Editor: Charles Petrie • petrie@stanford.edu

ERP is Dead, Long Live ERP

Paul Hofmann • *SAP Labs, Palo Alto*

Some mature information technologies, such as enterprise resource-planning (ERP) systems and relational databases (RDBs), are now undergoing commoditization, much like what happened in the automotive and chemical industries over the past 15 years. This trend is accelerated by reduced IT spending because of slowing economic growth.

In such an environment, market leaders in computing, networking, and telecommunications need to increase investment in disruptive markets and business models. Growing in a down market will require innovation; leaders must pick the right breakthrough technologies from current trends, including high-performance computing, pervasive connectivity, Web services, and service-oriented architecture (SOA).

Software makers face many business challenges in this situation, aggravated by software-as-a-service (SaaS) players moving into front-end enterprise applications. Yet, various technologies do have the potential to fuel innovations as the future of ERP applications unfolds.

Current Situation

Information technologies and systems typically mature and undergo commoditization as a constant stream of new technologies and applications with disruptive potential emerges. Some 15 years ago, for example, PCs and high-performance Windows NT and Unix servers replaced IBM's water-cooled computers. The first business applications for accounting and financials ran on mainframe host computers, but standardized client-server ERP products with company-specific adaptations replaced these tailor-made legacy host applications. The typical ERP system runs databases and applications on back-end servers while the user interface runs on PCs or laptops.

I see big technology and business model changes coming for ERP. New technology trends such as high-performance computing, pervasive connectivity, Web services, and SOA will affect the front end as well as the back end of ERP. For convenience, I use ERP as synonymous for back-end business apps because ERP is the main, and most important, back-end application. Indeed, most analysts consider new apps, such as supply-chain management (SCM) to be an extension of ERP.

Reliable Back-End Business Apps

ERP systems are the well-structured, reliable IT backbones that run the financials, human resources, and logistics of Fortune 500 companies worldwide. ERPs are large and complex — SAP ships more than 300 million lines of code in its business suite, for example — but they use proven technologies to reliably execute well-defined process flows in the form of very secure transactions. The resulting data are stored in RDBs and can be automatically monitored and audited to check for standards compliance, inventory levels and values, and so on. ERPs can scale to automate the millions of transactions in the retail banking business. The technology is mature and changes at a slow rate. Indeed, an ERP's average lifecycle is about 15 years.

User-Centric Front-End Apps

In contrast to the structured back-end ERP, enterprises also employ highly agile and adaptive front-end software for customer-relation management (CRM), mobile maintenance, call-center applications, email, project planning, and so on. This user-centric software, which often runs on mobile devices, uses lightweight technologies

 1089-7801/08/$25.00

such as Ajax, scripting languages, the Resource Description Framework (RDF), and representational state transfer (REST). These applications are collaborative, interactive, and exception-driven.

Rather than predefined business processes that must be audited later, they feature ad hoc, informal processes that are highly variable. These front-end applications have very short life cycles — like throwaway software — and consistency isn't the foremost goal. Consider the Google window of the Web: rather than a time-wise consistent picture, search results show snapshots of the Net from different times. Search results with content from the Super Bowl and CNN are always up to date, for example, but papers at .edu domains generally take a while to be included in the index, and updates at some science sites take about two months to appear in Google searches.

Enterprises clearly couldn't accept such inconsistency for accounting or human resources. In fact, one of the big contributions that ERP systems have made to corporate productivity is a globally consistent view of a company's financials and inventory at any time.

Front-end applications' limited reliability can lead to the loss of data and transactions, but that isn't generally a big deal for Web applications. If Amazon.com loses an order, for example, the user can simply reenter it. Most of us didn't even realize that Amazon was down for several hours on 15 February 2008.

Relevant Technology Megatrends

The flagship of the new generation of front-end software, Web 2.0 represents the fastest growing and most innovative part of the IT industry. How can business software providers bring together these disparate front-end and back-end technologies and architectures? To understand what might happen, we must examine some key technology trends that are relevant for ERP.

