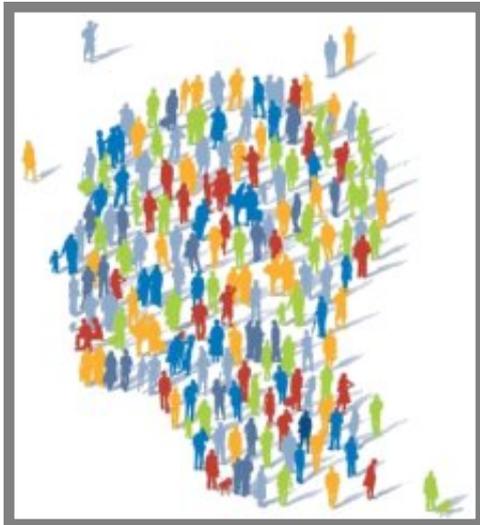


Democratizing Information with Semantics

by Mike Bergman - Monday, April 04, 2011

<http://www.mkbergman.com/953/democratizing-information-with-semantics/>



Self-service Information Management for Knowledge Workers

Though I have alluded to it numerous times in my past writings [\[1\]](#), I think one of the most pervasive and important benefits from semantic technologies in the enterprise will come from the democratization of information. These benefits will arise mostly from a fundamental change in how we manage and consume information. A new "system" of semantic technologies is now largely available that can put the collection, assembly, organization, analysis and presentation of information directly in the hands of those who need it most -- the consumers of information.

The idea of "democratizing information" has been around for a couple of decades, and has accelerated in incidence since the dominance of the Internet. Most commonly, the idea is associated with developments and notions in such areas as [citizen journalism](#), [crowdsourcing](#), the [wisdom of the crowd](#), [social bookmarking](#) (or collaborative tagging), and the democratic (small "d") access to publishing via new channels such as [blogs](#), [microblogs](#) (e.g., Twitter) and [wikis](#). To be sure, these kinds of democratic information will (and are) benefiting from the use and application of semantics.

But the trend I'm focusing on here is much different and quite new. It is the idea that enterprise knowledge workers can now take ownership and control of their knowledge management functions. In the process, prior bottlenecks due to IT can be relieved and massive new benefits can open up to the enterprise.

Decades-long Mismatches Between KM and IT

"Enterprise systems are doing it wrong. And not just a little bit, either. Orders of magnitude wrong. Billions and billions of dollars worth of wrong. Hang-our-heads-in-shame wrong. It's time to stop the madness."

It is no secret that IT has not served the enterprise knowledge management function well for decades. Transaction systems and database systems geared to fast indexing and access to datum have not proved well suited to information or [knowledge management](#). KM includes such applications as [business intelligence](#), [data warehousing](#), [data integration](#) and [federation](#), [enterprise information integration](#) and [management](#), [competitive intelligence](#), [knowledge representation](#), and so forth. [Information management](#) is a bit broader category, and adds such functions as [document management](#), [data management](#), [enterprise content management](#), enterprise or [controlled vocabularies](#), [systems analysis](#), [information standards](#) and information assets management to the basic functions of KM. Since the purpose of this piece is not to get into the epistemological differences between information and knowledge, I use these terms more-or-less interchangeably herein.

Knowledge and information management is very big business. Given the breadth and differences in defining the KM and IM markets, let's take as a proxy the business intelligence (BI) market, one of KM's most important elements. Various estimates from IDC, Gartner and others place the current value of BI software sales somewhere in the range of \$9 billion to \$11 billion annually [3]. Further, BI ranked number five on the list of the top 10 technology priorities for chief information officers (CIOs) in 2011. And this pertains to the structured component of information alone.

Yet, at the same time, BI-related projects continue to have high failure rates, often cited as in the 65% to higher range [4]. These failure rates are consistent with KM projects in general [5]. These failures are merely one expression of a constant litany of issues and concerns regarding the enterprise KM function:

Conventional KM Problem Area	Comments
Inflexible Reports	<ul style="list-style-type: none"> • reports are rarely "self-service" • new requests need to be placed in queue • 90% of stored report templates are never used • unlimited "slicing and dicing" not available
Inflexible Analysis	<ul style="list-style-type: none"> • analysis is rarely "self-service" • new requests need to be placed in queue • many requests not accepted due to schema rigidities, cascading changes needed • analysis options are "pre-canned", inflexible
Schema Bottlenecks	<ul style="list-style-type: none"> • brittleness of relational data model and typical star schema

