

Fresh Perspectives on the Semantic Enterprise

by Mike Bergman - Monday, September 28, 2009

<http://www.mkbergman.com/825/fresh-perspectives-on-the-semantic-enterprise/>



The Benefits are Greater — and Risks and Costs Lower — Than Many Realize

I have been meaning to write on the *semantic enterprise* for some time. I have been collecting notes on this topic since the publication by [PricewaterhouseCoopers](#) (PWC) of an [insightful 58-pp report](#) earlier this year [1]. The PWC folks put their finger squarely on the importance of ontologies and the delivery of semantic information via [linked data](#) in that publication.

The recent publication of a special issue of the *Cutter IT Journal* devoted to the semantic enterprise [2] has prompted me to finally put my notes in order. This *Cutter* volume has a couple of good articles including its editorial intro [3], but is overall spotty in quality and surprisingly unexciting. I think it gets some topics like the importance of semantics to data integration and business intelligence right, but in other areas is either flat wrong or misses the boat.

The biggest mistake are statements such as ". . . a revolutionary mindset will be needed in the way we've traditionally approached enterprise architecture" or that the ". . . semantic enterprise means rethinking everything."

This is just plain hooey. From the outset, let's make one thing clear: No one needs to replace anything in their existing architecture to begin with semantic technologies. Such overheated rhetoric is typical consultant hype and fundamentally mischaracterizes the role and use of semantics in the enterprise. (It also tends to scare CIOs and to close wallets.)

As an advocate for semantics in the enterprise, I can appreciate the attraction of framing the issue as one of revolution, paradigm shifts, and *The Next Big Thing*. Yes, there are manifest benefits and advantages for the semantic enterprise. And, sure, there will be changes and differences. But these changes can occur incrementally and at low risk while experience is gained.

The real key to the semantic enterprise is to build upon and leverage the assets that already exist. Semantic technologies enable us to do just that.

Think about semantic technologies as a new, adaptive layer in an emerging interoperable stack, and not as a wholesale replacement or substitution for all of the good stuff that has come before. Semantics are helping us to bridge and talk across multiple existing systems and schema. They are asking us to become multi-lingual while still allowing us to retain our native tongues. And, hey! we need not be instantly fluent in these new semantic languages in order to begin to gain immediate benefits.

As I noted in my popular article on the [Advantages and Myths of RDF](#) from earlier this year:

We can truly call RDF a disruptive data model or framework. But, it does so without disrupting what exists in the slightest. And that is a most remarkable achievement.

That is still a key takeaway message from this piece. But, let's look and list with a fresh perspective the advantages of moving toward the *semantic enterprise* [\[4\]](#).

Perspective #1: Incremental, Learn-as-you-Go is Best Strategy

For the interconnected reasons noted below, RDF and semantic technologies are inherently incremental, additive and adaptive. The RDF data model and the vocabularies built upon it allow us to progress in the sophistication of our expressions from [pidgin English](#) (simple *Dick sees Jane* triples or assertions) to elegant and expressive [King's English](#). Premised on the [open world assumption](#) (see below), we also have the freedom to only describe partial domains or problem areas.

From a risk standpoint, this is extremely important. To get started with semantic technologies we neither need to: 1) comprehensively describe or tackle the entire enterprise information space; nor 2) do so initially with precision and full expressiveness. We can be partial and somewhat crude or simplistic in our beginning efforts.

Also extremely important is that we can add expressivity and scope as we go. There is no penalty for starting small or simple and then growing in scope or sophistication. Just like progressing from a kindergarten reader to reading Tolstoy or Dickens, we can write and read schema of whatever complexity our current knowledge and understanding allow.

