom-up kind of fashion also benefits, for many of the same reasons, from having a shared ontology with the rest of the enterprise. This allows us to begin to interconnect our transactional systems, our knowledge management, our business intelligence, and our unstructured activity.

The Environment

Perhaps the really big payoff is in connecting our enterprise to the larger world. This includes our supply and demand chains, through which we are currently connected with e-commerce and B2B. It also includes the whole universe of our prospects who might not yet be in any of our systems including our CRM systems. Further, it includes the research that can be done outside our systems. This might be surveillance on competitors or it might be pure research. In many ways, this might be the largest area and the biggest opportunity. It is already a huge undertaking. Many companies are spending a great deal of time with their supply and demand chains especially the e-commerce and B2B aspects of them. On the Internet, search, which is in this category, is one of the largest and hottest product areas. We believe, with the adoption of the Semantic Web standards RDF and OWL, that the importance of this area is going to multiply greatly and the savvy enterprise will figure out how to tie its internal RDF- and OWL-expressed ontologies and systems with similar or at least map-able ontologies that can be committed to on the Web. We believe that the nature of search will be greatly changed by the addition of semantics to search technology. One of our favorite current examples is that trying to find something written *by* a U.S. President is very difficult because there's so much written *about* the U.S. President; no current keyword or relevance-related search will make that distinction. This is a semantic distinction; well captured by a simple and widely used ontology called the Dublin Core. This is an ontology about authorship of documents, and anyone committing to the Dublin Core agrees that their tags that have something to do with authorship or creation mean the same thing as the Dublin Core definition of document creation, which allows many heterogeneous sites to be searched and to obtain a common result.

How Does it Work?

Systems get their power (and their complexity) from the distinctions they make and use. In a very simple system we might sell "things" to "people." If that was all there was to it we wouldn't need massively complex ERP systems and the like. What happens is we have "requirements." The "requirements" generally entail our treating some things differently than others. So, we might decide that we have to treat "perishable items" different from the rest of the things we sell (keep track of how long they've been on hand, be careful to sell the oldest items first, etc.). In a traditional system, this means introducing additional schema and code and procedures such that we can treat some of the "things" differently. Each distinction introduced into a system increases the complexity of a system far more than it first seems. Every time we implement a distinction in a traditional system, not only does it increase complexity directly, but it has a much greater side effect. Anything else that previously dealt with the non-distinguished concept now has to deal with two or more implementations of similar but different concepts. Also, very often the distinction is buried where we only find out about it accidentally (in testing or even in production).

If we put a line of code in an application that says

"IF PURCHASE_ORDER > $1000…"

we have made a distinction (between expensive and non expensive Purchase Orders) and we treat one type differently than the other. In the semantic approach we bring these distinctions out in the open where people (and systems) can reason about them. In so doing we can stop the runaway growth in complexity in the source systems. For instance, we may have several systems that contain information on contracts, agreements, orders and transfers. We might impose a requirement to report "material transactions." The semantic approach would be to map all the existing systems' meta data to a shared enterprise ontology, so that we would know for instance that "PO_Date" was on a record that indicated a commitment (purchase orders are commitments) and that the date represents the effective date of the commitment. Having mapped the existing systems to a shared enterprise ontology is the first step. The second is to create a formal definition of a "material transaction." The accountants would obviously help with this, but imagine that you determined that a contract worth more than $1,000,000 or a purchase order for services

over $200,000 represented a material transaction. This definition could then be used by an inference engine to find material transactions in any system that committed to the ontology. Note that committing to the ontology merely means agreeing on the base terms (in this case contracts, agreements, etc.) and does not require implementation of or agreement on the definition or even the existence of the concept of material transaction.

How Should We Get Started?

The first few things to realize are:

- Getting started with Semantics does not require a large capital investment. Many tools are free. Infrastructure components are reasonably priced, certainly for proof of concept level experiments.

- It may take longer than you think. Many of the concepts require deep rethinking of how systems are put together, and as such it takes teams a while to adapt to the new ways of thinking.

- You need not adopt all the aspects of the framework, nor work in all the affected business areas to gain a benefit.

Armed with those insights, we'd suggest:

Do something. You cannot hurt yourself applying semantics to a current project. At the low end of the result spectrum you might gain some new insight into the problem, even if you still implement using tools and methods you are familiar with.

Start now. The long lead time and low capital investment mean that waiting may well put you at a competitive disadvantage.

Get educated. The field is far more vast than first meets the eye. Plan on a long, rewarding educational effort that will include books, internet research, training and conferences.

Consider standards. You can learn from the knowledge of others who have come before and standardized some of what they have learned. In many cases the standards are directly and freely implementable.

Communicate. You won't be pursing this alone. There are many active user groups on the internet, and there are consultants, vendors and peers eager to help with your efforts.

We believe that semantics is one of the major sea changes in our industry. After years of research and academic work it is being applied by early adopters in a number of important areas. The best time to start is now.

Summary

In this article, we introduced the topic of applying semantics to Information Systems. We surveyed the technology stack from underlying disciplines to standards, and we described at a high level how it is likely to change five broad categories of information systems.