	<ul style="list-style-type: none"> • crossing across schema or databases difficult • load and re-indexing cycles can limit access, impose expensive back-end requirements • can not (often) accommodate new data, structures
ETL Bottlenecks	<ul style="list-style-type: none"> • getting data into the system needs to be placed in queue • new external data requires extract, transform and load (ETL) routines to be written • schedule and update cycles can be a mismatch to access needs
Reliance on Intermediaries	<ul style="list-style-type: none"> • all problems above work through intermediaries • disconnect between those with need and decision-makers and those who implement the solutions • inherent issues in communicating requirements to implementers • related time delays to implementation exacerbate the communication of requirements
Specialized Expertise Required	<ul style="list-style-type: none"> • expertise and skill sets needed to implement solutions different from those of the knowledge consumer • inherent issues in communicating requirements to implementers • high costs for attracting necessary expertise • expertise is inherently an overhead function
Slow Response Time	<ul style="list-style-type: none"> • all problems above lead to delays, slow response • timely communications, analysis, decisions suffer • delays mean knowledge management is not an active "contact sport", becomes mired and unresponsive • some needs are just not requested because of these problems
Dependence on External Apps	<ul style="list-style-type: none"> • new apps need to be identified, procured

	<ul style="list-style-type: none"> • design and configuration of apps requires external expertise, programming skills • multiple sourcing of apps leads to frequent incompatibilities, high costs for integration, poor interoperability
Unmet Needs	<ul style="list-style-type: none"> • many KM needs are simply not requested • by the time responses are forthcoming, needs and imperatives have moved on • communications, analysis and decisions become hassles • the "contact sport" of active discovery and learning is unmet
High Opportunity Costs	<ul style="list-style-type: none"> • many KM insights are simply not discovered • delays and frustration adds to costs, friction, inefficiencies • no way to know the opportunity costs of what is not learned -- but, surely is high
High Failure Rates	<ul style="list-style-type: none"> • the net impact of all of the problems above is to lead to high failure rates (~60% to 70%) and unacceptable costs • reliance on IT for KM has utterly and totally failed

The seeming contradiction between continued growth and expenditures for information management coupled with continued high failure rates and disappointments is really an expression of the centrality of information to the modern enterprise. The funding and growth of the IT function is itself an expression of this centrality and perceived importance. These have been abiding trends in our transition to [information](#) or [knowledge economies](#).

Bray [2] places the fault for wasted initiatives within the culture of IT. I believe there is some truth to this -- variably, of course, depending on the specific enterprise. But the real culprit, I believe, has been the past need to "[intermediate](#)" a layer of software and IT expertise between knowledge workers and their source information. A progression of tasks has been necessary -- conducted over decades with advances and learning -- to get paper information into electronic form, get those forms to be understood and operate in some common ways, and then to develop tools, architectures and frameworks to make sense of it. Yet, as more tasks with required specialized skills have been added to this layer, the actual gulf between worker and information has increased. For example, enterprises still require the overhead and layers of IT to write SQL to get information out and then to prepare and fix reports.

On average, IT now consumes about 4% of all enterprise expenditures and employs about 6% of

enterprise workers [6]. IT has become a very thick intermediary layer, indeed! Yet, because of the advances and learning that has occurred in growing and nurturing this layer, we also now have the basis to begin to "disintermediate" the IT layer. Many, if not all, of the challenges noted in the table above can be improved by doing so.

Early Attempts at Self-service and Semantics

One current buzzword in business intelligence is "self service". By this term is meant giving knowledge workers the tools and systems for creating reports or doing analysis on their own without needing to work through (or be frustrated by) the IT layer. [Self-service software](#) was first postulated in the 1990s as a way for information consumers and authors (typically subject-matter experts) to automate some of their knowledge management tasks. Today, it is most commonly applied to self-service reporting or self-service analytics within the BI realm.

As a general proposition, self-service BI has been more myth than reality [7]. Forrester surveys, for example, indicate that IT still develops most BI applications. Of survey respondents in 2009, 70% responded that IT develops the enterprise's reports and dashboards [8]. However, that figure is not 100%, as it was just a decade earlier, and there is also notable success to some open source providers such as [BIRT](#) that address a wide range of reporting needs within a typical application, ranging from operational or enterprise reporting to multi-dimensional online analytical processing ([OLAP](#)).

