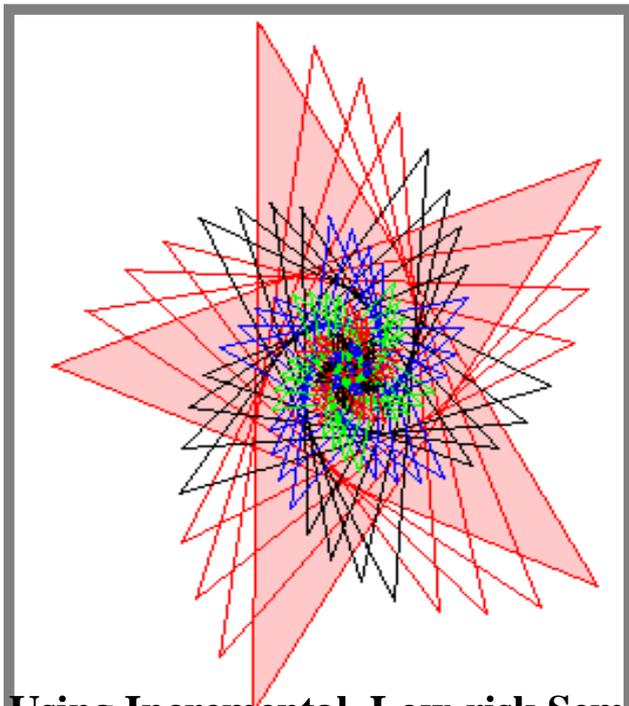


## 'Pay as You Benefit': A New Enterprise IT Strategy

by Mike Bergman - Monday, July 12, 2010

<http://www.mkbergman.com/896/pay-as-you-benefit-a-new-enterprise-it-strategy/>



### **Using Incremental, Low-risk Semantic and Open World Approaches**

OK. So, you're looking at your garage ... or your bedroom closet ... or your office and its files. They are a mess, and you can't find anything and you can't stuff anything more into the nooks, cubbies, crannies or cabinets. What do you do?

Well, when you finally get fed up and have a rainy day or some other excuse, you tackle the mess. Maybe you grab a big mug of coffee to prepare for the pending battle. Maybe you strip down to comfort clothes. Then, if you're like me, you begin to organize stuff into piles. Labeled piles and throwaway piles and any other piles that can provide a means to start bringing order to the chaos.

In the semantic Web world, there is a phrase coined by Jim Hendler that captures this approach: *A little semantics goes a long way* [1]. A little semantics, just like your labeled piles, helps to bring order to information chaos.

Mind you, this is not fancy or expensive stuff. In the case of my office, it is colored sheets of paper labeled with Magic Markers as "Taxes" or "Internal" or "Blog Posts" or whatever. Then, I begin sifting and distributing. In the case of the semantic world, these are classifying things into like categories and simply relating them to other categories with simple relationships, such as "is Part Of" or "is Narrower Than".

Of course, I could have approached my mess in a different way. I could have hired an efficiency expert to

come in, interview me and all of my employees and colleagues, gotten a written analysis and report, and then committed to a multi-week project to completely store and place every single last piece of paper in my office or organize every rake and set of abandoned golf clubs in my garage. When done, I would have shelled out much money and I suspect still not have been able to find anything.

