Cerebra®

SOA and Semantic Architectures

In Practice and Why to Bother

A Cerebra Technology Whitepaper

Version 1.0

November 2005

Executive Summary

Without an innovative data strategy, a Service-Oriented Architecture (SOA) is just more plumbing. The core business mission of the modern generation of SOA is two-fold: business agility and cost reduction. These objectives were dependent upon two technical capabilities: loose-coupling and dynamic discovery. But without "smart data," the SOA is just as brittle as EAI or CORBA before it.

Whether your interest is in enterprise software architecture, advanced digital IP networks of the future, or simply helping business users find stuff easier, you will face the inevitable truth that today's world is a system of systems with no center and little control over the edges. Ontology, OWL, and RDF provide the means for coping with this complexity. As it turns out, metadata and tagging is a strategic requirement for complex software environments.

The power of OWL and RDF lies in machine interpretable semantics. Developing semantic applications and semantic services for the SOA will inject a layer of flexibility between architectural components, ultimately proving a little extra "give" in the normally brittle connections between systems and sub-systems.

© 2005 Cerebra Inc. all rights reserved.



Table of Contents

ENTERPRISE ARCHITECTURE AND SEMANTICS 3

JOB ONE: THE BUSINESS VALUE CHAIN 4

ORGANIZE, INTERPRET, AUTOMATE – SEMANTIC METADATA 5

CURRENT ARCHITECTURAL MODELING PRACTICES 7

SEMANTICS AND THE MODERN SOA 9

MEDIATION ENGINE (AKA: ONTOLOGY OVER WSDL) 9
SEMANTIC REGISTRY (AKA: ONTOLOGY OVER UDDI) 9
PROCESS BROKER (AKA: ONTOLOGY OVER BPEL) 10
SYSTEM OF SYSTEMS (AKA: THE SEMANTIC SOA) 10

IF RISK TOLERANCE IS LOW, WHY BOTHER? 11

ABOUT THE AUTHOR 12

© 2005 Cerebra Inc. all rights reserved.

Without an innovative data strategy, a Service-Oriented Architecture (SOA) is just more plumbing. The core business mission of the modern generation of SOA is two-fold: business agility and cost reduction. These objectives were dependent upon two technical capabilities: loose-coupling and dynamic discovery. But without "smart data," the SOA is just as brittle as EAI or CORBA before it. Don't take my word for it.

> "Business agility depends on the ability to assemble, disassemble and rearrange application components. These actions require a comprehensive understanding of not only data representation ('syntax'), but also data's meaning and its relationships to other data and information — that is, the 'semantics.'"
> *-Intelligent Enterprise, "Start Making Sense […] Semantic Integration," October 2005*

> "You will waste your investment in SOA unless you have enterprise information that SOA can exploit. Machine-processable information cannot exist only for use with the application that created it. Any application and service (or trading partner) should be able to pull in content from anywhere in real time and be assured of any semantic differences; that information will have to be able to change, not remain frozen in its original form."
> *- Gartner, "Service-Oriented Business Applications Require EIM Strategy," March 2005*

> Defining the business meaning of transactions and data is the most intractable issue that IT managers face. Although the semantics challenge predates Web services -- vertical industries have been developing their own XML schema for years, for example -- SOAs bring semantics to the fore.
> *- Infoworld, "The Five Missing Pieces of SOA," October 2004*

The solution? A "semantic web" of "smart data" with SOA as the machine-to-machine middleware. In February of 2004 the World Wide Consortium (W3C) recommended two new global standards: OWL and RDF. Already responsible for standards like HTTP, HTML, XML, and Web Services, the W3C is clearly the most relevant standards organization for users of the Internet. But with the launch of OWL (Web Ontology Language) and RDF (Resource Description Framework) Sir Tim Berners Lee, the current Chair of W3C, intended to do no less than to change everything.

