Two Contrasting Styles for the Semantic Enterprise

by Mike Bergman - Monday, February 15, 2010

http://www.mkbergman.com/866/two-contrasting-styles-for-the-semantic-enterprise/

Our Own Approach is Adaptive and Incremental

It is gratifying to see the emergence of the term *semantic enterprise*, with much increased attention and commentary. But, similar to different styles and patterns in software programming, there is not a single (nor best, depending on circumstance) way to approach becoming a semantic enterprise.

In this piece I contrast two styles. The more traditional and familiar one is comprehensive, complete and "engineered" in its approach. The second, and emerging style, is more adaptive and incremental. While Structured Dynamics is a proponent and thought leader for the adaptive style, the use and applicability of either approach is really a function of objectives and circumstances. The choice of approach depends on use case, and should not be a dogmatic one.

Any time a contrast is posed, one should be on guard about setting up a rhetorical strawman. There may perhaps be a bit of this flavor in this article; if so, it is unintended. It is probably best to realize that there is a gradient -- or spectrum -- of possible approaches between these contrasting styles. The real message is to understand these differences such that you can comfortably place your own organization at the right points along this spectrum.

A Spectrum of Advantages and Differences

The general idea of semantics in the enterprise preceeds the use of the term, having been somewhat captured before by the ideas of enterprise application integration, enterprise information integration and other concepts even related to data federation and data warehousing stretching back to the 1980s. However, as a specific label, we can look back to the first mentions in the late 1990s and more concerted attention beginning from about 2002 or so onward [1]. As another indicator, since 2005 the Semantic Technology Conference has given specific prominence to the enterprise [2].

Throughout this period, the sense from academic papers, many vendors, and most pundits [3] has been on things like automated reasoning, machine-aided decision making, aspects of artificial intelligence, and so forth. The general tone is often framed as "revolution" or "massive changes" or something "entirely new." If you are a consultant or software/implementation vendor -- especially where VC money is backing the venture with hopes for big returns and home runs -- it may make cynical sense to sell such large and costly change.

I believe there are circumstances where the *Semantic Enterprise* writ this large may make sense and be financially justified. But, this kind of "big change" view has also seen relatively few visible (or successful) deployments. It has colored what it means to be a semantic enterprise. And, I believe, it has weakened market credibility by perhaps overpromising and underdelivering. The conventional view of what it is be a semantic enterprise deserves to be balanced.

So, as we balance this understanding of the semantic enterprise to one that is more nuanced, we can contrast the characteristics of the two apposite styles as follows:

Characteristics of the *Comprehensive, 'Engineered'* Style	**Characteristics of the *Adaptive, Incremental* Style**
• A focus on a more complete, comprehensive coverage of the semantics in the domain • More enterprise-wide, less partial or departmental • Greater emphasis on "closed world" approaches [4]; more akin to relational database architecting and schema • Expansion is possible, but effort may be somewhat complex • A general implication is to replace or supplant existing information structures with semantic ones • Not necessarily based on semantic Web standards and languages [5] (*e.g.*, may include Common Logic, frame logics, etc.) • Richer set of predicates (relations) • Though a distinction is maintained between schema and instances, their separation may not be consistently (physically) enforced • Often more complicated inferencing and logic tests • More complete enumeration and characterization of items • Much process around semantics agreement across groups	• An emphasis on a simpler, incremental, "learn as you go" approach • Start with single departments or limited vertical apps • Embedded in the "open world" approach [4], with incorporation of external information • Design and approach inherently allows incremental expansion and adaptation • A key premise is to build from and leverage existing information structures, vocabularies and assets • Fully based on semantic Web standards and languages [5], often including linked data [6] • Tends to start simply with hierarchical or related concepts (*e.g.*, SKOS) • Conscious distinction in the structure for handling schema separate from instances [7] • Inferencing logic based more on concept matching, or parent-child or part-of relationships • Degree of item characterization based on current scope • Initial semantic matching can be driven from existing assets • Fairly well-developed implementation

• Fairly well-developed implementation tools, including for ontology engineering • Implementation times in months to years • Implementation costs akin to traditional large-scale IT projects	tools, *except* for how to engage publics in the development process • Implementation times in weeks to months • Implementation costs driven by available budgets (and thus scope)

Note we have labeled the conventional approach as the "comprehensive, engineering" style; its contrast, and the one we position more closely to, is the "adaptive, incremental" style.

[Others have posited contrasting styles, most often as "top down" *v.* "bottom up." However, in one interpretation of that distinction, "top down" means a layer on top of the existing Web [8]. On the other hand, "top down" is more often understood in the sense of a "comprehensive, engineered" view, consistent with my own understanding [9]. Yet no matter which characterization, neither captures what I feel to be the more important considerations of mindset, logic and premise.]

Though the table above contrasts many points, I think there are two main distinctions to the adaptive approach. First, it firmly embraces the open world assumption. OWA is key to an incremental, "learn as you go" deployment that is also well suited to incorporation of external information. The second main distinction is to leverage and build from existing assets.