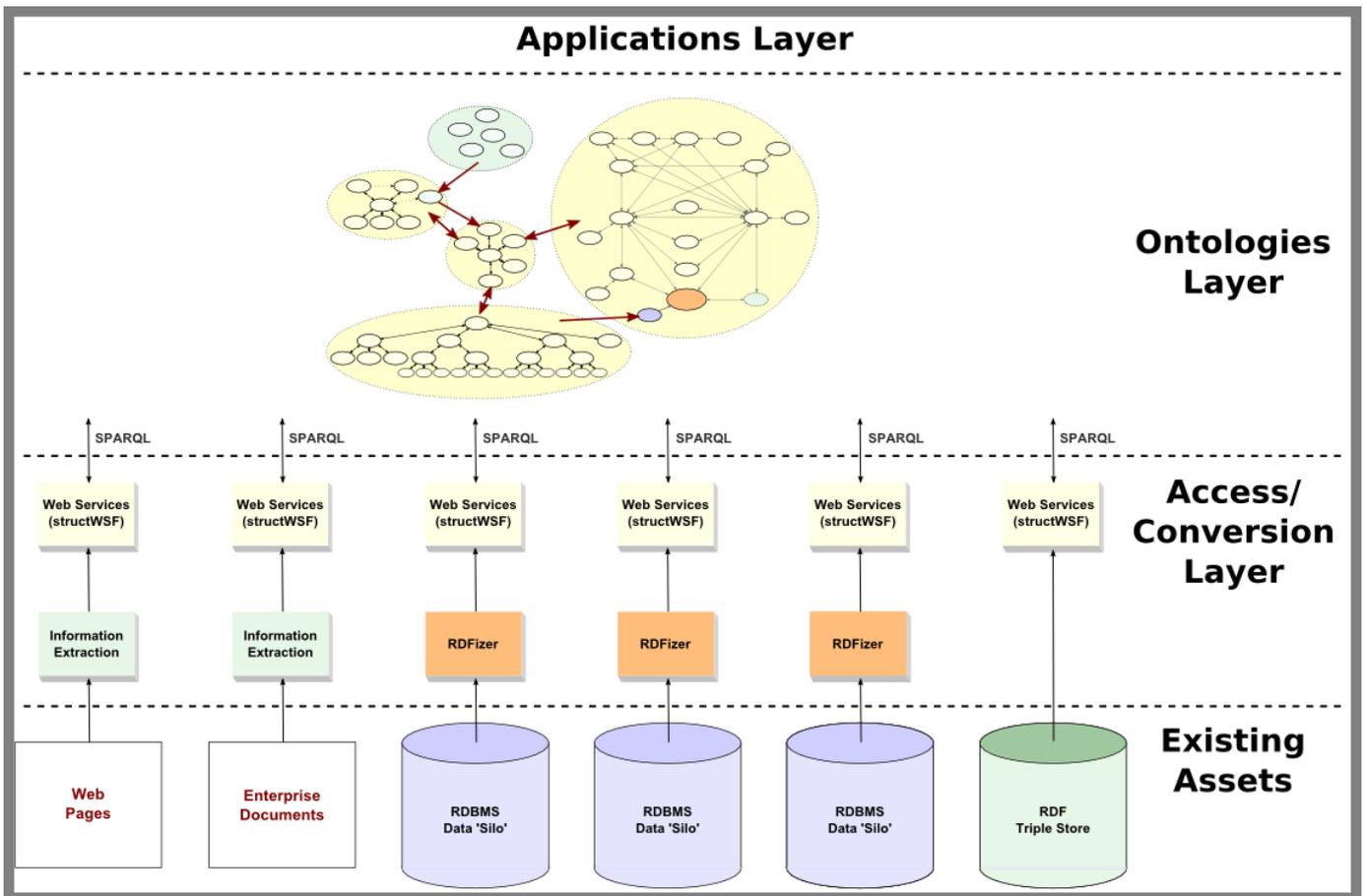
Perspective #2: Augment and Layer on to Existing Assets, Don't Replace Them!

Semantic technology does not change or alter the fact that most activities of the enterprise are transactional, communicative or documentary in nature. Structured, relational data systems for transactions or records are proven, performant and understood. Writing and publishing information, sometimes as documents and sometimes as spreadsheets or Web pages, is (and will remain) the major vehicle for communicating within the enterprise and to external constituents.