James Kobelius [8] is particularly bullish on the application of [Web 2.0 "mashup"](#) applications to knowledge worker purposes. Under this approach, Web-based applications are used and accessed directly by knowledge workers for charting and mapping purposes using [Ajax](#) or [Flash](#) widgets, such as [Google Maps](#). The conventional BI and KM vendors have begun to more more aggressively into this area. Some notable new entrants -- such as [Tableau](#), [Factual](#) or [Good Data](#) -- are also showing the way to more direct access, more flexible reporting and analysis widgets, and cleaner service or platform designs.

These initiatives reside at the display or reporting level. There is another group, including James Kobelius, [Neil Raden](#) or [Seth Earley](#), that have addressed how to get disparate information to talk together using [ontologies](#). They refer to "semanticizing" such traditional practices such as master data management ([MDM](#)), "ontologizing" [taxonomies](#), or adding Web 2.0 mashups to business intelligence. While these thoughts are moving in the right direction, and will bring incremental benefits, they still are far short of the potentials at hand.

Self-service Information Management

[_\(click to expand\)](#)

So far, in the KM realm, the application of semantics has tended to be limited to information extraction (tagging) of text documents and first attempts at using ontologies. The tagging component is essential to enable the 80% of information presently in textual documents to become first-class citizens within business intelligence or knowledge management. The ontology efforts to date appear to be more like thin veneers over traditional taxonomies. Rather than hierarchical structures, we now see graph-oriented ones, but still intended to fulfill the same tasks of enterprise metadata and vocabulary lookups.

The ontology efforts especially are just nibbling around the edges of what can be done with semantic

technologies. Rather than looking upon ontologies as just another dictionary (though that role is true), if we re-orient our thinking to make ontologies central to the KM function, a wealth of new opportunities and benefits arises.

A bit more than a year ago, we formulated the [Seven Pillars of the Open Semantic Enterprise](#), which included ontologies and related as some of the central components. In that article [9], we noted the particular applicability of semantic technologies to the information and knowledge management functions within enterprises. We asserted the benefits for embracing the open semantic enterprise as providing the organization greater insights with lower risk, lower cost, faster deployment, and more agile responsiveness. Since that time we have been deploying such systems and documenting those benefits.

Integral to the seven pillars are those aspects that lead to the democratization of information for the knowledge worker, what combined might be called "self-service information management". As the figure to the right shows, three of the seven pillars are essential building blocks to this capability, two pillars are further foundations to it, with the remaining two pillars only tangentially important.

What the combination of these pieces means is a fundamental change in how knowledge work is done. Through this approach, we can largely disintermediate IT from the knowledge function, can bring knowledge management directly into the hands of those who need it in real time, and fundamentally alter how knowledge management apps are designed and deployed. The best thing is these benefits are an incremental evolution, and retain the use and value of existing information assets.

Building Block #1: Adaptive Ontologies

Rather than peripheral lookup structures or thin veneers, ontologies play *the central role* in the design of self-service information management. We use the plural on purpose here: what is deployed is actually a library of complementary and modular ontologies that play a variety of roles. Combined, we call these libraries with their representative functions *adaptive ontologies*.

This library contains the expected and conventional domain ontologies. These represent the actual knowledge space for the domain at hand, and may be comprised of multiple different ontologies representing different domain or knowledge spaces. These standard semantic Web ontologies may range from the small and simple to the large and complex, and may perform the roles of defining relationships among concepts, integrating instance data, orienting to other knowledge and domains, or mapping to other schema.

From a best practices standpoint [10], we take special care in constructing these domain ontologies such that we provide labels and cues for user interfaces. 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. We also include a variety of synonyms and aliases (the combination of which we call *semsets*) for referring to concepts and instances in multiple ways and for aiding information extraction and tagging functions. (In addition to organizing and helping to interoperate contributing information, these domain ontologies are also used for what is called ontology-based information extraction ([OBIE](#)) via our scones [11] system.)

In addition the library of adaptive ontologies includes some administrative ontologies that guide how instance data can be imported and inter-related (via the Instance Record Object Notation, or [irON](#)); what information types drive what widgets (via the Semantic Component Ontology, or [SCO](#)); data mapping vocabularies ([UMBEL Vocabulary](#)); how to characterize datasets; and other potential specialty functionality.