High-Performance Computing

High-performance computing is one of the most important technology megatrends. By 2010, we'll see 10 to 100 cores per server along with 100-TByte memory available for business apps. The big driver for high-performance computing is multicore parallel computing. This will change the IT landscape like the PC's introduction did to the computer landscape, which was dominated by water-cooled host machines.

So far, software and hardware producers have lived in close symbiosis. Each new software release's appetite for performance has been fulfilled by faster computers with more powerful processors.

No More Free Lunch

Soon enough, this will change forever, and software companies won't be able to take advantage of Moore's law for "free" any longer. The aggregate processor performance will jump orders of magnitude, but the clock speed of each single core will max out because of the heat that would be generated by further increases.

Software must take advantage of the massive parallelism that multicore chips offer or else it won't run faster on the next-generation chips. Recognizing this fact, software companies are now talking to chip and hardware producers to understand how to design for multicore processors and parallelize applications to increase performance on future multicore chips.

Many open questions face researchers and developers, including how to program multicore. Off-chip memory bandwidth is a big issue. How can you guarantee cache coherence across, say, 1,000 cores for symmetric multiprocessing (SMP)? Energy also enters as a new constraint as cores might need to be tuned down to avoid overheating. Also, compilers can't currently deal with locality and distance. The MIT Computer Science and Artificial Intelligence Laboratory's Multicore Computer Project is pursuing a vision designed to address that challenge by bringing space into the digital abstraction.[1] This highly interdisciplinary project intends to find a way to digitize space and rethink all layers of computer science (operating system, hypervisor, and so on) in light of this abstraction.

Web Services and Multicore

Business applications will be in big trouble if they don't take advantage of multicore. A logical place to leverage it is in Web services, which offer a natural granularity for parallelization of business software. They're well-suited to use multicore, multithread processors effectively. Business applications will be able to parallelize service execution by detecting the scope for parallelism automatically. They should also be able to achieve scalability via distributed instances of individual logical services.

SOA and Web Services Let CIOs Sleep Better

Big corporations typically spend between 1 and 5 per cent of their revenue on IT. Maintenance is a primary cost factor because necessary updates tend to be expensive as IT engineers must adapt all past customizations and add-ons to work with new versions. Given that testing, alone, after such updates costs the Fortune 500 companies billions of dollars, it would be preferable to partition ERP systems into smaller building blocks. Updating ERP business functions separately would limit complexity, and reduce ownership, innovation, and maintenance costs.

SOA lets developers partition and decouple applications. It provides a bridge between incompatible tech-

nologies, thus increasing *interoperability* — a topic that's high on the list of things that keep CIOs awake at night.

Adapt or Die

Business processes become less static in the fast-changing business world. A shorter software life cycle would let companies adapt applications to changing business environments without having to wait for new updates and for their IT departments to implement the changes. It currently takes several months to change business software — often longer than it takes companies to change the business processes.

Web services are ideal for letting developers adapt ERP to fast-changing business processes. Web services can be orchestrated to create multiservice business processes that connect via SOA to the robust, reliable back end, thus adding easy-to-use additional functionality.

Two Birds; One Stone

Web services can take advantage of multicore computing. At the same time, they let us build applications that adapt easily to changing business environments.

Web services can also help assemble full-blown business applications that will initially be for the small-and-medium business market. Yet, assembling complex apps from Web services won't be easy. How will users find the right services from among hundreds or perhaps even thousands in a repository? Technologies such as semantics will help to discover and orchestrate these services. When prefabricated services don't deliver the desired functionality, developers will have to adapt them and glue the services together to use them as bigger building blocks. Dynamic scripting languages such as Ruby will play an important role as they enable modifications to services in a declarative way, defining additional functionality and gluing the services together into larger building blocks. Such languages will also bridge the ERP and Web 2.0 worlds. The key is to invite the best brains to program business apps by running dynamic scripting languages on the ERP stack.

Dynamic scripting languages also offer the benefits of high-performance computing to a broad community of software developers without knowledge of implementation details. The Multicore Association's Multicore Communications API (MCAPI) working group proposes a Ruby API for its MCAPI standard (www.multicore-association.org/workgroup/comapi.php).

