Enterprise Resource Planning for Information Technology

Charles Betz

Why is managing Information Technology so hard? Consider this old chestnut.

A scientist gave a lecture on basic cosmology at a local library. Afterwards, an elderly woman came up and asserted, "You know, the world is really just sitting on the back of a gigantic turtle."

"But what is that turtle standing on?"

"Another turtle."

*"And what is **that** turtle standing on?"*

"You can't fool me, young man; it's turtles all the way down!"

The enterprise IT problem – it's a stack of turtles. A hall of mirrors. We are seeking data about the data and process to manage the processing. This article attempts a brief overview on these problems of IT management, and what it would take to more effectively automate them. A particular goal is demonstrating that the problem is so complex that no one vendor can cover it all. A common framework is required.

ERP for IT?

As head of a metadata management capability at a Fortune 100 company, one question I've found gets senior IT executives' attention is "Where's your ERP solution?"

It was therefore gratifying recently to read in several major industry periodicals that Ralph Szygenda, Chief Information Officer of General Motors, and his senior staff are challenging their vendors for "Enterprise Resource Planning for Information Technology." Now that such prominent members of the IT community have raised the call, it's time to look at some of the issues and make a few recommendations.

Enterprise resource planning software comprehensively manages the needs of a major enterprise resource area: money, productive capital, people, stock of goods, or information. Vendors such as Oracle, PeopleSoft and SAP build sophisticated, process-centric solutions, on complex information structures implemented in relational databases, for the business organizations that manage the enterprise resource.

Of the major resource areas, only information (i.e., IT) lacks such comprehensively integrated vendor solutions. Reasons for this include:

- The concept of information as a resource is relatively new. ERP systems in other areas are founded on decades- (if not centuries-) old business processes, such as dual-entry accounting.

- Infrastructure budgets in IT emphasize hardware and are often not directly tied to high-visibility, business-sponsored projects. As a result, the potential IT ERP software market is seen as limited.

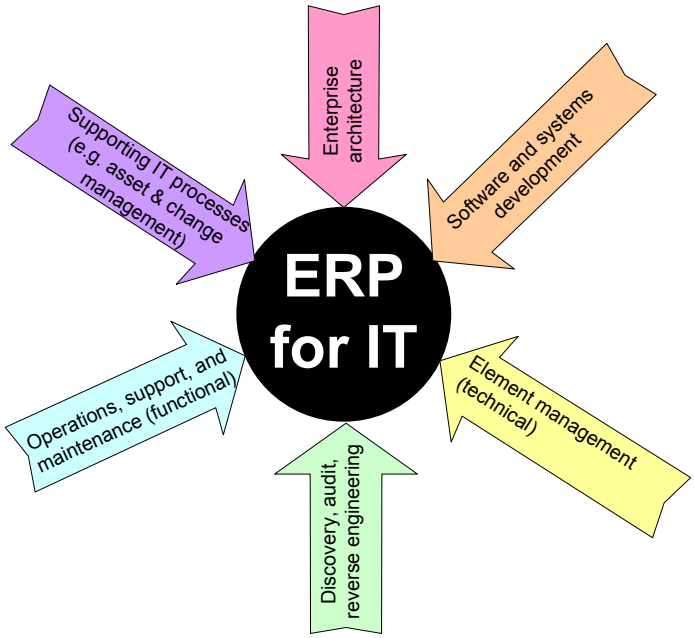
- The process discipline imposed by a true IT ERP solution would generate friction in most IT shops, especially if it involved short-term pain for high visibility, business-sponsored projects. Such projects have low tolerance for overhead imposed in the interest of longer-term IT efficiencies.

- Finally, there are formidable technical challenges, such as establishing workable information models for the problem domain.

Nevertheless, a convergence into ERP for IT is inevitable and necessary for such emerging areas as Business Activity Monitoring, Business Process Modeling/ Improvement, IT Service Management, increased outsourcing effectiveness, and other goals.