A forthcoming article will describe the composition and modularity typically found in a library of these adaptive ontologies.

In combination, these adaptive ontologies are, in effect, the "brains" of the self-service system. The best aspect of these ontologies is that they can be understood, created and maintained by knowledge workers. They constitute the only specification (other than theming, if desired) necessary to create self-service knowledge management environments.

Building Block #2: Ontology-driven Apps

The piece of the puzzle that implements the instruction sets within these adaptive ontologies are the ontology-driven apps, or *ODapps*. A recent article describes these structures in some detail [\[12\]](#).

ODapps are modular, generic software applications designed to operate in accordance with the specifications contained in the adaptive ontologies. ODapps fulfill specific generic tasks, consistent with their dedicated design to respond to adaptive ontologies. For example, current ontology-driven apps include imports and exports in various formats, dataset creation and management, data record creation and management, reporting, browsing, searching, data visualization and manipulation (through libraries of what we call *semantic components*), user access rights and permissions, and similar. These applications provide their specific functionality in response to the specifications in the ontologies fed to them.

ODapps are designed more similarly to widgets or API-based frameworks than to the dedicated software of the past, though the dedicated functionality (*e.g.*, graphing, reporting, etc.) is obviously quite similar. The major change in these ontology-driven apps is to accommodate a relatively common abstraction layer that responds to the structure and conventions of the guiding ontologies. The major advantage is that single generic applications can supply shared functionality based on any properly constructed adaptive ontology.

Generic functionality included in these ODapps are things like filtering, setting value ranges, choosing the specific display view, and invoking or not various display templates (akin to the [infoboxes](#) on Wikipedia). By nature of the data and the ontologies submitted to them, the ODapp signals to the user or consumer what displays, views, filters or slices-and-dices might be available to them. Fed different data and different ontologies, the ODapp would signal the user differently.

Because of their generic design, *driven* by the ontologies, only a relatively small number of ODapps needs to be created. Once created with appropriate generic functionality, application development is essentially over. It is through the additions and changes to the adaptive ontologies -- done by knowledge workers themselves -- that new capability and structure gets exposed through these ontology-driven apps. This innovation shifts the locus from software and programming to data and knowledge structures.

This democratization of IT means that everything in the knowledge management realm can become self service. Users and consumers can create their own analyses; develop their own reports; and package and disseminate what they and their colleagues need, when they need it. Through ontology-driven apps and adaptive ontologies, we turn prior software engineering practice on its head.

Building Block #3: Open World Assumption

Integral to this design is the embrace of the [open world assumption](#) [13]. Though not a specific artifact, as are adaptive ontologies or ODapps, the open-world approach is the logical underpinning that allows consumers or knowledge workers to add new information to the system as it is discovered or scoped. This nuance may sound esoteric, but traditional KM systems have a very different underpinning that leads to some nasty implications.

Because the predominant share of KM systems are based on relational database systems, they embody a [closed-world design](#). This works well for transaction systems or environments where the information domain is known and bounded, but does not apply to knowledge and changing information. Moreover, the schema that govern closed-world designs are brittle and hard to change and manage. It is this fact that has put KM squarely in the bailiwick of IT and has often led to delays and frustrations. Re-architecting or adding new schema views to an existing closed-world system can be fiendishly difficult.

This difficulty is a major reason why IT resists casual or constant changes to underlying data schema. Unfortunately, this makes these brittle schema difficult to extend and therefore generally unresponsive to changing and growing knowledge. As an environment for knowledge management, the relational data system and the closed-world approach are lousy foundations.

Other Building Blocks

As the self-service information management diagram above shows, RDF and Web services are two further important foundations. RDF ([Resource Description Framework](#)) is the canonical data model upon which all input information is represented. This means that the ODapp tools and the adaptive ontologies can work off a single model of knowledge representation. The Web service and architecture component is also helpful in that it allows Web 2.0 technologies to be brought to bear and allows distributed sources and users for the KM system. This provides scalability and distributed applicability, including on smartphones.

The other two pillars of the open semantic enterprise -- the layered approach and [linked data](#) -- are also helpful, but not necessarily integral to the KM and self-service perspectives presented herein.

