BPM: WILL IT BE THE WAVE OF THE FUTURE?

July 2003 Volume 3 Issue 7

WebServices JOURNAL

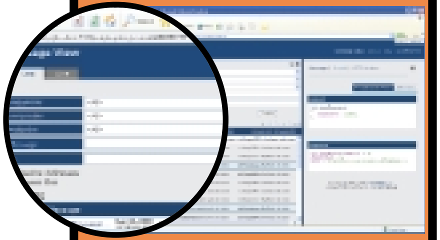
.NET J2EE XML

WSJ2.COM

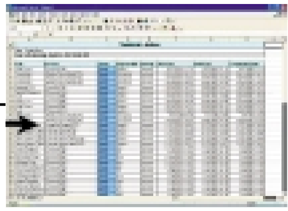
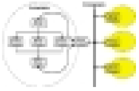
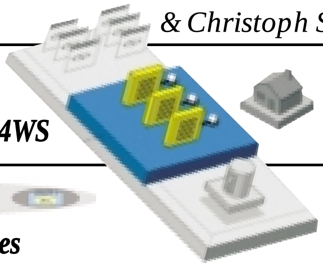
BPEL Make Your Services Flow

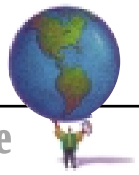
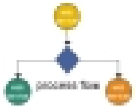
Composing Web services into business flows

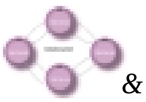
PAGE 16



FOCUS ON BPM



WebServices JOURNAL
.NET J2EE XML

INTERNATIONAL ADVISORY BOARD
Andrew Astor, David Chappell, Graham Glass, Tyler Jewell,
Paul Lipton, Anne Thomas Manes, Norbert Mikula,
Frank Moss, George Paolini, James Phillips, Simon Phipps

TECHNICAL ADVISORY BOARD
Bernhard Borges, JP Morgenthal, Andy Roberts,
Michael A. Sick, Simeon Simeonov

EDITORIAL
EDITOR-IN-CHIEF
Sean Rhody sean@sys-con.com
EDITORIAL DIRECTOR
Jeremy Geelan jeremy@sys-con.com
INDUSTRY EDITOR
Norbert Mikula norbert@sys-con.com
PRODUCT REVIEW EDITOR
Joe Mitchko joe@sys-con.com
.NET EDITOR
Dave Rader davidr@fusiontech.com
TECHNICAL EDITORS
Andrew Astor aastor@webmethods.com
David Chappell chappell@sonicsoftware.com
Anne Thomas Manes anne@manes.net
Mike Sick mike_a_sick@yahoo.com
EXECUTIVE EDITOR
Gail Schultz gail@sys-con.com
EDITOR
Nancy Valentine nancy@sys-con.com
ASSOCIATE EDITORS
Jamie Matusow jamie@sys-con.com
Jean Cassidy jean@sys-con.com
ASSISTANT EDITOR
Jennifer Van Winckel jennifer@sys-con.com

PRODUCTION
PRODUCTION CONSULTANT
Jim Morgan jim@sys-con.com
LEAD DESIGNER
Richard Silverberg richards@sys-con.com
ART DIRECTOR
Alex Botero alex@sys-con.com
ASSOCIATE ART DIRECTOR
Louis Cuffari louis@sys-con.com
ASSISTANT ART DIRECTOR
Tami Beatty tami @sys-con.com

CONTRIBUTORS TO THIS ISSUE
Michael Blank, Ryan Cairns, David Cameron,
Alejandro Danylyszyn, Peter Fingar, Mark Little,
Joan Lynn, Jay Pellant, Chris Peltz, Harvey Reed,
Sean Rhody, Cesare Rotundo, Christoph Schittko,
Doron Sherman, Howard Smith, Derick Townsend, Jim Webber

EDITORIAL OFFICES
SYS-CON MEDIA
135 CHESTNUT RIDGE ROAD, MONTVALE, NJ 07645
TELEPHONE: 201 802-3000 FAX: 201 782-9637
WEB SERVICES JOURNAL (ISSN# 1535-6906)
Is published monthly (12 times a year)
By SYS-CON Publications, Inc.
Periodicals postage pending
Montvale, NJ 07645 and additional mailing offices
POSTMASTER: Send address changes to:
WEB SERVICES JOURNAL, SYS-CON Publications, Inc.
135 Chestnut Ridge Road, Montvale, NJ 07645

Dance Lessons

Written by
Sean Rhody

Author Bio:

Sean Rhody is the editor-in-chief of Web Services Journal *and managing editor of* WebLogic Developer's Journal. *He is a respected industry expert and a consultant with a leading consulting services company.*
SEAN@SYS-CON.COM

Being left handed and, therefore, left footed, I once tried to take dancing lessons so I'd look a little less ungainly at family functions like weddings. My father, who's ambidextrous, makes it look easy. Of course, I always forget that when he was growing up, dancing was the main social activity. Naturally I found it much more difficult than it looks.

