Enterprise Resource Planning for IT

By Charles Betz

Why is managing IT so hard? Consider this old chestnut: A scientist gave a lecture on basic cosmology at a local library. Afterward, 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 is a stack of turtles. We're seeking data about the data and process to manage the processing. This article attempts a brief overview on these IT management problems and what's needed to more effectively automate them. The problem is so complex that no one vendor can cover it all; a common framework is required.

ERP for IT

Want to get senior IT executives' attention? Ask them, "Where's your ERP solution?"

Ralph Szygenda, CIO of General Motors, and his senior staff are challenging their vendors for ERP for IT. 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.

ERP 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-old business processes such as dual-entry accounting.
- Infrastructure budgets in IT emphasize hardware and often aren't directly tied to high-visibility, business-sponsored projects. So 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.

IT Problem Domain

Nevertheless, a convergence into ERP for IT is inevitable and necessary for such emerging areas as business activity monitoring (BAM), business process modeling (BPM), IT service management, increased outsourcing effectiveness, and other goals. Figure 1 depicts one representation of major process and data areas converging into the ERP for IT space. This model is narrowly focused on the modern IT organization as it's typically structured. From the top of Figure 1 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 (e.g., technologies, vendors, standards). 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 most of the actual IT budget. This domain includes both custom-built and 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 life cycle and the IT ERP capability:

business integration journal takeaways

BUSINESS

- Despite its success building solutions for capabilities such as finance, supply chain, and human resources, IT's own internal systems and processes often are fragmented, inefficient, and redundant.
- Desire for more effective outsourcing in particular is driving the call for "Enterprise Resource Planning for Information Technology."
- However, no vendor currently offers a comprehensively integrated ERP suite for IT, and this provides an opening for standards-based approaches.

TECHNOLOGY

- IT itself is probably the most challenging business area in which to apply information and process modelling techniques.
- However, this complexity is mitigated by the fact that large-scale IT has similarities across all industries, and is ripe for standardization.
- The Object Management Group should step up efforts toward completing the Software Portfolio Management Facility in particular, a draft specification with great relevance to the ERP for IT problem.

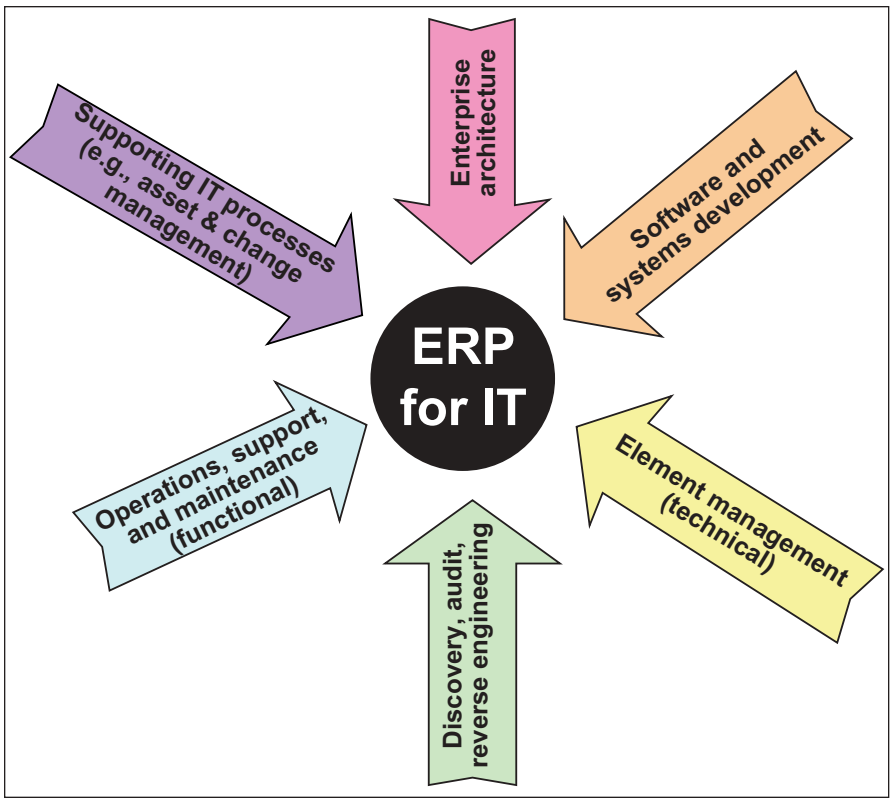

Figure 1—The ERP For IT Grand Convergence

- Effective portfolio and program management will remain elusive.
- Development will continue in fragmented and non-standard ways (hindering the rationalization of outsourcing).
- Deployment and software inventory processes will continue to be haphazard.
- Traceability between software projects and their delivered run-time 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
- Extraction, transformation, and loading (ETL) administration.