Benefits from Self-service Information Management

The benefits and flexibilities from self-service information management extend from top to bottom; from creating data and content to publishing and deploying it. Here is a listing of available potentials for self-service, drawing comparison to the current conventional approach dependent on IT:

Information Activity	Conventional Approach (IT)	Self-service Information
----------------------	----------------------------	--------------------------

		Management
Creating	<ul style="list-style-type: none"> • structured data only • not generally available directly to the knowledge worker 	<ul style="list-style-type: none"> • can create own datasets • can extract and transform own datasets • can tag and integrate non-structured (text + document) information • able to handle unstructured, semi-structured and structured data alike
Annotating	<ul style="list-style-type: none"> • not generally provided 	<ul style="list-style-type: none"> • completely open, flexible • can define own annotation fields, annotation schema (approaches)
Analyzing	<ul style="list-style-type: none"> • pre-canned functions • structure pre-defined • slow performance 	<ul style="list-style-type: none"> • all structural dimensions can be filtered • all values and ranges thereof can be filtered • multiple analysis display widgets selectable depending on the type of input data • real-time configuration • fast (nearly instantaneous) performance • provision of (nearly) real-time analytics • additional capabilities in inferencing and reasoning • modeling and understanding of complex graph and relationships structures (<i>e.g.</i>, social networks)
Reporting	<ul style="list-style-type: none"> • pre-canned templates or report writers • structure pre-defined 	<ul style="list-style-type: none"> • user-definable templates • templates automatically assignable by types of thing being reported • embeddable in Web pages, alternate presentation media • styling and theming flexibility
Visualizing	<ul style="list-style-type: none"> • very little done through IT 	<ul style="list-style-type: none"> • variety of visualization widgets available (<i>e.g.</i>, maps, charts, graphs, networks)

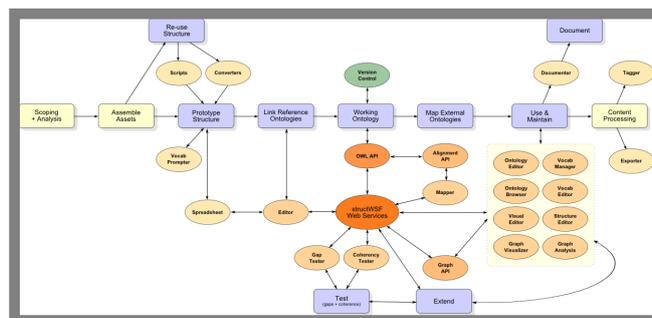
		<ul style="list-style-type: none"> • large-scale systems views possible • visual interactions (a la Web 2.0) possible
Collaborating	<ul style="list-style-type: none"> • very little done through IT • collaboration, if done, is via separate social media 	<ul style="list-style-type: none"> • completely open • variable access and permission rights by user or group • built-in to the entire infrastructure
Validating	<ul style="list-style-type: none"> • not directly done by knowledge worker • user input, if done, via problem tickets with delays 	<ul style="list-style-type: none"> • can be integrated into the business process or workflow • "soft" validations and ratings/rankings can also be included • consistency checking • satisfiability checking
Publishing	<ul style="list-style-type: none"> • limited to pre-canned reports 	<ul style="list-style-type: none"> • any report or analysis is available for publishing • documents and images and widget displays are available for publishing • multiple export formats means information, slices thereof, or analysis results thereof can be embedded and integrated into multiple presentation media
Re-purposing	<ul style="list-style-type: none"> • none directly by the knowledge worker 	<ul style="list-style-type: none"> • any report or analysis is available for re-purposing • documents and images and widget displays are available for re-purposing • canonical internal representations (RDF and XHTML) means available information can be deployed for a variety of purposes (Web pages, reports, documents, slide shows, etc.)
New Functionality	<ul style="list-style-type: none"> • none known, if not already listed 	<ul style="list-style-type: none"> • semantic querying • data visualization • text mining and tagging

		<ul style="list-style-type: none"> • categorization • graph mining • logic checking
Developing Apps	<ul style="list-style-type: none"> • none via the official systems by the knowledge worker • if done, via guerrilla apps 	<ul style="list-style-type: none"> • only generic apps needed • many fewer and more flexible apps push issue into the background
Dashboarding	<ul style="list-style-type: none"> • not available to most systems • if available, limited number of pre-canned options 	<ul style="list-style-type: none"> • any report or analysis is available for dashboarding • any widget is available for dashboarding • complete structure (typing, values, sources) available for filtering, "slicing and dicing" • all dashboard objects on a given canvas are linked, interoperate (selections in one widget reflected in other widgets) • dashboards may be made persistent for re-use, springboarding new dashboards (as templates)

The fact that any source -- internal or external -- or format -- unstructured, semi-structured and structured -- can be brought together with semantic technologies is a qualitative boost over existing KM approaches. Further, all information is exposed in simple text formats, which means it can be readily manipulated and managed with easy to understand tools and applications. Reliance on open standards and languages by semantic technologies also leads to greater use and availability of open source systems.