The big vision of the Semantic Web with a capital "S," as the famous 2001 Scientific American article puts it, "an Internet that knows what you mean," is still a hotly debated prognosis for the future. However, it is increasingly clear that early adopters are finding great use from the new technology in an unexpected place – behind the firewall. It seems that the ideal incubator for the Semantic Web is right inside the very same air-cooled data centers that house our old mainframe and ERP systems.

Enterprise Architecture and Semantics

So why would businesses be the optimal place for incubating new technology? After all, in these tightly competitive times, cash-starved IT groups rarely have time to work on any cool forward-thinking projects. Whereas the HTML Web initially took hold in University, and only made it to business after the geeks finally convinced their pointy-haired bosses that connecting people was good for goodness sake, the pundits for semantic metadata are finding eager ears and open pocket books because of one simple reason – it does stuff. Using technical words, the Semantic Web is executable.

Unlike tags in HTML and XML, the tags in RDF and OWL compose a network of data that lives outside individual documents or physical locations and can be re-categorized on-the-fly to suit the

© 2005 Cerebra Inc. all rights reserved.

context it is being used in at a given moment. However, there is no magic behind this power. As any good software architect knows, the principles behind good software design don't change. The fundamental design tools of abstraction and indirection empower and provide patterns of use for these W3C specifications.

Traditional application architectures might be, well, traditional, but that doesn't mean there's nothing to improve. In fact, in systems ranging from relational databases to object-oriented Java there is growing pressure to make them more flexible and in-touch with organizational change.

The great paradox in IT today is the pressure to decentralize and disperse information to the edges of the company, while simultaneously centralizing and managing content as a business asset. From a technical view, the actual business policies, rules, and domain information crucial to operations are spread thin across complex and disparate software systems. As policy control systems, for managing and executing business rules on all sorts of systems, and metadata management platforms, for linking and joining content in highly federated ways, both OWL and RDF based tools excel.

In their most basic sense, OWL and RDF do a great job at organizing things. In particular, the contents of information systems – which could be documents, database tables and columns, XML tags, SOA service interfaces, APIs, files and so on. What would you do once you've organized things? You'd let people find those things. Simply speaking, OWL and RDF help people find stuff.

But what about Google? Google is a great search engine. As it turns out, it may not be all that great for enterprise search. Google's Page Rank algorithms seem to work better with Internet pages than they do with the much smaller number of documents and users behind the enterprise firewall. It's not that it's bad, it's just not that much better than the last search engine you had. Alternately, combinations of ontology category schemes, with automatic document classification systems, can help you build a dynamic version of any particular taxonomy for enterprise content. Or, as one blogger puts it, "the best way to find stuff is not to lose it in the first place."

For the enterprise architecture, we can summarize the Semantic Web technology base by saying that it helps architects and end-users to organize, interpret, and automate business information among federated systems. OWL and RDF provide a great way to organize enterprise content, inference algorithms provide technical advancements for contextual interpretation of that content, and since the semantics become deterministic and machine-readable, they are key for successfully automating business events.

Job One: The Business Value Chain

So, what does any of this mean for your company? Why should any non-techie within business organizations pay attention at all?

In an era of thin margins and extraordinary demands for top-notch customer service, a more flexible software infrastructure can help reduce costs and further empower front line personnel to make smarter decisions. The term "adaptive enterprise" may sound like fluff to a technical guru, but it accurately conceptualizes what semantic technology can bring to any business organization.

For most industries, there are many crucial IT systems that must work in perfect concert each day in order for the business to operate. But in order for the business to grow and prosper, these very same IT systems must change to provide more services, more capabilities, at lower cost.

© 2005 Cerebra Inc. all rights reserved.



A few value chain questions to consider:

- Sales, Marketing, and Product Management
  - How easy is it to roll out new marketing and customer loyalty promotions?
  - How long does it take to change business policies for sales and product management?
  - Do you have a dynamic view of your customers? Their habits and patterns?
  - Do you have a dynamic view of your products? Their configurations and sales packages?