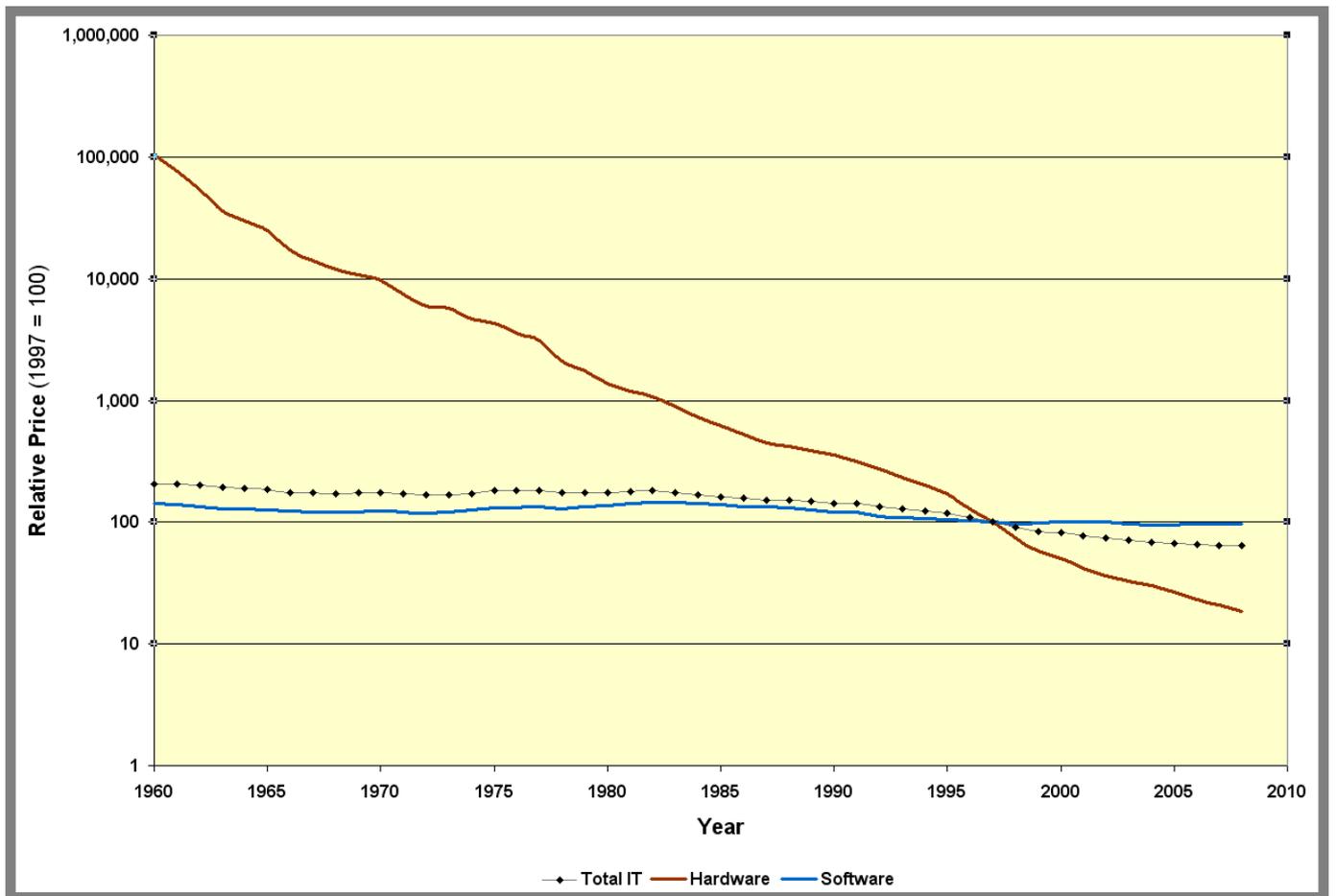
Sort of sounds like the traditional way IT does its business, doesn't it? To clean up their information messes, enterprises need to find a better strategy.

I'm not too long from having returned from the [SemTech](#) conference, which overall was quite an excellent show. But despite its emphasis on semantic technologies and their usefulness to businesses and enterprises, I found one critical theme unspoken: the ability of semantic approaches to change how enterprise IT actually *does* business. New ways have got to be found to clean up the many and growing information piles emerging all around us.

## **The Changing Nature of IT**

IT is -- and has been -- going through a fundamental set of changes for decades. In the last decade, these changes have led to lowered relative spending, a shift in spending priorities toward services, less innovation, and less productivity. Some data and observations by researchers and analysts document these trends.