In short, self-service information management approaches should be cheaper, faster, more responsive and more capable than current approaches.

Great Progress, with Ontology Management the Next Challenge



[\(click to expand\)](#)

Given these perspectives, hearing someone tout *data-driven applications* or advocate ontologies merely for metadata matching sounds positively Neanderthal. The prospects we have with semantic technologies, ontology-driven apps, and self-service information management systems mean so much more. The prospect at hand is to remake the entire knowledge management function, in the process bringing all aspects from creating and distributing knowledge products into the direct hands of the user. This is truly the *democratization of information!*

The absolutely fantastic news is none of this is theoretical or in the future. All pieces are presently proven, working and in hand. This is a practical vision, ready today.

Granted, like any new innovation, especially one that is infrastructural and systems-oriented, there are some weak or less-developed parts. These current gaps and needs include:

- Though tools exist, the state of ontology create, edit, manage, update, delete, map and validate tools could be greatly improved [14]. As the central drivers for ODapps, a simplification of tasks geared more to the knowledge worker, and not professional ontologists, is needed (see diagram to right for some of the needed functions). Some of these developments are underway, with more desired
- A relatively complete starting set of about 20 ODapps widgets is presently available. However, more are needed and for different deployment environments. BI analysis remains one weak area, as is an Ajax-based library
- The number of infobox templates is small, and better (WYSIWYG or graphical) create and manage utilities would be most useful, and
- User permission and authorization protocols exist, but are IP-based at present and could be beneficially expanded for different environments and use cases.

Yet, in the grand scheme of things, these gaps are relatively insignificant. The path and general architecture and design for moving forward are now clear.

Self-service information management via appropriately designed semantic technologies is now a reality. It promises to fulfill a vision of information access and control that has been frustrated for decades. We think these are exciting developments for the enterprise -- and for the individual knowledge hound. We welcome your [inquiries](#) and invite you to join our [open OSF group](#) to contribute your ideas.

[1] Including going all the way back to my description of purpose for this blog back in 2005; see the AI3 [Blogasbörd](#) where I state, "One of my central arguments [in this blog] is that an inexorable trend through history has been the '**democratization**' of **information**."

[2] Tim Bray, 2010. "[Doing it Wrong](#)," on his blog, January 5, 2010. The extensive comments are also worth a read.

[3] According to [Marketwire](#) quoting IDC, "Preliminary market sizing suggests that the business intelligence tools software market grew 2.6% in 2009 to reach \$8.1 billion. Given the current market assumptions regarding the global economy and demand drivers in the BI tools software market, IDC forecasts this market to grow at a compound annual growth rate of 6.9% through 2014 to \$11.3 billion." [CBR](#), citing Gartner, indicates the worldwide BI software market will grow 9.7 percent, reaching US\$10.8 billion in 2011. Gartner also said BI platforms would continue to be one of the fastest growing software markets. For a very good background on BI, see Rochelle Shaw, 2011. "[What is Business Intelligence](#)," posted in [Database Trends and Applications](#), January 7, 2011.

[4] According to this article, by Antone Gonsalves, [Poor Use Of Data Integration Tools Can Waste \\$500,000 Annually](#): Gartner (April 27, 2009), which reports on a recent Gartner Report, large global 2000 companies, using several data integration tools with overlapping features, can reduce costs by more than \$500,000 annually by eliminating redundant software and leveraging a shared services model. In a further report by Roman Stanek, [Business Intelligence Projects are Famous for Low Success Rates, High Costs and Time Overruns](#) (April 25, 2009), Gartner is talking about a dirty little secret in the world of data integration, the fact that the data integration technology in place is based on generations of data integration technology being layered in the enterprise over the years. Thus, technology that was purchased to solve data integration problems, and reduce costs, is actually making the data integration problem more complex and no longer cost efficient.