The IT problem domain

One representation of major process and data areas converging into the ERP for IT space is depicted in Figure 1. Rather than take the popular Zachman Framework as a point of departure, this model is much more narrowly focused on the modern IT organization as it is typically structured:



The ERP for IT grand convergence

From the top clockwise:

Enterprise Architecture includes high-level functional and process modeling, software portfolio management and program management, data management at the higher levels, and platform strategy (technologies, vendors, standards, and so forth). The IT finance capability also arguably belongs here.

Without a firm footing in enterprise architecture, the IT ERP effort will prove rudderless. The executives paying for IT at the highest level think in terms of the macro-level functions and processes defined in the enterprise architecture; these **must** be integrated into the IT ERP solution.

Software and systems development is what most people think of when they hear "IT," although operations and maintenance take the majority of the actual IT budget. This domain includes both custom built as well as package solutions, and as related to ERP for IT would cover all the tools used to deliver software, including project management packages.

Without integration between the software development lifecycle and the IT ERP capability:

- effective portfolio and program management will remain elusive

- development will continue in fragmented and nonstandard ways (hindering the rationalization of outsourcing)
- deployment and software inventory processes will continue to be haphazard
- traceability between software projects and their delivered runtime code will remain weak, perpetuating poor quality software

Technical element management is a general category for all the IT infrastructure areas requiring specialized tooling and skills, including:

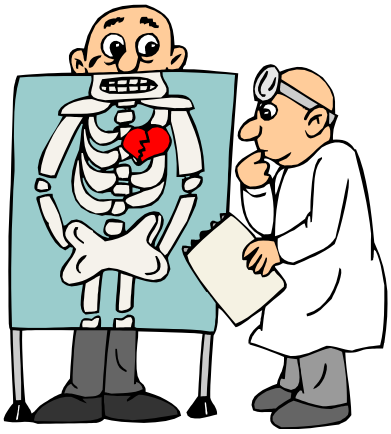
- Network and systems administration
- Database administration
- Messaging administration
- ETL administration

and others. Any one of these areas can contain a person's lifelong career path, and as capabilities they usually have strong, distinct teams – hence the differentiation between element management and general operations/help desk.

Today, they are often standalone silos with a craft mentality and insufficient integration with overall enterprise IT processes or systems. For example, a change ticket may tell a DBA to create a table, but the definition of that table is too often a manual, dual-entry process, with no automated routing of the technical specification (let alone execution managed by automated software release management).

Without alignment between IT element management and ERP for IT, these deep, essential areas will continue with limited visibility to senior executives. A craft mentality will persist, hindering improved automation. Finally, the business impact of critical IT incidents (which often first manifest in these areas) will remain obscured, and Business Activity Monitoring of IT operations will remain elusive.

IT Audit/Discovery. Many times, IT systems are built or altered without prior documentation. This is where scanning/discovery/reverse engineering comes in. This includes all the tools and techniques by which computerized systems can be understood *in situ*. Examples include data profiling, application mining, program understanding/reverse engineering, system fingerprinting, automated technology relationship mapping, and more.



The more advanced techniques and tools seek to tease out the emergent architectures and design patterns embodied in low-level artifacts – a significant theoretical challenge, and the subject of much research at organizations such as SEI.

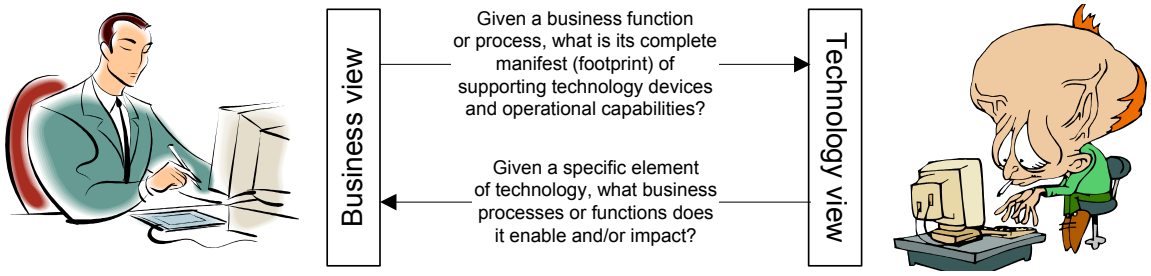
Unlike the others, this is not a well-established area in and of itself; its capabilities may be found spread across software development, element management, operational monitoring, and configuration management. However, it is a useful area to consider as distinct, and might emerge as an IT area of practice in its own right.