On its very face, it should be clear that the *meaning* of these activities -- their *semantics*, if you will -- is by nature an augmentation or added layer to how to conduct the activities themselves. Moreover, as we also know, these activities are undertaken for many different purposes and within many different contexts. The inherent meaning of these activities is also therefore contextual and varied.

This simple truth affirms that semantic technologies are not a starting basis, then, for these activities, but a way of expressing and interoperating their outcomes. Sure, some semantic understanding and common vocabularies at the front end can help bring consistency and a common language to an enterprise's activities. This is good practice, and the more that can be done within reason while not stifling innovation, all the better. But we all know that the budget department and function has its own way of doing things separate from sales or R&D. And that is perfectly OK and natural.

These observations -- in combination with semantic technologies -- can thus lead to a conceptual architecture for the enterprise that recognizes there are "silo" activities that can still be bridged with the semantic layer:



Under this conceptual architecture, "RDFizers" (similar to the [ETL](#) function) or information extractors working upon unstructured or semi-structured documents expose their underlying information assets in RDF-ready form. This RDF is characterized by one or more ontologies (multiples are actually natural and preferred [\[5\]](#)), which then can be queried using the semantic querying language, [SPARQL](#).

We have written at length about [proper separation](#) of instance records and data and schema, what is called the ABox and TBox, respectively, in description logics [\[6\]](#), a key logic premise to the semantic Web. Thus, through appropriate architecting of existing information assets, it is possible to leave those systems in place while still gaining the interoperability advantages of the semantic enterprise.

Another aspect of this information re-use is also a commitment to leverage existing schema structures, be they industry standards, XML, [MDM](#), relational schema or corporate taxonomies. The mappings of these structures in the resulting ontologies thus become the means to codify the enterprise's circumstances into an actionable set of relationships bridging across multiple, existing information assets.

Perspective #3: The First Major Benefit is from Data Federation

Clearly, then, the first obvious benefit to the semantic enterprise is to federate across existing data silos, as featured prominently in the figure above. Data federation has been the Holy Grail of IT systems and enterprises for more than three decades. Expensive and involved efforts from ETL and MDM and then to enterprise information integration ([EII](#)), enterprise application integration ([EAI](#)) and business intelligence ([BI](#)) have been a major focus.

Frankly, it is surprising that no known vendors in these spaces (aside from our own [Structured Dynamics](#), hehe) premise their offerings on RDF and semantic technologies. (Though some claim so.) This is a major opportunity area. (And we don't mind giving our competitors useful tips.)

Perspective #4: Wave Goodbye to Rigid, Inflexible Schema

Instance-level records and the ABox work well with relational databases. Their schema are simple and relatively fixed. This is fortunate, because such instance records are the basis of transactional systems where performance and throughput are necessary and valued.

But at the level of the enterprise itself -- what its business is, its business environment, what is constantly changing around it -- trying to model its world with relational schema has proven frustrating, brittle and inflexible. Though relational and RDF schema share much logically, the physical basis of the relational schema does not lend itself to changes and it lacks the flexibility and malleability of the graph-based RDF conceptual structure.

Knowledge management and business intelligence are by no means new concepts for the enterprise. What is new and exciting, however, is how the emergence of RDF and the semantic enterprise will open new doors and perspectives. Once freed of schema constraints, we should see the emergence of "agile KM" similar to the benefits of agile software development.

Because semantic technologies can operate in a layer apart from the standard data basis for the enterprise, there is also a smaller footprint and risk to experimenting at the KM or conceptual level. More options and more testing and much lower costs and risks will surely translate to more innovation.

Just as semantic technologies are poorly suited for transactional or throughput purposes, we should see the complementary and natural migration of KM to the semantic side of the shop. There are no

impediments for this migration to begin today. In the process, as yet unforeseen and manifest benefits in agility, experimentation, inferencing and reasoning, and therefore new insights, will emerge.