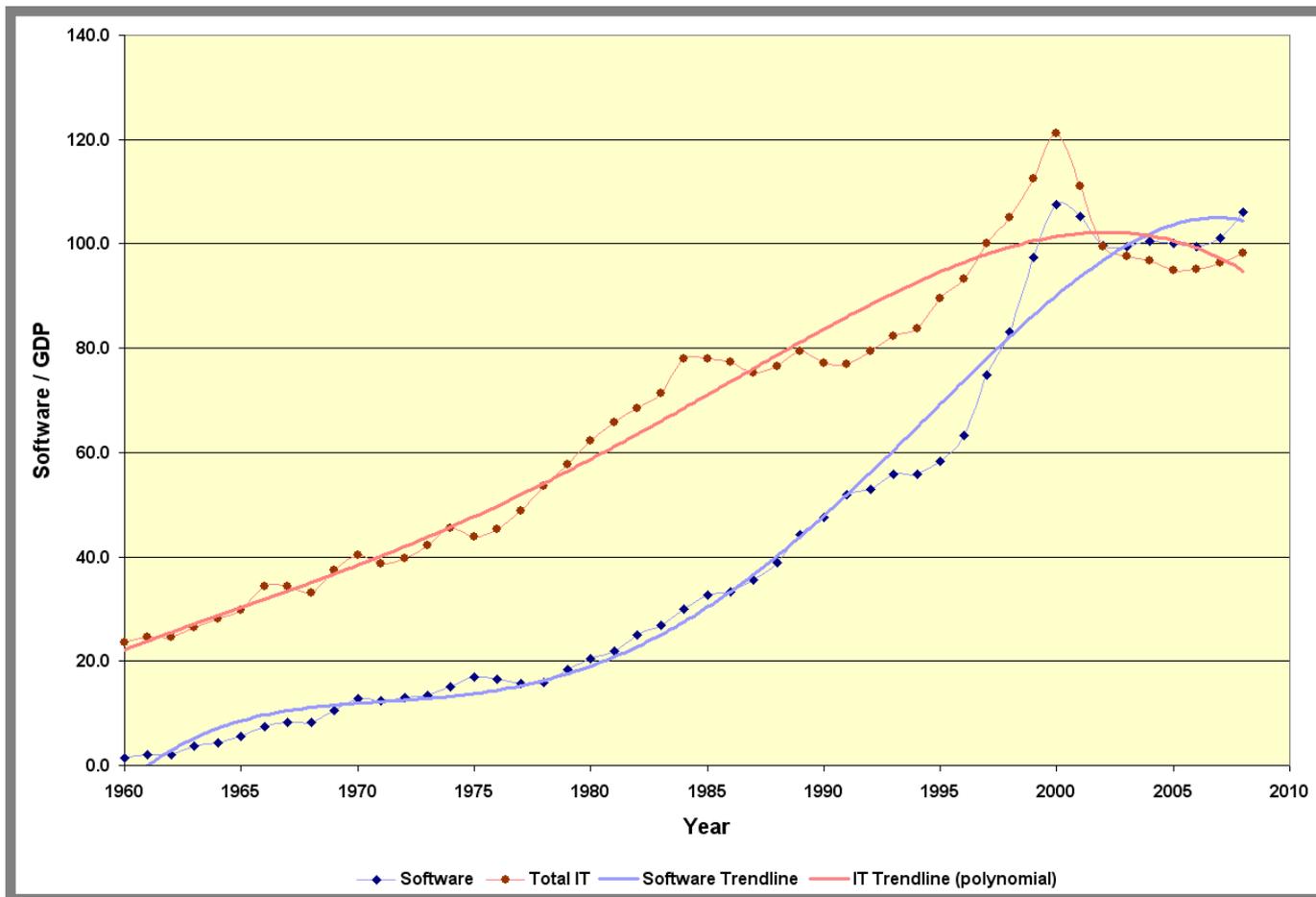
The following chart, using US Bureau of Economic Analysis data [\[2\]](#), shows the clear 50-year trend in declining hardware costs for enterprises, mostly resulting from the observation known as [Moore's Law](#). These massive hardware cost reductions (logarithmic scale) have also resulted in lower prices for IT as a whole. In 2008, for example, total relative IT prices were about two-thirds what they were a mere decade earlier:



Source: M.K. Bergman and Bureau of Economic Analysis [\[2\]](#) (click for [full size](#))