- Financial Reporting
  - How many people edit reports in spreadsheets after first generating them?
  - Do errors creep in that shouldn't?
  - Where are regulatory policies managed? How are policies enforced? Or audited?
  - Does data come from more than one system? Can you validate accuracy?

- Supply Chain Management and Logistics Planning
  - How predictable are routes and supply stock? Is ordering electronic?
  - Do many systems communicate in a workflow? Do they all have the same business vocabulary?
  - Are there standards in use for supplier communications? Are they adhered to?
  - How do your IT systems respond to unexpected events?

- Employee Tools
  - Do associates have the information they need at all times? Does this impact customer service?
  - Are there IT systems that make the workplace simpler and more enjoyable?

- Infrastructure
  - How much budget is tied up in maintenance?
  - Are ongoing projects strategic, or are they fixes for localized problems?
  - Do organizational changes usually require new programming or coding efforts?
  - How easy is it to find things? Documents? Services? Other Data?

These questions are open-ended and meant to get you thinking, but if you found yourself wishing for some better answers, there might be a role for semantic technologies to help. Somewhat ironically, SOA vendors and pundits use these same sorts of questions to "sell" the SOA vision. But since these value chain questions require data-intensive solutions, it should be clear that SOA alone will not win the day.

Cool technology that creates no efficiency, no new capability, or doesn't at least make a better employee, can be little more than a toy. Thus, deploying IT that is aligned with business value chain goals is so important. Thankfully for Sir Tim and the rest of the gang at the W3C, there are more than a few Global 2000 companies already exploiting semantics to develop those efficiencies, drive value, and ultimately – profit.

Organize, Interpret, Automate – Semantic Metadata

OWL and RDF emerged as standards to provide a capability that cannot easily be handled with existing XML and Java centric solutions. In short, the ability of a hard-wired infrastructure to dynamically change and respond to normal business events is woefully lacking. We as architects are good at creating software blueprints that scale well and encapsulate business functionality in discrete code layers. But the tools and languages available to us have limited our ability to build architectures, software or infrastructure, which can respond to unforeseen changes in business

© 2005 Cerebra Inc. all rights reserved.

operations. The constraining factor here has not been in our imagination, but rather in the options we had for effecting good solutions.

The Resource Description Framework (RDF) is a graph-based data system. It has been described as a more organic approach for implementing systems with rapidly changing data structures. It has also been described as a way of creating a network within the data. Still others talk about a schema-less database.

As opposed to a relational data system, which is built around the premise of data matrices, RDF's core unit of data is the "triple." An RDF triple consists of a Subject, and Object, and a Predicate. Here, a Predicate is roughly akin to a relationship. Thus, a unit of data might be Document authoredBy PersonName. The subject is Document, the object is PersonName, and the predicate is authoredBy. From this simple approach, the graph of attributes for a Document can be built up in fairly arbitrary manner, and complex chains of relations can be built, such as: from Document to Author, Author to Person, Person to Organization, Organization to Industry and so forth. The network of associations between units of data can grow organically and without concern for rigid ER or XML schemas that have to be dumped and rebuilt when the original assumptions about the model change. Figure 1 shows how RDF differs in design and intent from other kinds of data systems.