Web services choreography, orchestration, and general business process management are the programming equivalents of ballroom dancing. Web services itself is about making multiple computers work together. Sort of like a school dance, where the teachers come around and make the boys and girls dance together. The basics, the plumbing of Web services, UDDI, SOAP, WSDL, and XML, make the dance possible. They provide the underlying infrastructure that is necessary for multiple processes on multiple systems to communicate.

By themselves these basics can do a great deal of useful work. And fortunately, most software vendors have found it a relatively simple matter to add Web services support to their products, which provides the ubiquity needed for Web services to realize the "network effect."

But look at them closely and you realize that there are a host of additional things that you either want or need that just aren't provided by SOAP, WSDL, UDDI, and XML. Some of the most important functionality that is not present is the ability to construct a process out of a Web service, or from multiple Web services. It's easy to forget that Web services can have more than one method, and that you can construct a whole process (such as order processing) as a single set of methods if you so choose. What's not present is the ability to manage those processes, and combine them with branching, coordination, transactional integrity, and error handling.

Choreography and BPM are added-value aspects of Web services – much like the steps of a waltz, they organize what can otherwise be a chaotic occasion (ever seen a high school dance to modern music?). They are also likely to be the major vendor battleground over the next several years. The battle over plumbing is nearly over, with the standards evolving but not really changing all that much, and interoperability is there. Much like the J2EE world, where application server vendors first competed on standards but now work on value adds, the basics of Web services are becoming endemic in platforms. Two years ago, it was hard to find a platform with full support. Now it's surprising if the support isn't there. Two years from now, the plumbing will be forgotten and we will be working on a higher level of abstraction once again.

In the meantime, the BPM vendors have an education process to complete. Like most revolutionary tools, until you've been presented the value proposition it's difficult to imagine how to use such a tool, and what value it can bring to an organization. While I don't have space to try to put such a proposition into words here, think "agility." BPM is all about reacting quickly to business changes, and being able to redefine the business process as needed, instead of over a yearly update cycle.

Naturally, we've focused this month on those tools and standards that assist with BPM. We wanted to give you a glimpse of what the future holds for you. So read on, while I put these funny looking footprints on the floor and try to learn the Macarena… ⓔ

Written by Jim Webber & Mark Little

Introducing WS-Transaction

Part II

Using business activities

In July 2002, BEA, IBM, and Microsoft released a trio of specifications designed to support business transactions over Web services. BPEL4WS, WS-Transaction, and WS-Coordination together form the bedrock for reliably choreographing Web services–based applications.

In our previous articles (***WSJ***, Vol. 3, issues 5 and 6), we introduced WS-Coordination, a generic coordination framework for Web services, and showed how the WS-Coordination protocol can be augmented to provide atomic transactionality for Web services via the WS-Transaction Atomic Transaction model.

This article looks at support for extended transactions across Web services. We also show how these can be used to provide the basis for higher-level business process management and workflow technology.

Business Activities

Most business-to-business applications require transactional support in order to guarantee consistent outcome and correct execution. These applications often involve long-running computations, loosely coupled systems, and components that don't share data, location, or administration. It's difficult to incorporate atomic transactions within such architectures. For example, an online bookshop may reserve books for an individual for a specific period of time, but if the individual doesn't purchase the books within that period they will be "put back onto the shelf" for others to buy. Furthermore, because it is impossible for anyone to have an infinite supply of stock, some online shops may appear to reserve items, but in fact may allow others to preempt that reservation (i.e., the same book may be "reserved" for multiple users concurrently); a user may subsequently find that the item is no longer available, or has to be reordered for them.

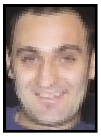
A business activity (BA) is designed specifically for these long-duration interactions, where exclusively locking resources is impossible or impractical. In this model, services are requested to do work, and where those services have the ability to undo any work, they inform the BA so that if the BA later decides to cancel the work (i.e., if the business activity suffers a failure), it can instruct the service to execute its undo behavior. The key point for business activities is that how services do their work and provide compensation mechanisms is not the domain of the WS-Transaction specification, but an implementation decision for the service provider.