[5] For example, see Roger Sessions, 2009. [Cost of IT Failure](#), September 28, 2009. This analysis suggests failure rates of 65% with a total estimated worldwide cost of \$6.2 trillion in 2009. Commenters have raised questions as to what constitutes failure and have questioned some of the analysis assumptions. Nonetheless, even with over-estimates, the scale of the numbers is alarming; see Jorge Dominguez, 2009. [The CHAOS Report 2009 on IT Project Failure](#), June 16, 2009, which indicates combined failure and challenge rates for IT projects have ranged from 65% to 84% over the period 1994 to 2009; see http://www.education.state.pa.us/portal/server.pt/gateway/PTARGS_0_2_690719_0_0_18/CHAOS%20Summary%202009.pdf. Also see Dan Galorath, 2008. [Software Project Failure Costs Billions; Better Estimation & Planning Can Help](#), June 7, 2008. In this report, Galorath compares and combines many of the available IT failure studies and summarizes that 3 of 5 IT projects do not do what they were supposed to for the expected costs, with 49% showing budget overruns, 47% showing higher than expected maintenance costs, and 41% failing to deliver expected business value; the anecdotal failure rate for years for IT projects has been claimed as 80%, with business intelligence and data warehousing particularly failure-prone areas; in 2001, a study by Mark N. Frolick and Keith Lindsey, [Critical Factors for Data Warehouse Failures](#), for the Data Warehousing Institute noted conventional wisdom says the failure rate of data warehousing projects is 70 to 80 percent, with a then-recent study in the insurance industry found a 90-percent failure rate. This report is useful for combining many historical studies.

[6] As taken from the Gartner [IT Metrics: IT Spending and Staffing Report, 2010](#); see <http://www.slideshare.net/dellenterprise/it-spending-and-staffing-report-2010>.

[7] Wayne W. Eckerson, 2007. "The Myth of Self-Service Business Intelligence," in *TDWI Online*, October 18, 2007; see <http://tdwi.org/articles/2007/10/18/the-myth-of-selfservice-bi.aspx>. "Business Intelligence projects are famous for low success rates, high costs and time overruns. The economics of BI are visibly broken, and have been for years. Yet BI remains the #1 technology priority according to Gartner."

[8] See James G. Kobiellus, 2009. [Mighty Mashups: Do-It-Yourself Business Intelligence For The New Economy](#), July 23, 2009, see <http://www.corda.com/pdfs/mighty-mashups-article.pdf>. In this report, Kobiellus, the lead author from a Forrester study (August 2008, *Global BI And Data Management Online Survey*) that surveyed 82 IT decision-makers, noted that just over 70% responded that IT develops their reports and dashboards. About 57% responded that power users did such development. Only 18.3% reported that BI development is done by end users with limited BI skills. .

[9] M.K. Bergman, 2010. "Seven Pillars of the Open Semantic Enterprise," in *AI3:::Adaptive Information* blog, January 12, 2010; see <http://www.mkbergman.com/859/seven-pillars-of-the-open-semantic-enterprise/>.

[10] There are a series of ongoing ontology best practices articles; see <http://www.mkbergman.com/category/ontology-best-practices/>.

[11] The *scones* (Subject Concept Or Named Entities) tagger provides information extraction of domain-specific

subject concepts and entities from unstructured text. It also provides disambiguation of this information based on the context of the source information. See further <http://techwiki.openstructs.org/index.php/Category:Scones>.

[12] M.K. Bergman, 2011. "Ontology-Driven Apps Using Generic Applications," in *AI3::Adaptive Information* blog, March 7, 2011; see <http://www.mkbergman.com/948/ontology-driven-apps-using-generic-applications/>.

[13] M.K. Bergman, 2009. "The Open World Assumption: Elephant in the Room," in *AI3::Adaptive Information* blog, December 21, 2009; see <http://www.mkbergman.com/852/the-open-world-assumption-elephant-in-the-room/>. The open world assumption (OWA) generally asserts that the lack of a given assertion or fact being available does not imply whether that possible assertion is true or false: it simply is not known. In other words, lack of knowledge does not imply falsity. Another way to say it is that everything is permitted until it is prohibited. OWA lends itself to incremental and incomplete approaches to various modeling problems.

[14] M.K. Bergman, 2010. "A New Landscape in Ontology Development Tools," in *AI3::Adaptive Information* blog, Sept. 7, 2010; see <http://www.mkbergman.com/909/a-new-landscape-in-ontology-development-tools/>.

PDF generated by *AI3::Adaptive Information* blog