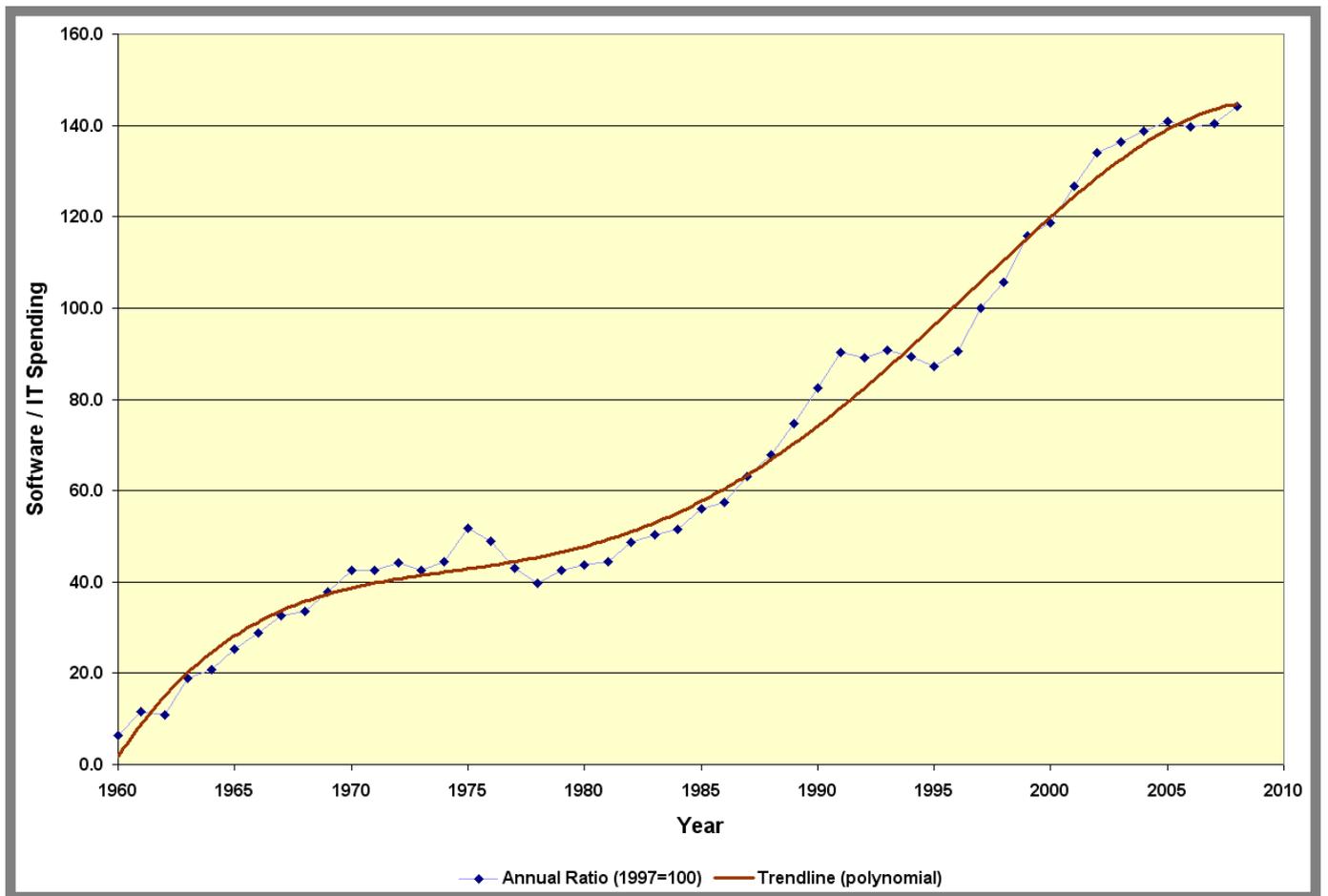
In contrast, relative prices for software and services have remained remarkably flat over this entire period, including for the past decade. This is somewhat surprising given the emergence of packaged software and more recently open source. However, relative percentage expenditures for custom software and software developed in-house have also remained strong over the past decade [\[3\]](#).

The mid- to late-1990s represented the high-water mark on many bases for enterprise IT, expenditures and vendors. Roughly in 1997 or so, the number of public enterprise software vendors peaked as did venture funding [\[4\]](#) and relative expenditures for IT in relation to GDP. There was a major uptick in relation to preparing for [Y2K](#) and a major downtick due to the [dot-com bubble](#), and then of course the past two years or so have seen a global economic downturn. But, as the figure below shows (red), the long-term trend tends to suggest a relative plateau for IT expenditures in relation to GDP somewhat around 2000:



Source: M.K. Bergman and Bureau of Economic Analysis [\[2\]](#) (click for [full size](#))

Yet, like the first chart, software seems to be bucking this trend (blue lines above). Though perhaps the rate of growth in expenditures for software is slowing a bit, it is still on a growth upslope, especially in relation to overall IT expenditures. The next chart, in fact, specifically compares software expenditures to total IT expenditures. Software expenditures are some 40% higher in relation to total IT than they were a mere decade ago:



Source: M.K. Bergman and Bureau of Economic Analysis [\[2\]](#) (click for [full size](#))

The mix of these software expenditures is also changing in major ways while stagnating in others.

The changing aspect is coming about from the shift of expenditures from license and maintenance fees to services. A number of software vendors began to see revenues from services overcome that from licensing in the 1990s. By the early 2000s, this was true for the enterprise software sector as a whole [4]. Today, service revenues account for 70% or so of aggregate sector revenues. Combined with the emergence of open source and other alternatives such as software as a service ([SaaS](#)), I think it fair to say that the era of proprietary software with exceedingly high margins from monopoly rents is over [5].

The stagnating aspect occurs in how the software expenditures are applied. According to Gartner, in the US, more than 70% of IT expenditures are devoted to simply running existing systems, with only about 11% of budgets devoted to innovation; other parts of the world spend nearly double on innovation and much lower for operations [6]. This relative lack of support for innovation and high percentages for running existing systems has held true for about a decade. Meanwhile, IT's contribution to US productivity has been declining since 2001 [7].