An ERP for IT function without an audit capability will have the credibility of a financial ERP package not backed by any audit – that is, none. Without actual inventory of what is in the data center and labs, process exceptions will not be identified, unused resources will remain allocated, rogue projects will be free to continue, and a true accounting of operational costs will remain elusive.

Operations, support, and maintenance are the primary "heads down" capabilities in the IT world itself, requiring call centers, 24 x 7 staffing, and the like. These more functionally-oriented capabilities are distinct from the deep, specialized technical element management teams that support them.

The software solutions here tend to be fairly mature, but silos– for example, it is not generally possible to correlate operational exceptions back to the software development process that created the defective software. Were inspections and testing fully carried out?

For another example, there is a gap between the ops/help desk capabilities and enterprise architecture concepts. An oft-stated vision in much recent product literature is the goal of **end-to-end traceability,** as illustrated below**.**



End to end traceability

Some operations framework vendors are starting to attempt to support high-level semantics in an attempt to bring BAM-type capabilities to operational monitoring. However, this begs the question – how are the processes to be described? A rich process modeling language is required, and it is doubtful that operations- and help desk-focused vendors will be able to accommodate the sophisticated conceptions of modern enterprise architecture software such as Popkin or Ptech.

The enterprise architects and analysts who are responsible for business process modeling will demand compatibility with their preferred tools rather than manually re-enter their models in process modeling bolt-ons to operations frameworks. Integration based on industry standards is required.

In sum, without a direct link to ops/help desk, an ERP for IT package will be unable to correlate key cost information on operational activities back to the business processes they support, and the upstream design/build activities which initiated them. Such traceability is essential to managing IT costs rationally, prioritizing support and maintenance activities, driving software quality, and managing IT operational risk.

Supporting IT Process includes those executed by smaller IT workgroups outside of the data center and help desk. These various processes help integrate the software development lifecycle into the enterprise (i.e. change management), as well as providing configuration management, asset management, systems deployment, capacity planning, and a host of other services.

These process areas are usually underserved in terms of IT automation; often they are run using Excel spreadsheets or at best Access databases. The complexity of Global 2000 IT environments and the desire for their greater efficiency is forcing a painful rationalization of such practices, and one might argue this major area is the primary driver for ERP for IT.

The industry is moving towards a consensus that this general area (plus operations/help desk) should be called IT Service Management, and the pre-eminent standard in this area is the UK's Information Technology Infrastructure Library (ITIL). The major trade association is the IT Service Management Forum.

The ITIL material focuses on "what," not "how," and tends to simply call for the existence of a best practice. ITIL badly needs reference information and process models; in fact, earlier incarnations of ITIL had more of this sort of analysis than the current iteration. This is not necessarily a bad thing, as other organizations and consortia (such as the OMG and the Distributed Management Task Force) engage in complementary work. Below is a matrix demonstrating the possible relationship between the OMG standards and well-known IT process models. (A first cut, the Xs are debatable.) A similar analysis could be done for the lower-level standard models developed by the Distributed Management Task Force. The point is that IT has rich and robust standards for both process and data, yet little work has been done to integrate them.