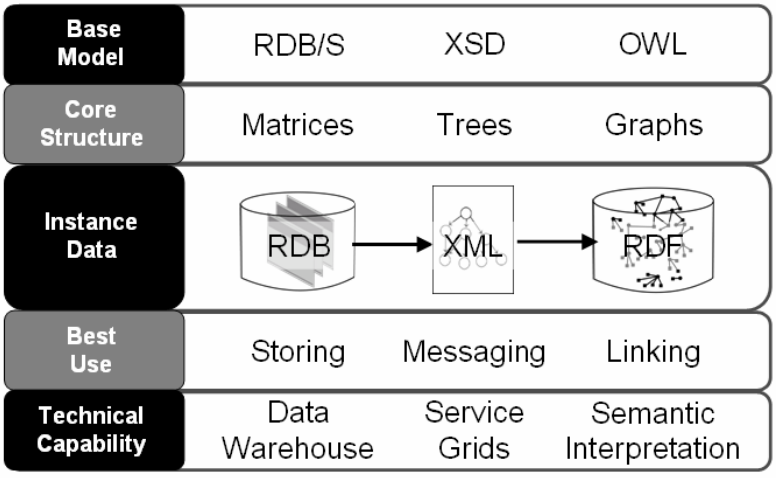


Figure 1: Basic Comparison of Data Formats

The Web Ontology Language is a semantic network approach towards constraining RDF graph data. It uses a type of logic called Description Logic to build up associations in the schema, data, and relationships that can be used to infer new information from explicitly stated information. In Figure 2 we see a simple example of how the data model can state that a leader of a tour is a member of a tour, and that a member of a tour is a golfer, the conclusion of such a relationship may be that the leader of a tour is a golfer. In this way, we can write queries that ask for all known golfers, and the sub-system can infer who is a golfer or not without having to explicitly identify every player or leader as a golfer. This capability makes it simpler to write conceptual queries in business terms, leaving aside the rigid categorizations imposed by relational or XML subsystems and instead using the conceptual framework provided by OWL as a metadata layer above the source systems and SOA infrastructure.

© 2005 Cerebra Inc. all rights reserved.

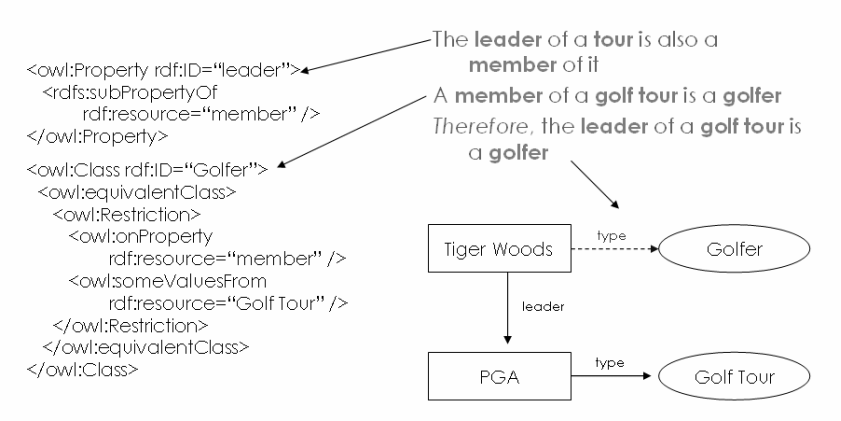


Figure 2: Simple Example of Inference

Similarly, the OWL models can be used to contain and express rules about how the data should appear in context. By dynamically reclassifying data at runtime, according to the situation at hand, we can present data in the terms that are appropriate for the view. Additionally, we can implement business policies, such as "all customers who spend more than $10k per month and are in good standing are Gold Club customers." These policies can then be used to drive business applications that rely on the proper categorization of data, and auditable policies, for presentation to end-users. Achieving these capabilities in a data model is no small feat – it means that we as solution architects can encode operational business rules in a metadata layer, which can be reclassified on demand to reconstitute the business view, or present an audit trail for what business policies are being enforced at a given time.

In sum, the combination of OWL and RDF can create a metadata logical façade to enterprise software systems. This façade is the key to developing an infrastructure always provides for an organized view into federated software systems, which can in turn allow your business users to accurately interpret confusing volumes of data, and also to automate processes based on the data classifications that are derived at runtime.

Current Architectural Modeling Practices

In practice there a number of new things to consider when working with semantic metadata, and a number of older disciplines that must be adhered to diligently. There is no magic with semantic technology. The old-fashioned practice of discipline in modeling a domain, the diligence to make it accurate, and the flexibility to change assumptions along the way are still of primary importance.