## What is the Cause for IT's Ills?

Last year, PricewaterhouseCoopers published a major report with the provocative title, "[Why Isn't IT Spending Creating More Value?](#)" [7]. The 42-page report covered many of the aspects above. Among other factors, the PWC authors speculated that:

*As consumption of IT increases and as technologies change and advance, businesses have been left to cobble together disparate software and hardware systems and tools. The end result? Unchecked IT spending, unneeded complexity, redundant systems, underutilized hardware and data centers, the need for expensive IT security, and, inevitably, diminishing returns from IT. In short, low levels of IT productivity create conditions for an IT cost crisis. [7]*

I suppose one could add to this litany other factors such as the growth and emergence of the Internet, sector consolidations through mergers and acquisitions, the rise of open source and alternatives such as SaaS, etc.

But which of these are causes? Which are symptoms? And which might only be consequences or coincident?

To be sure, all recognize the explosion of digital data and information, with sources and formats springing up faster than [Whack-a-Mole](#). It is such an evident and ubiquitous phenomenon that pointing to it as a cause appears on the face of it quite obvious. Also obvious is that these new sources carry with them a diversity of systems and tools. While not categorically stated as such, it appears that PWC fingers the difficulties of "cobbling" these systems together as the root cause for low productivity and thus the IT cost crisis.

I agree totally that these are symptoms of what we see in IT's current circumstance. I would even say these factors are a [proximate cause](#) to these ills. But I disagree they are the [root cause](#). To discover that root, I believe, we must look deeper to mindset and assumptions.

## Closed World Mindset as the Root Cause

There are some phenomena that are so obvious that they are easily missed. Not seeing your fingertip six inches between your eyes is one of these. We aren't used to focusing on things so near at hand.

So, let's look for a moment at the [closed world assumption](#) (CWA), a key underpinning to most standard relational data systems and enterprise schema and logics. CWA is the logic assumption that what is not currently known to be true, is false. If CWA is not directly familiar to you that is understandable; it is an implied assumption of these systems and logics. As such, it is not often inspected directly and therefore not often questioned [\[8\]](#).

With regard to standard IT systems, the closed world assumption has two important aspects:

1. The assumption is that the information domain at hand is *complete* [\[9\]](#), and
2. The related [negation as failure](#), which assumes every predicate to be false that cannot be proved to be true.

On the face of them, these assumptions seem tame enough. And, indeed, there are some enterprise data systems that absolutely rely on them for efficient processing and completion times, such as most transaction systems. CWA is absolutely the appropriate design for such applications.

However, for knowledge management or representation applications -- that is, applications which involve combining or using heterogeneous data or information from multiple data sources, which are *exactly* the same sources requiring information "cobbling" noted above by PWC -- there are two very critical implications of the closed-world assumption (CWA):

1. Efforts or projects can not be undertaken incrementally; if done in pieces, each piece must be complete and consistent, which is expensive to scope and do
2. To be consistent and explicit, the predicates (properties or relationships) must also be complex to model the "reality" of the system, which is also expensive to scope and do [\[10\]](#).

The net effect, which I have argued before, most notably in a major piece about the [open world assumption](#) [\[11\]](#), is that typical projects with a knowledge management aspect have become costly, take very long to complete, often fail, and require much planning and coordination. These facts have been true for three decades as enterprises have attempted to extract knowledge from their electronic information using closed world approaches based on relational systems. And, as recognized by PWC, these problems are only getting worse with growth in diversity and scope of systems.

The implications of closed world v. open world approaches are absolutely at the root of the causes leading to declining productivity, low innovation, significant failures and increasing costs -- all exacerbated with more data and more systems -- now characterizing traditional enterprise IT. Moreover, it is not a problem for open world systems to link to and incorporate closed world approaches. With open world, there is no

need for [Hobson's choices](#). Unfortunately, such is not true when one begins with a closed world premise.

## **Incremental is Good: Pay as You Go**

As best as I can tell, Alon Halevy was the first to use the phrase "[pay as you go](#)" in 2006 to describe the incremental aspect of the open world approach in relation to the semantic Web [12]. The "pay as you go" phrase had been applied earlier to data management and storage and had also been used to describe phone calling plans.