Object Management Group standards		CMM				ITIL							
		Process management	Project management	Engineering	Support	Data management	Configuration management	Release management	Change management	Incident management	Problem management	Availability management	Capacity management
	Software Process Engineering Metamodel	x	x	x	x			x					
	Unified Modeling Language			x									
	Component view			x			x	x	x				
	Deployment view			x			x	x	x	x	x	x	x
	UML Profile for Enterprise Application Integration			x									
	Enterprise Distributed Object Computing			x									
	Organization Structure Facility		x			x				x	x		
	Common Warehouse Metamodel			x		x							
	Entity/relationship			x		x							
	Relational					x			x				
	Software deployment			x			x	x	x	x	x	x	x
	Warehouse operations				x					x	x	x	x

Process-centric versus information-centric standards

Without the common rationalization and centralization of these processes in an IT ERP solution, they will continue to suffer from inconsistent interpretation and application, poor data integrity due to multiple masters and unclear maintenance protocols, uncontrolled, hairball interfaces, and a host of other common problems that IT organizations are well familiar with **in their clients' systems.** IT is its own worst customer, a barefoot cobbler's child.

Let's turn to some background and deeper perspectives.

Computer Assisted Software Engineering: the previous generation

There is nothing new under the sun. The history of computer assisted software engineering stands out as a stark lesson for the IT ERP project. We all know CASE failed. Or did it? Many mission-critical, mainframe-based systems were built and are maintained on first-generation CASE tools to this day. Failure? Let's say rather that

- CASE was oversold,
- it didn't scale down or out to smaller, more heterogeneous distributed architectures very well,

- its tooling suffered from monolithic proprietary architectures,

- and in the end it was consolidated and milked for licensing revenue; investment stagnated.

But much of the IT ERP effort will reflect the problems of first generation CASE; the leaders of the IT ERP project should review this history. Some of the industry's brightest minds put much high quality (and still relevant) thought into this, including various efforts to standardize necessary semantics: CDIF, IRDS, PCTE, and others. These efforts converged in the work of the Object Management Group, with its layered modeling paradigm.

There is no meaningful competition to the OMG's work, which is capable of absorbing and representing virtually any modeling language imaginable (for example, the Common Warehouse Metamodel supports entity/relationship modeling). The OMG is not just about object modeling any more; they have defined the equivalent of TCP/IP for modeling languages and metadata.

Such standards unification will enable the creation of an ecosystem of ERP IT vendors specializing in various areas, assured of their solutions' interoperability through protocols such as XML Metadata Interchange. Without such unification, IT ERP will repeat the mistakes of monolithic, single-vendor ERP solutions.

If the "IT Doesn't Matter" thesis of Nicholas Carr is even partially correct, it supports the position that **both vendors and consumers of IT should increase their support of standards bodies,** so that fewer resources are spent re-inventing commodity wheels.

Metadata

The ERP for IT project will live and die by the quality of its normalized data structures, a situation also encountered by first generation CASE. In fact, the most challenging aspect of supporting first generation, repository-based CASE tools arguably was understanding the information model.

If framed as a class or entity/relationship model (more precisely, a **metamodel**), the major IT entities might be:

- System
- Component
- Feed (or Flow)
- Process
- Event
- Interface
- Datastore
- Data Element
- Project
- Document
- Artifact/Deliverable
- Party (individual or team)
- Device
- Incident

and many others.

It's a tough problem area to model. The subtyping hierarchies are deep, recursion is rampant, and many-many relationships are numerous – on the whole, a nontrivial modeling problem, compounded by the need to support abstractions (e.g. logical/physical).

None of these requirements are well supported by mainstream relational database technology, and leading metadata platforms are therefore based on an object-oriented layer which supports robust inheritance, OO associations, and graph queries.

In fact, a new trend in metadata is the highly graph-centric approach, emerging as perhaps an overly extreme response to the difficulties of metamodeling the IT problem domain. For example, the ITIL concept of a "configuration management database" uses "configuration item" (CI) as a catchall general type for any item of interest to IT's internal processes. However, ITIL does not seem to call for a robust relationship model (e.g. subtyping, cardinality and other constraints) with which to describe and enforce the valid relationships between CIs. Gartner Group has also identified a new industry sector called Technology Relationship Mapping; represented by vendors such as Troux, the emphasis again is primarily on the graph and much less on the valid semantics of the various node connections.