The BA defines a protocol for Web services–based applications to enable existing business processing and workflow systems to wrap their proprietary mechanisms and interoperate across implementations and business boundaries.

A BA may be partitioned into *scopes* – business tasks or units of work using a collection of Web services. Scopes can be nested to arbitrary degrees, forming parent and child relationships, where a parent scope can select which child tasks to include in the overall outcome protocol for a specific business activity, so nonatomic outcomes are possible. In a manner similar to traditional nested transactions, if a child task experiences an error it can be caught by the parent, who may be able to compensate and continue processing.

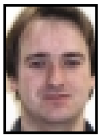
When a child task completes it can either leave the business activity or signal to the

AUTHOR BIO:



Dr. Jim Webber is an architect and Web services fanatic at Arjuna Technologies, where he works on Web services transactioning and Grid computing technology. Prior to joining Arjuna Technologies, he was the lead developer with Hewlett-Packard working on their BTP-based Web Services Transactions product – the industry's first Web Services Transaction solution. An active speaker and Web Services proponent, Jim is the coauthor of Developing Enterprise Web Services.

JIM.WEBBER@ARJUNA.COM



Dr. Mark Little is chief architect, transactions, for Arjuna Technologies Limited, a spin-off from Hewlett-Packard that develops transaction technologies for J2EE and Web services. Previously, Mark was a distinguished engineer and architect at HP Middleware, where he led the transactions teams. He is a member of the expert group for JSR 95 and JSR 117, is the specification lead for JSR 156, and is active on the OTS Revision Task Force and the OASIS Business Transactions Protocol specification. Mark is the coauthor of J2EE 1.4 Bible *and* Java Transactions for Architects.

MARK.LITTLE@ARJUNA.COM

parent that the work it has done can be compensated later. In the latter case, the compensation task may be called by the parent should it ultimately need to undo the work performed by the child.

Unlike the atomic transaction protocol model, where participants inform the coordinator of their state only when asked, a task within a BA can specify its outcome to the parent directly without waiting for a request. When tasks fail, the notification can be used by the business activity exception handler to modify the goals and drive processing forward without waiting meekly until the end of the transaction to admit to having failed – a well-designed BA should be proactive if it is to be performant.

Underpinning all of this are three fundamental assumptions:

- All state transitions are reliably recorded, including application state and coordination metadata (the record of sent and received messages).
- All request messages are acknowledged, so problems are detected as early as possible. This eliminates unnecessary tasks and can detect a problem earlier, when rectifying it is simpler and less expensive.
- As with atomic transactions, a response is defined as a separate operation and not as the output of the request. Message input-output implementations will typically have timeouts that are too short for some business activity responses. If the response is not received after a timeout, it is sent again. This is repeated until a response is received. The request receiver discards all but one identical request received.

The business activity model has multiple protocols: BusinessAgreement and BusinessAgreementWithComplete. However, unlike the AT protocol, which is driven from the coordinator down to participants, this protocol is driven from the participants upwards.

Under the BusinessAgreement protocol, a child activity is initially created in the Active state; if it finishes the work it was created to do and no more participation is required within the scope of the BA (such as when the activity operates on immutable data), the child can unilaterally send an exited message to the parent. However, if the child task finishes and wishes to continue in the BA, then it must be able to compensate for the work it has performed. In this case it sends a completed message to the parent and waits to receive the final outcome of the BA from the parent. This outcome will be either a close message – the BA has completed successfully – or a compensate message – the parent activity requires that the child task reverse its work.

The BusinessAgreementWithComplete protocol is identical to the BusinessAgreement protocol with the exception that the child cannot autonomously decide to end its participation in the business activity, even if it can be compensated. Rather, the child task relies upon the parent to inform it when the child has received all requests for it to perform work. The parent does this by sending the complete message to the child, which then acts as it does in the BusinessAgreement protocol.