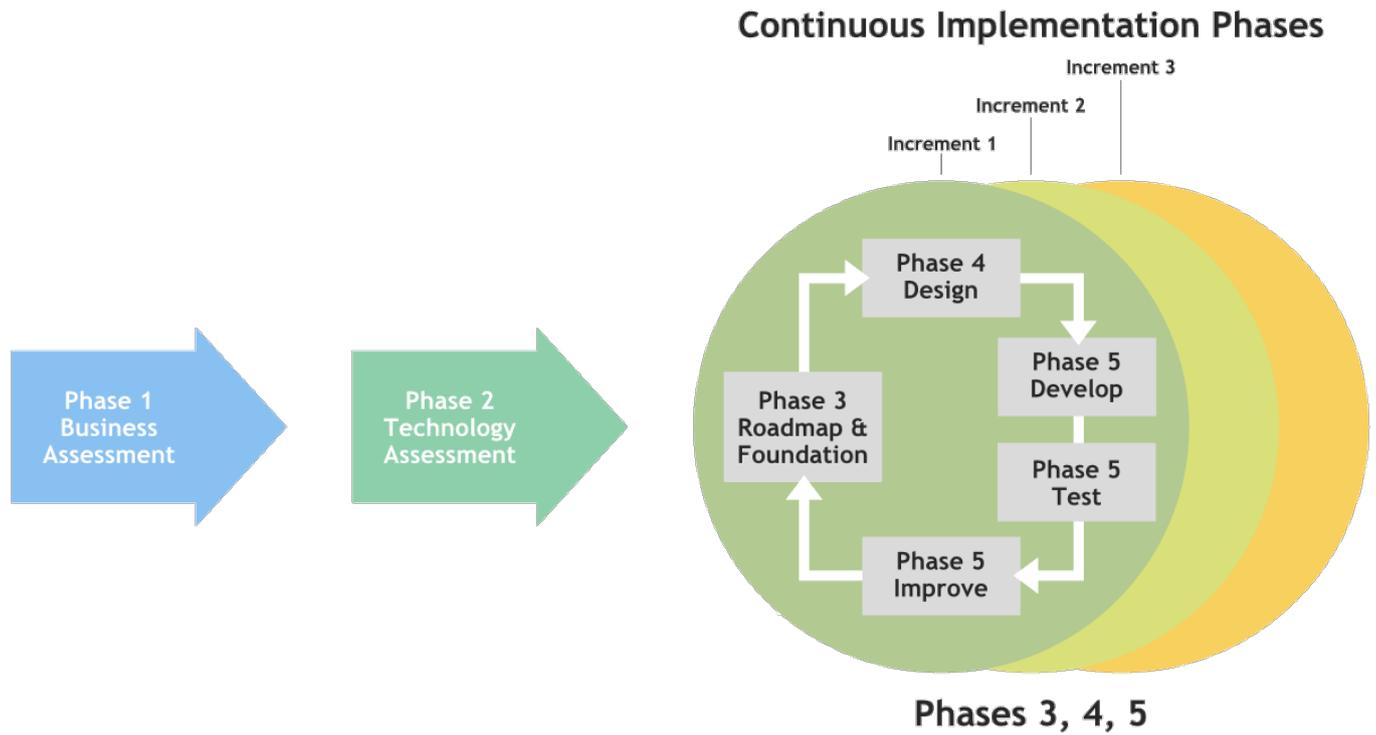
Incremental concepts and "agility" have been popular topics for the past five to ten years in IT, most often related to software development. And, while "incremental" sounds good in relation to enterprise projects, especially of a knowledge management or information integration/federation nature, the actual methodologies put forward were anything but incremental in their conceptual underpinnings.

Unfortunately, the "pay as you go" phrase has (and still is) largely confined to incremental, open world approaches involving the semantic Web. How this approach might apply and benefit enterprises has yet to be articulated. Nonetheless, I like the phrase, and I think it evokes the right mindset. In fact, I think with linked data and many other aspects of the current semantic Web we are seeing such approaches come to fruition. Inch-by-inch, brick-by-brick, data on the Web is getting exposed and interlinked. "Pay as you go" is incremental, and that is good.

## **Purposeful is Better: Pay as You Benefit**

Yet the idea of "pay as you benefit" is more purposeful, able to be planned and implemented, and founded on standard enterprise cost-benefit principles. I think it is a better (and more nuanced) expression of the "pay as you go" mindset in an enterprise setting. What it means is you can start small and be incomplete. You can target any domain or department or scope that is most useful and illustrative for your organization. You can deploy your first stand-ups as proofs-of-concept or sandboxes. And, you can build on each prior step with each subsequent one.

One of the reasons we ([Structured Dynamics](#)) embraced the [MIKE2.0](#) methodology [13] was its inherent incremental character. (Government deployments often call them "spirals".) In general, the five phases of MIKE2.0 can be represented as follows:



(click for [full size](#))

It is specifically during the fifth phase, testing and improvement, that quantitative and qualitative benefits from the current increment are calculated and documented. This evolving methodology is where the enterprise can assess the results of its prior investment and scope and budget for the next one. These can be quick, rapid increments, or more involved ones, depending on the schedule, prior results and risk

profile of the enterprise (or department) at that time.