Most early adopters of ontology and graph systems (OWL and RDF respectively) rely on old object-oriented (OO) techniques of verb-noun extraction and taxonomy methods to define a domain. But, unlike OO methodologies, designing an ontology focuses on the structural components of a class rather than the operational properties of a class. For example, there are new considerations to consider such as the logic of relationship types, OWL provides for expressive relationships like:

- Equivalent classes (Class A equals Class B)
- Unions of classes (Class A or Class B)
- Intersections of classes (Class A and Class B)
- Disjoint classes (Class A implies not Class B)
- Inverse (Relationship X is inverse of Relationship Z)
- Transitivity (A ancestor B and B ancestor C, then A ancestor C)
- Etc.

© 2005 Cerebra Inc. all rights reserved.

These new kinds of relationships, called axioms and class constructors in OWL parlance, need to be evaluated in context of the domain model which is being built. Designers and analysts who are comfortable with modeling with the Unified Modeling Language (UML) and Entity-Relationship (ER) approaches can usually be trained to model with OWL and RDF in short order. Increasingly, business taxonomy designers are being leveraged to bridge the gap between business vocabularies and programmer designs.

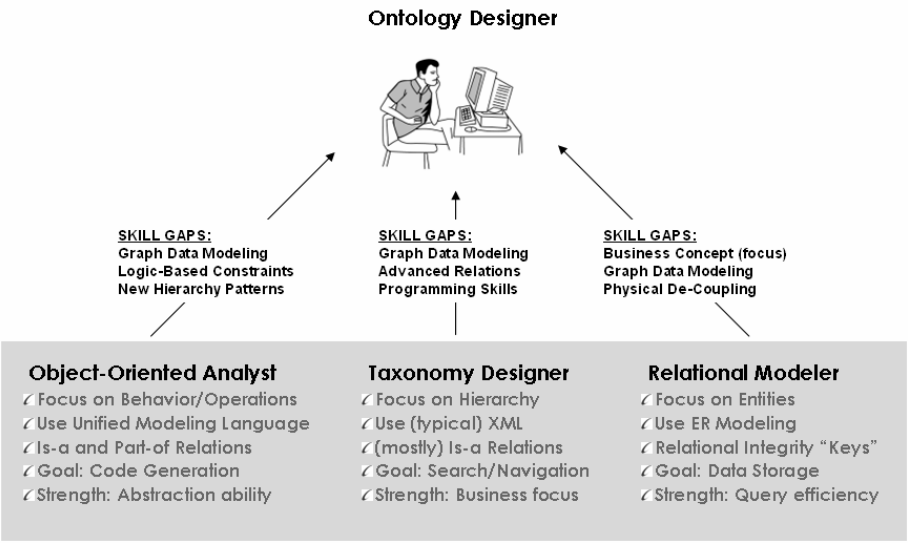


Figure 3: Skill Gaps for Ontology Designer Role

Odds are that skilled ontologists already exist within your organization. They are the people tasked with creating database models, taxonomies, or object oriented design, but remain dissatisfied when elegance of their original designs must change for software implementation or because of departmental politics. The tools for maintaining conceptual modeling purity and accommodating differing views of the truth simply haven't existed before OWL and RDF technology. Ontology designers will have the business focus of a taxonomist, the conceptual abstraction abilities of an OO analyst, and the eye for modeling efficiency that relational models possess. Fortunately, these skills gaps are easily addressed with a few weeks training.

For the most part the above discussion implies a manual effort. In fact, there are at least three methodologies under development for just such hands-on modeling in these new languages, several of which have explicit ties to UML methodologies. Likewise, there are several automated approaches to model development which can speed the ontology creation process considerably. For instance, there are plenty of tools available that can generate OWL models from pre-existing UML, ER, or XML Schema models. These can be used to generate a "seedling ontology" that could greatly speed the time to development. Although quality can be a substantial issue when generating ontologies for extremely de-normalized legacy models, there are regularly substantial benefits to using an automated process for ontology creation in actual practice. In fact, some newer tools are using Latent Semantic Indexing (LSI) to actually generate ontology from unstructured text documents. For example, newspaper articles have been used to create ontologies that capture relationships between current events, very useful artifacts for certain kinds of businesses.