Any one of these areas and others 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're often stand-alone 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 (BAM) 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. 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. This is a significant theoretical challenge and the subject of much research.

Unlike the others, this isn't 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's 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. Without actual inventory of what's in the data center and labs, process exceptions won't 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, continuous 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 isn't 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's a gap between the operations/help desk capabilities and enterprise architecture concepts. An oft-stated vision in much recent product literature is the goal of end-to-end traceability, which is shown in Figure 2.

Some operations solution vendors are starting to support high-level semantics to bring BAM-type capabilities to operational monitoring. However, this begs a question. How are the processes to be described? A rich process modeling language is required and it's doubtful that operations- and help desk-focused vendors can accommodate sophisticated modern enterprise architecture software such as Popkin or Ptech.

The enterprise architects and analysts responsible for BPM 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.

Without a direct link to operations or

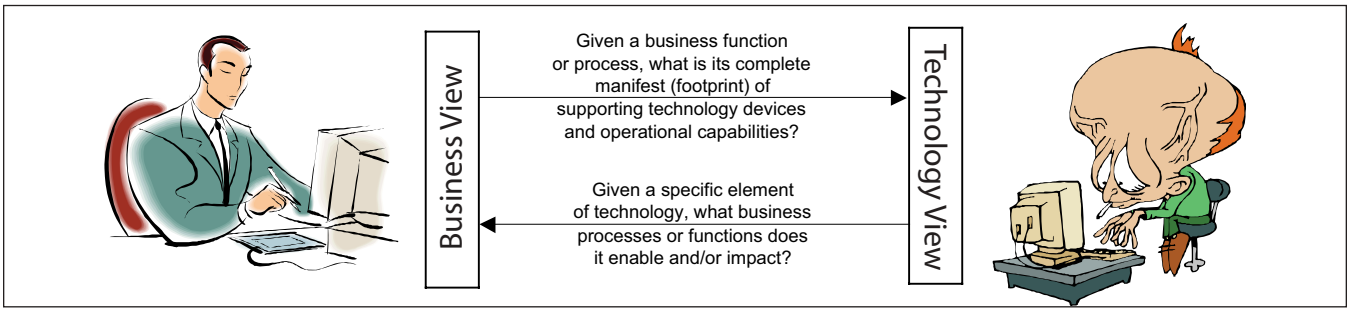

Figure 2—End-to-End Traceability

help desks, 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 that 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 the data center and help desk. These various processes help integrate the software development life cycle into the enterprise (i.e., change management), as well as providing configuration management, asset management, systems deployment, capacity planning, and other services.

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

The industry is moving toward a consensus that this general area should be called IT service management, and the pre-eminent standard in this area is the U.K.'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; earlier incarnations of ITIL had more of this sort of analysis than the current iteration. This isn't necessarily bad, as other organizations and consortia—such as the Object Management Group (OMG) and the Distributed Management Task Force—engage in complementary work. Figure 3 is a matrix demonstrating the possible relationship between the OMG standards and well-known IT process models. A similar analysis could be done for lower-level standard models developed by the Distributed Management Task Force. The point is that IT has rich, robust standards for both process and data, yet little work has been done to integrate them.

Without the common rationalization and centralization of these processes in an IT ERP solution, they'll 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. IT is its own worst customer—a barefoot cobbler's child.

Let's turn to some background and deeper perspectives.

CASE: The Previous Generation

There's nothing new under the sun. The history of computer-assisted software engineering (CASE) stands out as a stark lesson for the IT ERP project. We all know CASE failed. Or did it? Many strategic, 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 well to smaller, more heterogeneous distributed architectures.
- Its tooling suffered from monolithic proprietary architectures.
- Ultimately, 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. These efforts converged in the work of the OMG, with its layered modeling paradigm.