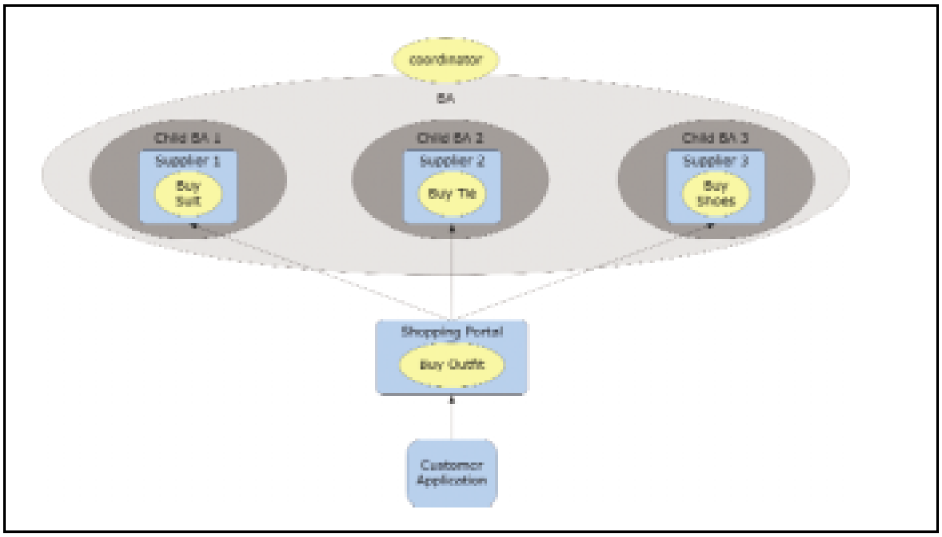


FIGURE 1 Using a business activity to support compensating transactions across enterprise boundaries

The crux of the BA model, compared to the AT model, is that it allows the participation of services that cannot or will not lock resources for extended periods.

While the full ACID semantics are not maintained by a BA, consistency can be maintained through compensation, although writing correct compensating actions (and thus overall system consistency) is delegated to the developers of the services controlled by the BA. Such compensations may use backward error recovery, but typically employ forward recovery.

Coordinating Business Activities on the Web

However, the real beauty of the Web services model is that it is highly modular. Capitalizing on that modularity, consider the case shown in Figure 1, where a shopping portal uses several suppliers to deliver a richer shopping experience to the customer.

In this case, a BA is used since there is no close trust relationship between any of the suppliers (indeed they are probably competitors), and purchases are committed immediately as per the BA model. In the non-failure case, things are straightforward and each child BA reports back that it has completed to the coordinator via a completed message.

The failure case, however, is a little more interesting (see Figure 2). Let's assume that Supplier 2 could not source the tie that the customer wanted and its corresponding BA fails. It reports the failure back to the coordinator through a faulted message. On receiving this message, the logic driving the BA, which we assume to be a workflow script residing in the portal service, is invoked to deal with the fault. In this case, the logic uses forward error recovery to try to obtain the item from an alternative supplier.

If the forward error recovery works, and the alternate supplier's Web service confirms that it is able to source the desired item, then the BA proceeds normally, executing subsequent child BAs until completion. If, however, the BA cannot make forward progress and it thus has no option but to go backwards and compensate previous successfully completed activities. Note that failed activities are not compensated because their state is, by definition, unknown.

Once the compensation has taken place

successfully (remember that an added complexity is that compensations can themselves fail), the system should be in a state that is semantically equivalent to the state it was in before the purchase operations were carried out. The shopping portal service knows the status of the transaction from the coordinator, and can then report back to the customer application that the order didn't complete.

Business Activities and BPEL4WS

During the execution of a business process, like our shopping portal example, data in the various systems that the process encompasses changes. Normally such data is held in mission-critical enterprise databases and queues, which have ACID transactional properties to ensure data integrity. This can lead to a situation whereby a number of valid commits to databases could have been made during the course of a process, but where the overall process might fail, leaving work partially completed. In such situations the reversal of partial work cannot rely on backward error recovery mechanisms – rollback – supported by the databases since the updates to the database will have been long since committed. Instead, we must compensate at the application level by performing the logical reverse of each activity that was executed as part of our process, from the most recently executed scope back to the earliest executed scope. This model is known as a *saga*, and is the default compensation model supported by BPEL4WS.