Like any good software technology, there are patterns of use emerging that experts are capturing. The W3C has a best practices working group comprise of dozens of expert users that are documenting and communicating best practice design patterns for modeling. These patterns are

© 2005 Cerebra Inc. all rights reserved.

the basis for many of the public domain industry ontology models currently being built in Life Sciences and the Defense industries.

Semantics and the Modern SOA

Service-Oriented Architectures are all the rage these days. Like the Semantic Web, SOA applies the pattern of indirection. But whereas the SOA is a solution for loose-coupling at the messaging layer, the Semantic Web family of specifications is for loose-coupling at the data layer. Without a semantic layer, the SOA will remain forever brittle and become more expensive the more data that flows through it. Examples show how OWL can be applied as a federated modeling layer over UDDI, WSDL, and BPEL. Likewise, OWL and RDF can help drive SOA service utilities like Registries, Repositories, Process Tools, and Mediation Engines. The jury is still out on where semantics will most improve the SOA, but that it can and will – the jury is in.

Mediation Engine (AKA: Ontology over WSDL)

Since 1994 there has been a gradually building discussion about a "data mediation" capability in software. Although there is still disagreement on exactly what it means, there are now tangible examples of the capabilities it can provide. Roughly speaking, the Mediation Engine is a service that provides applications a common façade to query, transform, protect, notify and publish data from one or more disparate Web services. On one hand it uses ontology to automate the transformation of data between different XML vocabularies, on the other hand it combines the power of structured queries and inference algorithms to filter data into the correct context of the client applications. At a very basic level, ontologies can be related to WSDL 2.0 message attributes such that the vocabulary terms (APIs and XML elements) are then related to conceptual metadata. The Mediation Engine uses that conceptual metadata to automate data behavior in a mixed service environment composed of multiple organizations, communities of interest, and end-users.

There are no standards for how a Mediation Engine should be constructed. However, there are industry efforts at specifying the kinds of metadata links between WSDL and OWL that would be required. Two examples of open efforts are OWL-S and WSDL-S. Predictably, the OWL-S format starts with OWL and adds a WSDL service grounding and metadata model (along with process description). In contrast, the WSDL-S specification starts with WSDL and adds links to OWL concepts. Both can be effective for enhancing service capabilities, OWL-S goes further with modeling sophistication, but WSDL-S is less invasive into existing WSDL implementations.

Some vendors have implemented commercial approaches to the Mediation Engine as well. For example, the Cerebra Server uses a completely decoupled approach to mapping OWL ontology concepts to WSDL attributes. OWL concepts are annotated with descriptions about the physical location and bindings of WSDL attributes. At runtime the mediation capability processes queries, using inference to resolve semantic inconsistencies between data meaning, and then issues WSDL calls over SOAP to retrieve corresponding data from the source service. In this manner data from multiple services can be aggregated and filtered, using security ontologies for example, in response to client requests.

Semantic Registry (AKA: Ontology over UDDI)

Unlike the Mediation Engine, a Semantic Registry does not use metadata to link directly to the data exposed by Web services. Instead, the Semantic Registry enhances typical UDDI taxonomy to better describe and link the categories of the functions that each service exposes. In other words, the Semantic Registry is metadata about the service categories, not the service data.

Today there are two philosophies about this approach. One approach relies on the conceptual domain model for service data to provide enough information about the data to allow for discovery of new services based purely on the metadata about the available service data and client specified "goals." Both OWL-S and WSMO (Web Services Modeling Ontology) take this approach. They each call for a "Service Profile/Capability" to contain descriptions about the

© 2005 Cerebra Inc. all rights reserved.

service data, which would then be queried directly for discovery purposes. This “goal-driven” discovery approach can be quite powerful, but is also more difficult to implement since it relies on very rich tagging of service capabilities.