A Spectrum of Applications

Yet as noted in the opening, which of these approaches makes better sense depends on circumstance. One aspect of circumstance is available budget and deployment times for pilots or proofs-of-concept. Another aspect, of course, is the planned use or application for the deployment.

These are by no means hard distinctions, but in general we can see these contrasting approaches applying to the following uses:

Applications and Uses for the *Comprehensive, 'Engineered'* Style (*i.e.*, more CWA driven)	**Applications and Uses for the *Adaptive, Incremental* Style** (*i.e.*, more OWA driven)
• Bounded, "inward" applications (high degree of control and completeness) • Engineering enterprises • Technical domains and organizations • Aeronautics • Pharmaceuticals • Chemicals • Petroleum • Energy • A/E firms (construction)	• External facing applications, organizations (customers, incorporation of external data) • Faceted Search • Taxonomy updates • Multi-domain master data management (MDM) • Simple (initially) inferencing • Consumer products • Finance • Health care • Knowledge enterprises

A critical distinction is the nature of the enterprise itself. "External-facing" enterprises or functions that want or need to incorporate much external information (say, marketing or competitive intelligence) are advised to look closely at the adaptive approach. Organizations that have more complete control over their circumstances should perhaps focus on the conventional approach.

Adoption Thresholds and Risks

In previous writings I have pointed to the manifest benefits that can accrue to the semantic enterprise [see, esp. 10]. But we also have witnessed nearly a decade of promotion for semantics in the enterprise, with perhaps a lack of progress in some areas or unmet promises in others. These raise questions and skepticism of the real eventual costs and benefits.

I believe some of this skepticism is inherent with anything new -- the general IT fatigue from what the current "next great thing" might be. But I also believe that some of this skepticism results from an approach to semantics in the enterprise that is both lengthy to deploy and high cost.

The key advantage of the adaptive, incremental approach is that the whole IT game in the enterprise can change. An open world approach enables adoption as it proves itself and as budgets allow. Commitments made under this approach have, in essence, permanent value. Past fears and concerns about making "wrong" bets no longer apply. With learning, targets can be re-adjusted, structure re-defined and applications re-focused, all as new discoveries and broadening scope dictate.

This does not make the adaptive approach better than the conventional one. But, it does make it less risky and, well, more *adaptive*.

---

[1] For example, the earliest Google mentions on "semantic enterprise" date to about 1998 or 1999. In 2002, the University of Georgia and Amit Sheth offered the first known academic course on the Semantic Enterprise; see http://lsdis.cs.uga.edu/SemanticEnterprise/.

[2] See the conference guide for the Semantic Technology Conference 2005. The sixth one, the 2010 Semantic Technology Conference, is upcoming on June 21-25 in San Francisco.

[3] See, for example, Mitchell Ummell, ed., 2009. “The Rise of the Semantic Enterprise,” special dedicated edition of the *Cutter IT Journal*, Vol. 22(9), 40 pp., September 2009. See http://www.cutter.com/offers/semanticenterprise.html (after filling out contact form). Partially in response to this conventional view, I wrote [10]. In that article I offered as a working definition that "*a **semantic enterprise** is one that adopts the languages and standards of the* semantic Web *. . . and applies them to the issues of information interoperability, preferably using the best practices of* linked data." That happens to be Structured Dynamics' preferred definition, though as this posting indicates, there is a spectrum of definitions of the term.

[4] See, M.K. Bergman, 2009. “The Open World Assumption: Elephant in the Room“, *AI3:::Adaptive Information* blog, December 21, 2009.

[5] See for example RDF, RDFS, OWL , SKOS and SPARQL and others.

[6] Linked data is a set of best practices for publishing and deploying instance and class data using the RDF data model. Two of the best practices are to name the data objects using uniform resource identifiers (URIs), and to expose the data for access via the HTTP protocol. Both of these practices enable the Web to become a distributed

database, which also means that Web architectures can also be readily employed.

[7] We use a basis in description logics for defining the roles and splits in schema and instances. As we define it:

“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.”

[8] One article that got quite a bit of play a few years back was A. Iskold, 2007. "Top Down: A New Approach to the Semantic Web," in *ReadWrite Web*, Sept. 20, 2007. The problem with this terminology is that it offers a completely different sense of "top down" to traditional uses. In Iskold's argument, his "top down" is a layering on top of the existing Web.

[9] The more traditional view of "top down" with respect to the semantic Web is in relation to how the system is constructed. This is reflected well in a presentation from the NSF Workshop on DB & IS Research for Semantic Web and Enterprises, April 3, 2002, entitled "The 'Emergent, Semantic Web: Top Down Design or Bottom Up Consensus? ". Under this view, top down is design and committee-driven; bottom up is more decentralized and based on social processes, which is more akin to Iskold's "top down."

[10] M.K. Bergman, 2009. "Fresh Perspectives on the Semantic Enterprise," *AI3:::Adaptive Information* blog, Sept. 28, 2009.