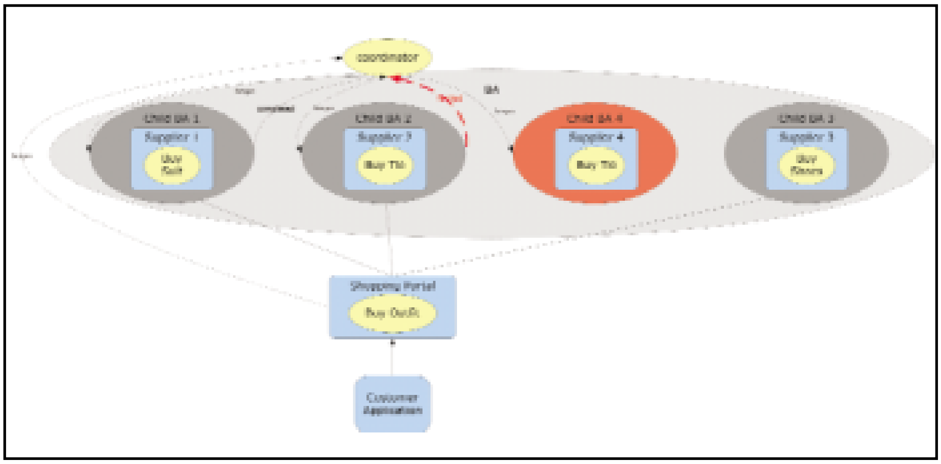


FIGURE 2 Error handling and forward recovery

The BPEL4WS specification suggests WS-Transaction Business Activity as the protocol of choice for managing transactions that support the interactions of process instances running within different enterprise systems. A business activity is used both as the means of grouping distributed activities into a single logical unit of work and the dissemination of the outcome of that unit of work – whether all scopes completed successfully or need to be compensated.

If each of the Web services in our shopping portal example were implemented as BPEL4WS workflow scripts, the messages from the BA protocol messages from the coordinator could be consumed by those workflow scripts and used to instigate any compensating activities for those activities. The execution of compensating activities caused by the coordinator sending compensate messages to the participants returns the process as a whole to the same state logically as it was before the process executed.

Relationship to OASIS BTP

The OASIS Business Transactions Protocol (BTP) was developed by a consortium of companies, including Hewlett-Packard, Oracle, and BEA, to tackle a similar problem to WS-Transaction: business-to-business transactions in loosely coupled domains. BTP was designed with loose coupling of services in mind and integration with existing enterprise transaction systems was not a high priority. Web services were also not the only deployment environment considered by the BTP developers so the specification only defines an XML protocol message set, and leaves the binding of this message set to specific deployment domains.

BTP defines two transaction models: *atoms*, which guarantee atomicity of decision among participants; and *cohesions*, which allow relaxed atomicity such that subsets of participants can see different outcomes in a controlled manner. Both models use a two-phase completion protocol, which deliberately does not require ACID semantics: although it is similar to the 2PC protocol used by WS-Transaction Atomic Transactions, it is used purely to attain consensus and no semantics can be inferred from higher-level services that use atoms. An implementer of a BTP participant is free to use compensation techniques in the second-phase operations to guarantee atomicity if that model best suits the business.

Both atoms and cohesions also use the open-top coordination protocol, whereby both phases of the two-phase protocol must be explicitly executed by users. Because no time limit is implied between the two phases of the completion protocol, this explicit separation of the phases is intended to allow businesses to better model their business processes.

Although at least in theory WS-Transaction and BTP are intended to address the same problem domain, there are significant differences between them. BTP allows business-level negotiation to occur during many points in the protocol in its Qualifier mechanism; WS-Transaction does not have such a capability.

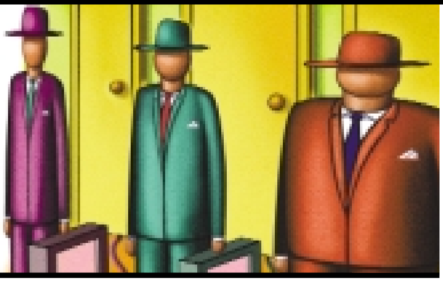
Summary

Over the course of these articles, we've seen both the atomic AT protocol and the non-ACID BA designed to support long-running transactions. While both the AT and BA models will be available to Web services developers directly through toolkits, it is the BA model that is supported by the BPEL4WS standard to provide distributed transaction support for business processes.

Written by Howard Smith & Peter Fingar

Twenty-First Century Business Architecture

The future is here



While the vision of process management is not new, existing theories and systems have not been able to cope with the reality of business processes – until now. By placing business processes on center stage, as first class citizens in computing, corporations can gain the capabilities they need to innovate, reenergize performance, and deliver the value today's markets demand.

AUTHOR BIOS:

Howard Smith is CTO (Europe) of Computer Sciences Corporation and cochair of the Business Process Management Initiative (BPMI.org). Computer Sciences Corporation has been applying BPM thinking to the design of enterprise architectures since 1998. He is the coauthor of Business Process Management: The Third Wave *(www.bpm3.com).*
AUTHORS@BPM3.COM

Peter Fingar is an executive partner with the digital strategy firm, the Greystone Group. He delivers keynotes worldwide and is the author of The Death of "e" and The Birth of the Real New Economy *and* Enterprise E-Commerce. *He is the coauthor of* Business Process Management: The Third Wave *(www.bpm3.com).*
AUTHORS@BPM3.COM

Business process management (BPM) systems discover what you do, and then manage the life cycle of improvement and optimization in a way that translates directly to operation. They see the world in terms of processes using notations and representations business people intuitively understand, and reflect the nature of the way business has to be – connected, collaborative, asynchronous, coordinated, conversational, and constantly changing.