Less radically, the Object Management Group's family of modeling languages, descended from classic entity-relationship modeling, is the leading candidate for industry standards. Most of the core IT concepts listed above, well partitioned and with detailed, precise subtyping and constraint semantics bearing the mark of much high-quality thought and debate, can be found in current approved or pending OMG standards.

However, the OMG's work has some key gaps for an IT ERP solution. Modeling IT's financials in particular appears to be poorly addressed, and the standard of greatest relevance to enterprise IT--the Software Portfolio Management Facility--has languished in the OMG's approval process, overshadowed by the current UML revisions.

There are also unreconciled competing efforts (e.g. the Distributed Management Task Force and the Business Process Modeling Language effort, as well as the more academic work around ontologies and the Semantic Web). The IT ERP project needs to identify and back the standards players with the greatest commitment to interoperability, a commitment noticeably lacking in the BPML leadership for example.

Logical versus physical: the abstraction problem

The issue of metadata leads to the logical/physical (or "what versus how") distinction, identified years ago and well detailed in (for example) the Zachman Framework. This is **not** a discussion of data modeling, although data is a good place to start in understanding the issues of abstraction. Enterprise application integration and business process also require logical/physical mapping. For example, a high level integration diagram might show systems as key abstractions with the interfaces between them as simple lines. A physical decomposition of this would show the actual components, servers, and queues implementing the source and target systems and the data flow in between.

Tracing from the abstraction of what a complex system does, to the reality of how it is physically built, is a problem no other ERP domain faces in quite the same way. Both physical and logical metadata suffer from distinct challenges in and of themselves, and the additional, critical task of mapping between them is expensive and difficult. (The OMG's Model Driven Architecture attempts to address precisely this challenge.)

Conceptually, physical metadata is relatively straightforward to understand– there is little dispute about what things **are**. It also is amenable to automated discovery and correlation processes, and the tooling in this area becomes more sophisticated every year.

Logical representation, on the other hand, requires 1) the creation of sophisticated consensus among a community of users about what the key abstractions **mean** and then 2) the use of those structures in expensive collaborative analysis to actually build out a logical conception of the enterprise systems. The first consensus has proved elusive unless the logical *metamodel* – the concepts or language used to describe the logical system – is very carefully crafted. If too complex, people become bewildered and tune out. If too simple, people have to interpret the concepts and mis-alignment easily emerges.

Even when the logical metamodel is well-established, **maintaining** the metadata in it (and keeping it traced to lower and higher levels) has historically proven costly and challenging, resulting in an unfortunate yo-yo commitment to disciplines like enterprise architecture. ("Too expensive! Get rid of it! We don't know what we have! Start it up again!" And so on.)

Why is logical/physical important? One reason is that in the decision support sense, the physical **rolls up** into the logical. Executives rely on rollups as abstractions, and unfortunately in the IT problem domain the dimensions are

many and nowhere near as well established as such classical data warehousing hierarchies as item, time, and location.

Integration metadata

One area not well addressed by any standards or notations to date is precisely documenting integrations among large, heterogeneous, distributed systems. This is a requirement for any solution purporting to provide an ERP for IT capability, one fraught with challenges:

The heterogeneity of the EAI world. As well documented in *BI Journal* and by authors like David Linthicum, the integration problem encompasses a bewildering variety of technologies: messaging, FTP, database middleware, application servers, message brokers, and more.

The "hairball" nature of the metadata. A data dictionary is amenable to standard reporting techniques, to answer well defined questions such as:

- What columns are on this table?
- What tables are in this schema?
- What schemas are in this database?

And so forth. Integration metadata, by contrast, is concerned with end to end semantics:

- Given system A and system B, show how data moves between them, the components involved, and the business processes supported.

This question is an order of magnitude harder than the data questions above. In the data world, we know that schemas contain tables contain columns. With integration flows, we have no such certainty. Generally, we can know neither the number of "hops," nor their type, between two endpoints **prior** to the execution of a query. We also may want to artificially limit the scope of the hairball pulled back into the report; simply asking for "everything connected between two systems" runs into the basic problem that in a hairball (formally speaking, a graph), everything may be connected. Such "bill of materials" or "network" queries are difficult at best in the relational database world, driving the practitioner to object oriented solutions such as OMG MOF-based repositories, or graph-centric TRM solutions such as Troux.