There's no meaningful competition to the OMG's work, which can absorb and represent virtually any modeling language imaginable, including the Common Warehouse Metamodel, which supports entity/relationship modeling. The OMG isn't just about object modeling any more; they've defined the equivalent of TCP/IP for modeling languages and metadata.

Such standards unification will enable 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 fewer resources are spent re-inventing commodities.

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. Understanding the information model was the most challenging aspect of supporting first-generation, repository-based CASE tools.

If framed as a class or entity/relationship model (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.

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

None of these requirements are well-supported by mainstream relational database technology. Leading metadata platforms are therefore based on an object-oriented (OO) layer that 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 doesn't 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 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 OMG'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 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. 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 Unified Modeling Language (UML) revisions. Moreover, there are competing efforts (e.g., the Distributed Management Task Force, the Business Process Modeling Language [BPML] effort, and more academic work around ontologies and the wemantic 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.

The Abstraction Problem

The issue of metadata leads to the logical/physical (or "what vs. how") distinction, identified years ago. This isn't a discussion of data modeling, although data is a good place to start in understanding the issues of abstraction. EAI and business processes 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's 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. 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's little dispute about what things are. It also is amenable to automated discovery and correlation processes, and the tooling in this area becomes more

Object Management Group Standards	CMM				ITIL					
	Process Management	Project Management	Engineering	Support	Data Management	Configuration Management	Release Management	Change Management	Incident Management	Problem 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
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
Warehouse operations				x					x	x

Figure 3—Process-Centric vs. Informatioin-Centric Standards

sophisticated every year.

Logical representation, on the other hand, requires:

- The creation of sophisticated consensus among a community of users about what the key abstractions mean
- 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 carefully crafted. If too complex, people become bewildered and tune out. If too simple, people have to interpret the concepts and misalignment 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 such as enterprise architecture. Logical/physical is important because, in decision support sense, the physical rolls up into the logical. Executives rely on rollups as abstractions. In the IT problem domain, the dimensions are many and nowhere near as well-established as classical data warehousing hierarchies such as item, time, and location.

Integration Metadata

One area not well-addressed by any standards or notations is precisely documenting integration among large, heterogeneous, distributed systems. This is a requirement for any solution purporting to provide an ERP for IT capability. Challenges include:

The heterogeneity of the EAI world: The integration problem encompasses a bewildering variety of technologies: messaging, FTP, database middleware, application servers, message brokers, and more.

The "hairball" nature of 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?

Integration metadata, by contrast, is concerned with end-to-end semantics. It explores how, given System A and System B, data moves between them and business processes are supported.

This is much harder than the data questions above. In the data world, we know that schemas contain tables that 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 *before* query execution. We also may want to artificially limit the scope of the hairball pulled back into the report because everything may be connected.

Logical/physical: The logical/physical problem is especially acute with integration because both the logical and physical integration worlds require graph structures to manage their data (they're both hairballs) and tracing high-level information flows to their physical implementations is a significant challenge. The problem isn't 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 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 typically *aren't* stored in repositories, although that would be ideal for management and accessibility. 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 integration and systems engineering.

Possible solutions: Architecture Description Languages (ADLs) have been a popular avenue of investigation. The OMG 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's been other relevant OMG work. 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 as 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 aren't 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 don't begin to approach the scope outlined here. Complete coverage would be a formidable challenge for a single vendor if that vendor sought to avoid any use of industry standards and, instead, imposed a proprietary approach. However, it's clear that the big players are starting to think about their strategies in this area. For example, HP has extended its OpenView suite; IBM has offered 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 ideal.

Enterprise information technology, if not the hand that steers the rudder, is at least the hinge upon which the rudder pivots. It's 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, positive effect. Onward to IT ERP! BIJ

The author acknowledges John Schmidt, Pete Rivett, Sean Goggins, and Peggy Dora for their feedback on this article.

About the Author

Charles Betz heads the Metadata Management Office for electronics retailer Best Buy. He previously worked for Target Corp. (where he built that company's metadata repository), and also for Andersen Consulting as technical architect for ERP solutions. Voice: 612-251-9888; e-Mail: charb@visi.com; Website: www.erp4it.com.