Reengineering Reengineering

During the business process reengineering wave of the 1990s, management prophets' books of stories about other companies were all you had to guide the transformation of your business. Although the underlying theories were based on age-old common sense and general systems theory proposed 50 years earlier, reengineering advocates offered no path to execution. New processes could be envisaged but what happened next? There was no *engineering* in reengineering. Instead, processes were handed off – or, more precisely, thrown over the wall – to IT.

By contrast, the process-managed enterprise takes control of internal processes and communicates with a universal process language that enables partners and internal business units to execute on a shared vision – to understand each other's operations in detail, jointly design processes, and manage the entire life cycle of their business improvement initiatives. Companies embracing this approach to enterprise computing are using a new class of mission-critical infrastructure, a new category of software, the Business Process Management System (BPMS) – a *business platform* for business processes that exploits a company's existing technology infrastructures and assets.

Today, the vast majority of employees in large enterprises rely on nothing more than e-mail, spreadsheets, and word processing tools to coordinate work. Beyond this, automation is provided by expensive software applications maintained solely in the data center and by the staff of the IT department. Yet the majority of automation tasks needed each and every day in business are modest in relation to the complexity of today's IT systems. For example, nearly everyone needs more visibility of and control over the activities around them, as they interact with colleagues, partners and customers. Such iterations and communications are indeed the essence of business processes. Business users need control of information flows so that everyone remains focused on the task and is coordinated with everyone else – business processes follow no simple pattern and cannot be packaged easily.

Perhaps 80% of process-related tasks and their coordination could be designed and implemented by business people themselves – if only they had properly designed tools to enable them to directly manipulate their business processes. Moreover, business people should be able to implement changes to live business processes, meaning that the life cycle of the process design and modification needs to be where the process is used, not in the data center. Such environments are possible as companies acquire a BPMS capability. By contrast, business intelligence and action lag behind the current business activity if business processes are ingrained in rigid and brittle software systems.

Direct Representation and Manipulation

When business people develop numerical models using spreadsheets, they do not confuse the model and tool – the model and its representation has nothing to do with information technology. For example, a spreadsheet model might represent a budget, or an analysis of an engineering part. In contrast, the distinction between model and tool is frequently lost when trying to define BPM, which is often equated with systems integration or composite software development. While a BPMS may be integrated with other computing systems, and while the ingrained processes in those systems may be reused to create new process models, the process model itself has little to do with "systems integration" or "composite applications," and a lot to do with the budgeting or the engineering process.

> “BPM systems are helping organizations to obliterate, not just bridge, the business-IT divide by placing control of business processes directly in the hands of business people”

Many in the IT industry perceive BPM only as a better, faster, cheaper way to integrate applications, and this view is exacerbated by the focus on languages used to support Web services orchestration, such as BPEL. For all that is written about such languages you would think that BPM is only about systems interoperability, application integration, and a smart new way to develop more software. This thinking totally misses the point. BPM is about better, faster, cheaper business processes, not better, faster, cheaper technology.

BPM technologies provide *direct representation* of business processes, and then open those processes to complete life-cycle management: from discovery to design, deployment, execution, operations, analysis, and optimization. Tell business people that BPM is about technical integration and watch their eyes glaze over. Tell them their "problem" is applications integration or composite applications and watch them excuse themselves from the conversation.

In short, integration technology, however wrapped in process clothing, solves only an integration need. This is not to say that integration products cannot evolve to become BPM products, or that BPM products cannot provide integration, but the distinction needs to be made. What distinguishes BPMS is its central focus on the *direct representation and manipulation* of business processes, just as RDBMS provides the representation and manipulation of business data and the spreadsheet provides the representation and manipulation of numerical data. On the other hand, a comprehensive BPMS incorporates robust application integration facilities, as corporations need to integrate automated processes in legacy systems or best-of-breed packages to the BPM level – integrate once to the BPM level of abstraction, then develop and manage many business processes without returning to the technology plumbing. Processes embedded in legacy systems can be made "reusable," and are mandatory participants in many of the business processes companies wish to manage more actively and directly.

BPM systems are helping organizations to obliterate, not just bridge, the business-IT divide by placing control of business processes directly in the hands of business people, including front-line workers. Personal, workgroup, and departmental BPM tools, akin to tools commonly found in office productivity suites, are emerging. The role of IT is changing, away from custom development of more and more application software and toward the provision of BPM systems. Imagine a "Process Office" suite providing an integrated, process-centric approach to collaboration, computation, work management, process modeling, and simulation.