Much is made of "incremental" or "agile" deployments within enterprises, but the nature of the traditional data system (and its closed world assumption) can act to undermine these laudable steps. The inherent nature of an open world approach, matched with methodologies and best practices, can work wonderfully with KM-related projects.

## **Quite Simply a Different Way to Do Business**

We see in our current IT circumstances a number of embedded practices and assumptions. We have been assuming control and completeness -- the closed world opposite to the open world approach. We have thus embraced and promoted "global" or enterprise-wide solutions: be they desktop operating systems or browsers or expensive enterprise-level proprietary software solutions. This scope leads to immense hurdle rates and risks: we better get our choices right up front, because if we don't, the department or enterprise are at risk. We have an inward focus about our own resources, our own networks, our own systems. Meanwhile, when we look outward, we wonder how all of these new Web companies can grow and expand so rapidly in comparison to us.

Clearly, we are seeing shifts to more services than products, more open source, more outsourcing, and more software as a service. Yet, because of the legacy of decades-long commitments from prior IT investment and the failures of many hyped "solutions" such as ERP or BI or data warehousing or a dozen others, we also see a decline and a reluctance for IT to embrace new and transforming approaches. Our prior choices were practically tantamount to "betting the enterprise." What if our new approaches fail as so many of their predecessors did? In a demanding, competitive environment can we afford to make such wrong choices again with such immense implications?

Yet, now that information technology is a given, it only seems natural that its role becomes an integral part of the enterprise, and not a special function. Like procurement, IT has matured to become a support function. Businesses should not succeed or fail based on the types of pencils and paper stock they use; so should they not depend on the software support choices that IT makes. Enterprises are now past the need to get "computerized"; they are thoroughly so. But our understanding of IT's role and position has not evolved with its own success.

The first whiffs of these challenges to IT's initial hegemony came from the departmental introduction of PCs and local networks in the early 1980s. It has continued with desktop software, spreadsheets and Web portals and sites. Large, mature companies awoke in horror in the last decade to discover they had hundreds -- sometimes thousands -- of Web sites and content dissemination points over which IT had little or no control. Such is the nature of entropy, and it is a fact for any organization of any size.

So, now, with strategies such as "pay as you benefit," there is no longer an excuse not to innovate. There is not a justification to put off testing and discovering benefits that the open world and semantic approaches can bring to your organization. There is now a basis to make the case and set the affordable budgets within desirable timelines for becoming a semantic enterprise.

Mindsets and expectations do require some adjustment. For example, not everything will be known or modeled in early phases. But, is that also not true in any "real" real world? We're not talking high-

throughput transaction systems here, but beginning to pull together and link the information that is important to your organization strategically.

Remember the intro statement that "a little semantics goes a long way"? Well, that truth -- and it is true -- when combined with incremental deployment firmly tied to demonstrable results, promises quite simply a different way to do business. Never before have enterprises had working and winnable approaches such as this to test and innovate and learn and discover. Jump on in; the water is clear and warm.

And, oh, as to that mess in your closet or garage? Well, if you adhere to CWA, you will need to define a place for everything to go before you can start cleaning things up. I say: forget those false hurdles. If you'd really want to make a dent in the mess, grab a broom and start cleaning.

---

[1] Jim Hendler, "a little semantics goes a long way." See <http://www.cs.rpi.edu/~hendler/LittleSemanticsWeb.html>.

[2] All starting data is for the United States *only* and comes from the U.S. Bureau of Economic Analysis, U.S. Department of Commerce. The data tables were downloaded from the BEA Web site at <http://www.bea.gov/national/nipaweb/SelectTable.asp>. GDP data is from Section 1; enterprise private investment data from Section 5. For reasons as described in the text, all relative BEA numbers were re-adjusted from a 2005 baseline to 1997 based on absolute figures. Software figures and expenditures include packaged software, custom software and software developed in-house, but excludes software bundled or included within hardware.

[3] Data not shown; see the "[Software Investment and Prices, by Type](http://www.bea.gov/national/info_comm_tech.htm)" data on the BEA Web page [http://www.bea.gov/national/info\\_comm\\_tech.htm](http://www.bea.gov/national/info_comm_tech.htm).