An intermediary step being explored by some (including Cerebra) is to simply extend the UDDI taxonomy with OWL and RDF extensions, thereby enabling rich category descriptions and policies for how to inference across distributed UDDI registries. In this approach the service category metadata is decoupled from UDDI descriptions with only the UDDI Service Key as a pointer to the low-level URI. Thus, the ontology can be used to assist end-user navigation of taxonomy, with inference to resolve semantic inconsistencies, but ultimately pointing back to some UDDI server on the Web.

Process Broker (AKA: Ontology over BPEL)

This is the least developed of the semantic capabilities for SOA. Currently the work at the Digital Enterprise Research Institute (DERI) on WSMO is probably the most advanced, although Nokia, SRI and Carnegie Mellon have submitted the OWL-S proposal to the W3C – which also uses a goal-driven approach for process composition. In general, both groups are focused on created a fully automated execution environment for Web Services – using ontology as the intelligence behind the scenes. Look for these to mature into products by late 2006 and 2007.

Taking the middle road, Oracle and Cerebra have developed a BPEL process service that uses ontology to classify and coordinate service processes. Essentially, processes as BPEL files can be described using OWL, which can then be used to discover the processes from within new client applications. Although it isn’t fully automated in practice, it makes smart use of semantic technology to partially automate the selection of available secondary services when primary services are unavailable. Full automation of a service execution environment would require the elimination of brittle BPEL conversation specifications, or at least the conversion of BPEL files into a more goal-driven framework like OWL-S or WSMO.

System of Systems (AKA: the Semantic SOA)

Whether your interest is in enterprise software architecture, advanced digital IP networks of the future, or simply helping business users find stuff easier, you will face the inevitable truth that today’s world is a system of systems with no center and little control over the edges. Ontology, OWL, and RDF provide the means for coping with this complexity. As it turns out, metadata and tagging is a strategic requirement for complex software environments.

However, the data architecture is only one side of the coin. In order for the data architecture to be executable and dynamic there must be platform and tool support to make it a reality. The SOA and Grid concepts provide the ideal launching point for these capabilities. Because the SOA encourages conversion to XML data exchanges, the syntactic and structural elements can be agreed to. Then the semantic issues take center stage, which is where OWL and RDF add new capabilities and value. In Figure 4 the system of systems are the “Application Services” that wrap existing and greenfield IT applications. Mediation, Process and Registry services, as shown in Figure 4, provide the utilities to achieve broad-scale network automation.

© 2005 Cerebra Inc. all rights reserved.