Aberdeen Group elaborates, "The BPM category may arguably provide the greatest return on investment compared to any other category available on the market today. BPM gives organizations the ability to cut operational costs at a time when the economic downturn makes it increasingly difficult to boost revenues... Business Process Management enables government agencies to dismantle obsolete bureaucratic divisions by cutting the labor- and paper-intensive inefficiency from manual, back-end processes. Faster and auditable processes allow employees to do more in less time, reducing paper use as well as administrative overhead and resources." In short, BPM is becoming the bedrock for a whole new world of *process work*.

Imagine a sales campaign "application." It could be developed upon a relational database management system (RDBMS), but would the data model and software provide the flexibility required? Would such an application naturally fit and adapt to the business process? Companies in different industries have diverse needs for sales campaign automation, and individual companies in the same sector compete with each other by differentiating sales processes. Packaging a sales-campaign application in software on a static data model seems inappropriate. Not only is each sales campaign in each company different, they are different within the same company for different types of products and services. In addition, as each campaign progresses, processes associated with each prospective customer may have to vary widely from the initial "sales plan." Therefore, instead of packaging the sales campaign as a software application, why not deliver it as a process? Give business people tools to build their own sales process. Allow them to customize the process for each customer. Give them the tools to include participants in the campaign as required, including employees, partners, systems, and information sources. Let the BPMS manage the end-to-end state of all processes. Provide business people with the tools they need to query the state of the campaign along key dimensions such as customer, product, and part; and based on this business intelligence, make adjustments to the process in order to respond to individual customer needs.

A Formal Foundation

Many trends have converged to create the brave new world of business process management – workflow management, business process modeling, quality management, business reengineering, change management, and distributed computing, to name but a few. Yet there was a vital and missing ingredient, the direct manipulation of business processes. The IT facade behind today's business processes (consisting of disjointed data models, application logic, workflows, and integration systems, repeated a hundred times in a hundred silos) can now be rationalized, not by replacing previous investments but by exploiting what they offer in combination, recast in the form of new process models and systems. It has taken the IT industry 20 years to find a way to represent the computational elements needed for a unified *process representation* on which to build tools that can be used to conceive, compose, and put new processes into operation.

The unifying theories needed for *business process computing* lie in an obscure branch of mathematics called the pi-calculus, whose conceptual father is Robin Milner, professor of computer science at Cambridge University and a Turing Award winner. Pi-calculus plays a role in BPMS similar to the roles finite state machines play in the business spreadsheet and the relational algebra in database management systems. Pi-calculus and related formalisms are complex, but business people couldn't care less about formalisms. On the other hand, the automation tools they use, each and every day, depend upon such science for robustness and reliability. By representing business processes in a mathematically formalized way, processes developed in one part of the business, or by a business partner, can be connected, combined and analyzed in real time, providing a foundation for the true real-time enterprise behind the real-time enterprise slogan. While the notion of a real-time enterprise is all about agility, and while the basis for *technical agility* may be a service-oriented architecture (SOA), the basis for *business agility* is BPM, making the SOA necessary but insufficient for meeting today's business needs. Just as the operating system emerged as the platform for the RDBMS capability, Web services and SOA are emerging as the platform for BPMS capability.

The central insight of pi-calculus is that all processes are acts of *communication*. This paradigm has enabled the Business Process Management Initiative (BPMI.org) to define document structures that capture the day-to-day communication that occurs in business at all levels – formal, informal, asynchronous, synchronous, human originated, or machine initiated. Such process document structures can be used to define any process, from the highest level of business strategy to the most basic numerical computation. Process-based documents can evolve with the business – just like spreadsheets and word processing documents do today. Milner calls such processes *mobile*, reflecting the dynamic, agile, real-time and adaptive nature of real business processes, not the rigid automation functions of typical hard-coded computer applications.

Unlike application packages, the BPMS adapts to a company's processes, not the other way around. The BPMS platform is targeted at a new hybrid business role that combines the skills of the enterprise data architect and enterprise business architect, allowing them to create process tools for all employees to power their work, each and every day – the *process architect* is the true architect of 21st century business, and BPMS is the foundation of 21st century enterprise architecture. To wit, a global telecommunications operator moving into broadband used BPM to facilitate the aggressive acquisition of large numbers of new customers, accelerating customer satisfaction well beyond the competition. Flexible new processes allowed the operator to collect, store, and queue orders to ensure that customers did not experience outages from failures in dependent systems operated by third parties. BPM allowed the customer service staff to be flexible in responding to numerous and diverse customer requests – it seems every customer has a custom need that demands custom business processes to fulfill. BPM insulated the customer service representative from changes occurring in third-party service providers and the changes arising from the unbundling of service elements as a result of deregulation.