[4] Michael A. Cusumano, 2008. "The Changing Software Business: Moving from Products to Services," Massachusetts Institute of Technology, in *Computer*, Vol 41 (1): 20-27, January 2008. See <http://www.iae.univ-lille1.fr/SitesProjets/bmcommunity/Research/cusumano.pdf>. This shift has occurred despite the recognition that potential gross margins from software packages can exceed 90% due to zero costs of reproduction. As Cusumano notes in a rule, "99 percent of zero is zero: The great profit opportunity from software products becomes theoretical and not practical" if not sold. Also, another interesting observation made by Cusumano is that in the shift to services vendors with both low percentages and high percentages of services, or what he calls the "sweet spots", show higher contributions to profitability than vendors in the middle. He posits that low percentage vendors are getting mostly profitable maintenance fees, while those above 60% in services show profitability due to learning more replicable and systematic processes and approaches for service delivery.

[5] While we may occasionally see some vendors successfully buck this trend, I suspect these will only occur for established vendors with established platform advantages or for isolated applications where the innovating vendors have a significant first-mover advantage.

[6] Gartner calls the innovation category "transform"; see Gartner, Incorporated, 2009. "IT Software and Services, 2007-2010," see <http://www.slideshare.net/rsink/gartner-report-it-spending-2010>. Also, see Jed Rubin and Howard Rubin, 2006. "Worldwide IT Benchmark Service New Trends & Findings for 2007: Strategic Performance Management and Measurement," from *Gartner Consulting Worldwide IT Benchmark Service*; see [http://www.gartner.com/teleconferences/attributes/attr\\_161183\\_115.pdf](http://www.gartner.com/teleconferences/attributes/attr_161183_115.pdf).

[7] PricewaterhouseCoopers, 2009. "Why Isn't IT Spending Creating More Value?", see [http://www.pwc.com/en\\_US/us/increasing-it-effectiveness/assets/it\\_spending\\_creating\\_value.pdf](http://www.pwc.com/en_US/us/increasing-it-effectiveness/assets/it_spending_creating_value.pdf).

[8] Though relational database systems did not begin with an understanding of CWA, but rather [Edgar Codd's 12](#)

[rules](#), the understandings of these were formulated later by [Raymond Reiter](#). Reiter first described the basis of CWA in 1978, and then provided an axiomatization of relational databases and their deductive generalizations and basis in CWA in 1984; see [http://prism.cs.umd.edu/papers/Min02:reiter\\_memoriam/memoriam-tp1p.pdf](http://prism.cs.umd.edu/papers/Min02:reiter_memoriam/memoriam-tp1p.pdf).

[9] Relational database systems also assume unique names for objects, which, while not perhaps the best design for federated systems, can be overcome in other ways.

[10] For semantics-related projects there is a corollary problem to the use of CWA which is the need for upfront agreement on what all predicates "mean", which is difficult if not impossible in reality when different perspectives are the explicit purpose for the integration.

[11] See M. K. Bergman, 2009. [The Open World Assumption: Elephant in the Room](#), December 21, 2009. 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.

[12] This was also the first instance (I believe) of Alon coining the "dataspace" term. First use of the "pay as you go" phrase was, Alon Halevy, Michael Franklin, and David Maier, 2006. "Principles of Dataspace Systems," in *Proceedings of ACM Symposium on Principles of Database Systems*, pp: 1-9. See also the slides accompanying that talk, Alon Halevy, 2006. "Principles of Dataspace Systems (PODS)," June 26, 2006; see <http://www.cs.washington.edu/homes/alon/files/pods06-keynote.ppt>, 2006. More explicitly the next year see Jayant Madhavan, Shirley Cohen, Xin (Luna) Dong, Alon Y. Halevy, Shawn R. Jeffery, David Ko, and Cong Yu, 2007. "Web-scale Data Integration: You Can Afford to Pay as You Go." in *3rd Conf. on Innovative Data Systems Research (CIDR)*, pp 342-350, see <http://research.yahoo.com/files/paygo.pdf>. The term has been picked up by many others, notably Rada Chirkova, Dongfeng Chen, Fereidoon Sadriz and Timo J. Salo, 2007. "Pay-As-You-Go Information Integration: The Semantic Model Approach," see <ftp://ftp.csc.ncsu.edu/pub/tech/2007/TR-2007-30.pdf>; and most recently papers by Gerhard Weikum on RDF-3X; see <http://domino.mpi-inf.mpg.de/internet/reports.nsf/c125634c000710ceec125613300585c64/70e8f906d8090e6bc125757f00448ec9!OpenDocument&ExpandSection=-1>.

[13] See M.K. Bergman, 2010. ["MIKE2.0: Open Source Information Development in the Enterprise,"](#) *AI3 Blog* posting, February 23, 2010; and M.K. Bergman, 2010. ["Open SEAS: A Framework to Transition to a Semantic Enterprise,"](#) *AI3 Blog* posting, March 1, 2010.