Real-Time ERP

One way to rejuvenate ERP would be to add a real-time stack for massive parallel real-time service execution, like those used at Amazon and Google. Many future ERP enhancements will also need real-time data processing — handling hundreds of thousands of events per second to, for example, integrate sensor data, deliver business intelligence, run complex event processing, or complete real-time billing (phone companies) and meter reading (utilities).

Multicore Is a Disruptive Innovation

Decentralized real-time information storage is an immediate application of multicore multithreading. Consider, for example, the billing process at most telecommunications providers. At the end of each month, it takes days to build the customer bills by searching through and aggregating data from the massive amounts of customer usage data stored in huge RDBs. Defining one thread per customer lets the providers add usage information after each call. The threads can be distributed because they're independent of each other, and the provider can trigger the customer bill at any time by adding customer master data (address, rate, and so on) to the usage data accumulated by a given thread.

The thread agility of some concurrency platforms, such as Microsoft's Robotics Studio with its concurrency and coordination runtime (CCR) and decentralized software services (DSS) features, support business applications with long-living threads. Threads can go to sleep on disk and be executed at any time later in memory. Such concurrency platforms enable massively parallel computing: as one simulation of the US pharmaceutical supply chain illustrated, they can handle more than 5 Tbytes of data created each week.[2]

ERP will need a multithreading stack for massive real-time service execution, like at Amazon and Google, as well as for connecting to sensor networks, complex-event processing, real-time billing, and so on. Introducing a multithreading stack for business applications will take advantage of future quantum leaps in chip performance by transitioning to Web services that naturally partition the software into parallel threads. It isn't obvious how to transform existing ERP platforms to multithreaded models, execute suitable primitives in multiple threads, add a new multicore stack, or reengineer the existing platform. Many companies, including SAP, have these types of projects under way. Regardless of how it is accomplished, transforming existing business applications to multithreaded models will be a very challenging task.

Mobile Computing

Pervasive connectivity and ubiquitous device-to-device connectivity is yet another megatrend. Estimates suggest that there are some 2 billion mobile users worldwide — more than 20 percent of the Earth's population. Wireless and embedded computing is becoming mainstream as handhelds,

wearable systems, appliances, cars, medical implants, and other devices with considerable computing power continue to make our lives easier, almost without our noticing.

Ubiquitous and mobile computing will connect the Internet to objects that move along the supply chain and throughout everyday life. (We'll be able to Google our lost keys.). Mobile phones will dominate among Web-access devices, spurring myriad user-centric apps running on dedicated hardware. The iPhone and SAP's collaboration with RIM to access enterprise apps via BlackBerry are just the beginning. The traditional model of Web access via a browser running on an Intel chip with Microsoft Windows is losing importance. Most new Web users will be using mobile phones to access the Web. I think that's why Microsoft would like to buy Yahoo, and it's another reason that lightweight open browsers such as Safari are likely to see tremendous growth in usage among developers.

The Mobile World Will Conquer from the Front End

Pervasive computing will accelerate mobile usage of lightweight business applications in the front end. Mobile devices bring together Web 2.0 computing and front-end apps. Take, for example, Ajax and Safari, which business-application companies as well as consumer-application companies use to create software as diverse as mobile CRM, mobile maintenance, and mobile inspection (quality, safety, sanitation, etc.) for hotels, retail, hospitals and airlines.

Web services and multicore are among the most important enablers for these developments. Web services and SOA will not only help reduce maintenance costs for classical ERP but also make it more flexible so that it can adapt to fast-changing business processes. Expert business users and ecosystem developers will be able to use Web services to orchestrate new business software. Dynamic scripting languages such as Ruby will play an important role in adapting raw Web services to usable building blocks and building add-ons for classical ERP.

Long Live ERP

Business software companies will push the edge of the classical large-scale online transaction processing (OLTP) ERP as they continue to run larger numbers of transactions in areas such as banking and government. With an ERP that handles millions of transactions per second and writes about 2.5 Tbytes per second to disk, Deutsche Postbank has reached the limit for bank retail business automation with existing systems. Automating big banks, such as Citibank, will be a future challenge for large-scale OLPT ERP systems.