Logical/physical. The logical/physical problem is especially acute with integration as well, because both the logical and physical integration worlds require graph structures to manage their data (they are both hairballs), and tracing high-level information flows to their physical implementations is a significant challenge. The problem is not just one hairball, but at least two, and each and every strand of hair in one ball must be tied to its counterpart(s) in another.

Presentation problems. IT departments traditionally create large "wallpaper" diagrams documenting systems architectures. As the late Dr. Bernard Boar lamented, such diagrams usually follow no blueprinting standard, but do embed common understandings and serve as important references. These diagrams are typically **not** stored in repositories, although that would be ideal from a management and accessibility point of view. Ideally, such diagrams should be simply specialized views of one integrated model.

Repository-based data modeling tools have such capabilities. **However, a key enabler for data-centric CASE tooling was the emergence of entity/relationship diagramming as a consensus language for describing data.** A similar common language is required for the integration space and the systems engineering space more generally.

Possible solutions. Much research has gone into software architecture description in recent years, sponsored by organizations like SEI. "Architecture Description Languages" (ADLs) have been a popular avenue of investigation.

The Object Management Group has also made several attempts to support this area, but the ADL community has been skeptical of UML for architecture description. The forthcoming UML 2 standard reflects some of these

critiques, and there has been other relevant OMG work (EDOC, CWM, EAI/UML, and others). Some of these standards however are highly complex, and there are usability questions.

Future directions

If the IT ERP project is to truly encompass its potential, it must start by building on the ongoing attempts to standardize CASE. In fact, CASE might be redefined from Computer Assisted **Software** Engineering to Computer Assisted **Systems** Engineering, to better reflect the configuration management and operations components. The OMG standards clearly represent the culmination of attempts to standardize CASE and are the leading option for a foundation that could bring everything together.

The alternatives are not satisfying. A major IT vendor might declare the first IT ERP solution, attempting to become the PeopleSoft or SAP of IT. (There are smaller vendors already claiming this, but they do not begin to approach the scope outlined in this article.) Complete coverage would be a formidable challenge for a single vendor, if that vendor sought to avoid any use of industry standards and instead impose a proprietary approach. However, it is clear that the big players are starting to think about their strategies in this area, HP with recent extensions to its OpenView suite, IBM with its OMG-based eMOF foundation.

Another scenario might be that GM starts to drive the whole project, much as Wal-Mart has started to impose its supply chain protocols on an entire industry.

Clearly, a standards-based ecosystem allowing rich specialization and niche players all based on the common OMG semantic bus would be the ideal.

As the self-reflective imagery at the start of this article attempted to communicate, enterprise information technology, if not the hand that steers the rudder, is at least the hinge upon which the rudder pivots. It is highly leveraged, and any improvement in its management should have a multiplier effect on the enterprise's effectiveness. An industrywide improvement could have a similar effect on society's general prosperity.

Onward to IT ERP!

Charles Betz heads the Metadata Management Office for electronics retailer Best Buy. He previously worked for Target Corporation (where he built that company's metadata repository), and also for Andersen Consulting as technical architect for ERP solutions. He received his Master of Science in Software Engineering from the University of Minnesota in May 2003; his research topic was "The Convergence of Metadata and IT Service Management."

He is an active member of the professional community, belonging to the IEEE, ACM, and DAMA (Data Management Association). He has presented several times to national and local DAMA sessions, on "Object Orientation meets the Relational Database" and "The OMG's Metadata Standard," and was invited by the EAI Industry Consortium in June to present "IT Doesn't Matter. Metadata Does."

Acknowledgements go to John Schmidt, Pete Rivett, Sean Goggins, and Peggy Dora for their careful feedback on this article. The opinions expressed here are the author's alone.