Perspective #5: Data-driven Apps Shift the Software Paradigm

The same ontologies that guide the data federation and interoperability layer can also do double-duty as the specifications for *data-driven applications*. The premise is really quite simple: Once it is realized that the inherent information structure contained within ontologies can guide hierarchies, facets, structured retrievals and inferencing, the logical software design is then to "drive" the application solely based on that structure. And, once *that* insight is realized, then it becomes important, as a best practice, to add further specifications in order to also carry along the information useful for "driving" user interfaces [7].

Thus, while ontologies are often thought solely to be for the purpose of machine interpretation and communication, this double-duty purpose now tells us that useful labels and such for human use and consumption is also an important goal.

When these best practices of structure and useful human labels are made real, it then becomes possible to develop generic software applications, the operations of which vary solely by the nature of the structure and ontologies fed to them. In other words, ontologies now become the application, not custom-written software.

Of course, this does not remove the requirement to develop and write software. But the nature and focus of that development shifts dramatically.

From the outset, data-driven software applications are designed to be responsive to the structure fed them. Granted, specific applications in such areas as search, report writing, analysis, data visualization, import and export, format conversions, and the like, still must be written. But, when done, they require little or no further modification to respond to whatever compliant ontologies are fed to them -- irrespective of domain or scope.

It thus becomes possible to see a relatively small number of these generic apps that can respond to any compliant structure.

The shift this represents can be illustrated by two areas that have been traditional choke points for IT within the enterprise: queries to local data stores (in order to get needed information for analysis and decisions) and report writers (necessary to communicate with management and constituents).

It is not unusual to hear of weeks or months delays in IT groups responding to such requests. It is not that the IT departments are lazy or unresponsive, but that the schema and tools used to fulfill their user demands are not flexible.

It is hard to know just how large the huge upside is for data-driven apps and generic tools. But, this may prove to be of even greater import than overcoming the data federation challenge.

In any event, while potentially disruptive, this prospect of data-driven applications can start small and exist in parallel with all existing ways of doing business. Yes, the upside is huge, but it need not be

gained by abandoning what already works.

Perspective #6: Adaptive Ontologies Flatten, Democratize the KM Process

So, assume, then, a knowledge management (KM) environment supported by these data-driven apps. What perspective arises from this prospect?

One obvious perspective is where the KM effort shifts to become the actual description, nature and relationships of the information environment. In other words, ontologies themselves become the focus of effort and development. The KM problem no longer needs to be abstracted to the IT department or third-party software. The actual concepts, terminology and relations that comprise coherent ontologies now become the foundation of KM activities.

An earlier perspective emphasized how most any existing structure can become a starting basis for ontologies and their vocabularies, from spreadsheets to naïve data structures and lists and taxonomies. So, while producing an operating ontology that meets the best practice thresholds noted herein has certain requirements, kicking off or contributing to this process poses few technical or technology demands.

The skills needed to create these adaptive ontologies are logic, coherent thinking and domain knowledge. That is, any subject matter expert or knowledge worker worth keeping on the payroll has, by definition, the necessary skills to contribute to useful ontology development and refinement.

With adaptive ontologies powering data-driven apps we thus see a shift in roles and responsibilities away from IT to knowledge workers themselves. This shift acts to democratize the knowledge management function and flatten the organization.

Perspective #7: The Semantic Enterprise is 'Open' to the World

Enterprise information systems, particularly relational ones, embody a [closed world assumption](#) that holds that any statement that is not known to be true is false. This premise works well where there is complete coverage of the entities within a knowledge base, such as the enumeration of all customers or all products of an enterprise.

Yet, in the real ("open") world there is no guarantee or likelihood of complete coverage. Thus, under an [open world assumption](#) the lack of a given assertion or fact being available neither implies whether that possible assertion is true or false: it simply is not known. An open world assumption is one of the key factors for enabling adaptive ontologies to grow incrementally. It is also the basis for enabling linkage to external (and surely incomplete) datasets.

Fortunately, there is no requirement for enterprises to make some philosophical commitment to either closed- or open-world systems or reasoning. It is perfectly acceptable to combine traditional closed-world relational systems with open-world reasoning at the ontology level. It is also not necessary to make any choices or trade-offs about using public v. private data or combinations thereof. All combinations are acceptable and easily accommodated.