Thanks to these mega ERP applications and the long life cycle, we'll see ERP for a long time to come. After all, some chemical companies still use 30-year-old host COBOL software running on IMS for parts of their accounting, such as calculating inventory values for chemicals produced in complex networks of interlinked bills of materials.

Disruptive Business Models

The classical ERP revenue stream is changing as disruptive business models are emerging. Business software makers currently derive their revenue from license sales, professional services, and annual maintenance. In the future, they'll have to look to the new business model developed by SaaS vendors. The high operating margin for companies such as Salesforce.com, WebEx, Workday, and Amazon (in its role as business service provider) comes from the fact that they have no incremental costs once they've built the software and ensured that it runs on the platform. Open source companies have shown that, even for software installed on premises, the revenue doesn't have to come from selling the applications but rather from offering the services to install, configure, and run them.

Getting More Strategic

On the customer side, IT departments will need to focus less on running data centers and more on how to strategically support their companies' businesses. For instance, Intel's IT chief architect Gregg Wyant is responsible for all IT-related innovations, and he is currently overseeing Intel IT's plan for consolidating from 117 to 8 data center locations worldwide. All big corporations push virtualization to take advantage of high performance computing and cheap large-scale information stores. Running their many (legacy) back-end applications in virtual environments will help companies reduce costs and risks and gain manageability.

Dynamic scripting languages such as Ruby will play an important role in adapting raw Web services.

Information Workers Will Take Over

In addition to hardware and software consolidation, IT departments will undergo another big transformation as companies need less support from them. Business process (or ecosystem) experts will adapt their corporate apps with the Web services they

find at service parks such as Salesforce.com's AppExchange or SAP's Enterprise Service Repository.[3]

For the next step, Web-scale architectures like those offered by Google and Amazon will see wide use not only for consumer applications but also for cloud computing and business-process outsourcing. Amazon Services already runs the e-business for America Online, Target, and others. Amazon handles inventory and runs fulfillment and customer services. For example, visitors to Target.com will see the familiar Target bull's-eye logo, but Amazon is making it work (www.seomarketingresearch.com/reverse-outsourcing/).

In the near future, we'll see ERP systems run in the cloud even for multibillion-dollar companies. Intel, for example, intends to achieve a net present value of US$550 to 650 million and overall cost-avoidance savings of US$1 billion or more through its data center efficiency program (www.intel.com/it/apr.htm). The three underlying tenets are: standardizing processes and design specifications, increasing compute utilization, and reducing data centers through consolidation.

One day, some ERP instances (or even one) will be available as SaaS for the Fortune 500 companies, although that will take a while. After all, it took Unix about 35 years to make it from inception to running shop floors, planes, power plants, and so on.

But two strong forces are coming to bear. From the bottom, the economics of multicore technologies are making SaaS and cloud computing the inevitable winners. From the front end, Web 2.0 for the enterprise will let smart people script services for special purposes that connect to the cloud and run on mobile devices.

With people expecting ERP at their fingertips, using their PDAs to take advantage of massive power on the back end, enterprise IT is likely to be in for radical changes.

References

1. A. Agarwal et al., "The Angstrom Multicore Computer Project: Bringing Space into the Digital Abstraction," presented at the 1st Ann. Industrial Affiliates Program Conf., MIT, 2008.
2. J.R. Williams et al., "Modeling Supply Chain Network Traffic," *RFID Technology and Applications*, S.B. Miles, S.E. Sarma and J.R. Williams, eds., Cambridge University Press, 2008, p. 87; www.cambridge.org/catalogue/catalogue.asp?isbn=9780521880930.
3. C. Petrie and C. Bussler, "The Myth of Open Web Services: The Rise of the Service Parks," *IEEE Internet Computing*, May/Jun 2008; http://www-cdr.stanford.edu/~petrie/online/peer2peer/serviceparks.pdf.

Paul Hofmann is vice-president of research at SAP Labs, Palo Alto. His research interests include IT productivity, supply-chain management, artificial intelligence, semantics, and Web services. Hofmann has a PhD in Physics from Technical University at Darmstadt, Germany. He is a member of the IEEE and has written several books on supply-chain management. Contact him at paul.hofmann@sap.com.