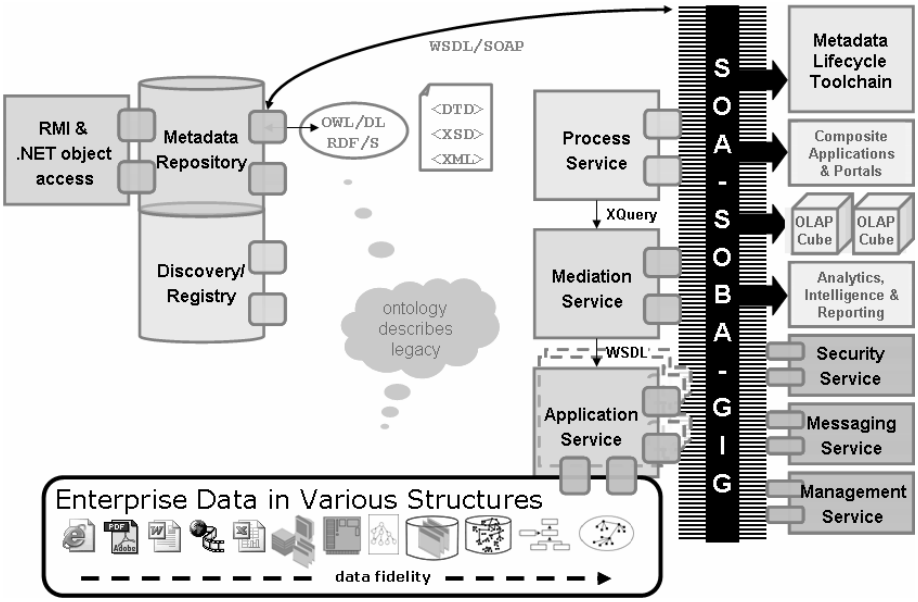


Figure 4: Example System of Systems Architecture with SOA/Grid Utilities

As far as mature software services go, both semantic Mediation and Registry services have been developed, deployed, and been shown to achieve tremendous value. Approaches to semantic Process orchestration and choreography are still in test-bed phases but have certainly been prototyped extensively. Without this sort of open framework for systems of systems interoperability, IT buyers will be relegated to expensive and closed vendor solutions into the indefinite future.

If Risk Tolerance is Low, Why Bother?

Although we have been discussing many of the end vision benefits for the technology, there is a very simple answer to the question of "why bother?" It is true that to fully leverage the power of ontologies and graph systems a considered architectural approach is required. As discussed in earlier examples, business software and infrastructure software stands to benefit in very large ways. But if an organization's appetite for change is small, ontologies and graph systems can still be very useful as a simple data dictionary. At the bare minimum, an OWL and RDF based data dictionary provide organizations a reference guide to the terminologies and vocabularies in use within the enterprise. Importantly, dictionaries in this format can evolve over time quite efficiently and incrementally be used to link together SOA services, databases, and other IT systems – eating the elephant one bite at a time as they say.

Sometimes these incremental modernization strategies are the best for cautious IT groups.

Nobody ever said that software architecture is easy. Like structural architects who design bridges and buildings, our software blueprints eventually come to life as highly-complicated and interleaved engineering marvels. Planning for loose-coupling, ease-of-change, and overall systems agility must be grounded in real, not imagined, software capabilities. Relying on Java coding patterns, SOAP interfaces and XML documents can only get us so far.

The power of OWL and RDF lies in machine interpretable semantics. Developing semantic applications and semantic services for the SOA will inject a layer of flexibility between architectural components, ultimately proving a little extra "give" in the normally brittle connections between systems and sub-systems. Business users need not know about the inner workings of semantic architectures, just as office-dwellers needn't know about how earthquake-resilient

© 2005 Cerebra Inc. all rights reserved.

buildings are constructed, so long as the functions they need can keep pace with the rate of change in their organizations.

There are many best practices emerging for eliciting and managing semantics. Perhaps the most frequently abused idea about semantics is that they must be applied across the enterprise to have value. In fact, the best place to start is in smaller, well constrained domains that can achieve high value in short order. Even completely decoupled systems, such as data dictionaries, can benefit from these newer and more flexible data languages that have emerged from the W3C. A little semantics goes a long way.

About the Author

Jeff Pollock is a technology leader and author of "Adaptive Information: Improving Business Through Semantic Interoperability, Grid Computing, and Enterprise Integration." (John Wiley & Sons, 2004) As Vice President of Technology at Cerebra, Mr. Pollock is responsible for product strategy, management and the envisioning of next-generation adaptive enterprise software. Previously, as Chief Technology Officer of Modulant, he delivered one of the industry's first market-ready semantic integration middleware platforms in 2001. Throughout his career, he has architected, designed, and built application server/middleware solutions for Fortune 500 and US Government clients. Prior to Modulant, Mr. Pollock was a Principal Engineer with Modem Media and Senior Architect with Ernst & Young's Center for Technology Enablement. He is also a frequent speaker at industry conferences, author for industry journals, active member of W3C and OASIS, and formerly an engineering instructor with University of California at Berkeley's Extension on the subjects of object-oriented systems, software development process and software systems architecture.

© 2005 Cerebra Inc. all rights reserved.