As noted, one advantage of open-world reasoning at the ontological level is the ability to readily change and grow the conceptual understanding and coverage of the world, including incorporation of external ontologies and data. Since this can easily co-exist with underlying closed-world data, the semantic enterprise can readily bridge both worlds.

Perspective #8: The Semantic Enterprise is a Disruptive Innovation, without Being Disruptive

Unfortunately, as a relatively new area there are advantages for some pundits or consultants to present the semantic Web as more complicated and commitment-laden than it need be. Either the proponents of that viewpoint don't know what they are saying, or are being cynical to the market. The major point underlying the fresh perspectives herein is to iterate that it is quite possible to start small, and do so with low cost and risk.

While it is true that semantic technologies within the enterprise promise some startling upside potentials and disruptions to the old ways of doing business, the total beauty of RDF and its capabilities and this layered model is that those promises can be realized incrementally and without hard choices. No, it is not for free: a commitment to begin the process and to learn is necessary. But, yes, it can be done so with exciting enterprise-wide benefits at a pace and risk level that is comfortable.

The good news about the dedicated issue of the *Cutter IT Journal* and the earlier PWC publication is that the importance of semantic technologies to the enterprise is now beginning to receive its just due. But as we ramp up this visibility, let's be sure that we frame these costs and benefits with the right perspectives.

The semantic enterprise offers some important new benefits not obtainable from prior approaches and technologies. And, the best news is that these advantages can be obtained incrementally and at low risk and cost while leveraging prior investments and information assets.

[1] Paul Horowitz, ed., 2009. *Technology Forecast: A Quarterly Journal*, PricewaterhouseCoopers, Spring 2009, 58 pp. See <http://www.pwc.com/us/en/technology-forecast/spring2009/index.jhtml> (after filling out contact form). I reviewed this publication in an [earlier post](#).

[2] Mitchell Ummell, ed., 2009. "The Rise of the Semantic Enterprise," special dedicated edition of the *Cutter IT Journal*, Vol. 22(9), 40pp., September 2009. See <http://www.cutter.com/offers/semanticerprise.html> (after filling out contact form).

[3] It is really not my purpose to review the *Cutter IT Journal* issue nor to point out specific articles that are weaker than others. It is excellent we are getting this degree of attention, and for that I recommend signing up and reading the issue yourself. IMO, the two useful articles are: John Kuriakose, "Understanding and Adopting Semantic Web Technology," pp. 10-18; and Shamod Lacoul, "Leveraging the Semantic Web for Data Integration," pp. 19-23.

[4] As a working definition, a *semantic enterprise* is one that adopts the languages and standards of the [semantic Web](#), including [RDF](#), [RDFS](#), [OWL](#) and [SPARQL](#) and [others](#), and applies them to the issues of information interoperability, preferably using the best practices of [linked data](#).

[5] One prevalent misconception is that it is desirable to have a single, large, comprehensive ontology. In fact, multiple ontologies, developing and growing on multiple tracks in various contexts, are much preferable. This

decentralized approach brings ontology development closer to ultimate users, allows departmental efforts to proceed at different paces, and lowers risk.

[6] Here is our standard [working definition](#) for [description logics](#):

“Description logics and their semantics traditionally split *concepts* and their relationships from the different treatment of *instances* and their attributes and roles, expressed as fact assertions. The concept split is known as the TBox (for *terminological* knowledge, the basis for *T* in *TBox*) and represents the schema or taxonomy of the domain at hand. The TBox is the structural and intensional component of conceptual relationships. The second split of instances is known as the ABox (for *assertions*, the basis for *A* in *ABox*) and describes the attributes of instances (and individuals), the roles between instances, and other assertions about instances regarding their class membership with the TBox concepts.”

[7] I first introduced this topic in [Ontologies as the 'Engine' for Data-Driven Applications](#). Some of the user interface considerations that can be driven by *adaptive ontologies* include: attribute labels and tooltips; navigation and browsing structures and trees; menu structures; auto-completion of entered data; contextual dropdown list choices; spell checkers; online help systems; etc.