The Business Process Management System

Process management borrows and combines features from a number of familiar tools and technologies, but differs in its central focus on communicating processes. BPM feels similar to computer-aided software engineering (CASE) because of its emphasis on graphical notation, collaborative discovery, and design. It shares with workflow management a focus on scripted events and task management. From the viewpoint of the systems architect, comparisons can be drawn with transaction processing monitors and application servers. For ERP practitioners, BPM's focus on process definition and optimization will have strong associations. Developers who have struggled with legacy system integration and who have employed enterprise application integration solutions will recognize similar ideas in BPM, especially where applications are to play key roles in end-to-end business processes. Process analysis tools used in conjunction with the BPMS will be familiar to users of online analytical processing (OLAP).

Because of these prior experiences, a company's existing IT skills can be readily transferred to the world of BPM. However, do not let these similarities lead you to conclude that BPM is simply a repackaging of existing technology – this year's latest IT branding. Although many vendors will try to hijack the BPM title, the reality of BPM is that it is unique in its ability to provide a top-down approach to business design that isolates business users from the vagaries of the numerous enterprise application systems already in place. Process models act like "live applications," but they are only process-schemas deployed on and directly executed by the BPMS. There is no waterfall model of process development as there is in software development. Top to bottom, at all levels of the process model the process representation is the same and is directly executed by a BPMS in the way that an RDBMS directly executes database structured query language commands. But "direct execution" does not quite capture what is going on, for processes are often confused with more traditional software procedures or scripted workflows. Rather, the process definition is a declarative description of the *now* and the *future* – as instances of the process are created, like a new row being added to a spreadsheet or a database, they proceed in line with their design.

In look and feel, the BPMS is to the business process designer what a design workstation is to the automobile designer. The computer-aided design and computer-aided manufacturing (CAD/CAM) system of the automobile designer becomes the computer-aided modeling/computer-aided deployment (CAM/CAD) system of the business process designer. Underlying the BPMS, as in the case of CAD/CAM systems, is a digital representation and simulation of the real "thing" with which the designer is working. While the automobile designer works with digital artifacts such as tires, engines, body frames, and aerodynamics, the process designer works with digital artifacts such as orders, suppliers' fulfillment services, third-party billing services, bills for materials, shipping schedules of trading partners, and so on. When the automobile engineer pushes the "make it so button," the computer-aided manufacturing part of the system actually implements the building of the new car. When the business process engineer pushes the "make it so button," the computer-aided deployment part of the system actually implements the mission-critical, end-to-end business process across the disparate legacy systems inside the enterprise and across the value chain.

What about all the C++, Java, scripting, EAI, and other computer technologies that are involved? Where did they go in all this? They are still there, only now it is the BPMS that deals with them, not the designers and other business people who use the business process workstations and the underlying BPMS. With the BPMS, business information systems are developed and evolved by manipulating the business process directly using the language and concepts of business, not the language and concepts of machines.

BPM is all about raising the level of abstraction from machine concepts to business concepts. Although BPM isn't a panacea for all computing needs, it requires a deliberate step of abstraction and application integration to a common process model, and fulfills an increasing number of business-critical needs.

Crossing the Chasm

Business process management products are available from many vendors, in versions ranging from departmental workgroup solutions to enterprise-scale infrastructures – a spectrum of solutions to meet diverse needs. Not all BPM systems, by any means, use the pi-calculus formalism yet, or process languages built from it such as the Business Process Modeling Language (BPML) published by BPMI.org. But as other make-do approaches hit technological walls, this will change. The underlying mathematics of pi-calculus and the semantics of BPML are hard to ignore, for these foundations are paramount to robust and reliable business process management. Process languages, such as the vendor-sponsored Business Process Execution Language for Web Services (BPEL4WS), will converge and evolve towards the needs of a BPMS with a solid mathematical underpinning.

Today, BPEL is primarily advocated for loosely coupled application integration and development, but as the needs for BPM go well beyond Web services and simple workflow requirements, BPEL will require the same theoretical foundation. CIOs will rightly disregard any other simplistic BPM "layer" as "yet another point solution" unless BPM systems can be shown that they embody a strong formal model of enterprise computing and mobile processes. Only then can BPMS migrate from a niche category to a mainstream platform, similar to what companies already know and understand in other areas of IT support such as relational data management and network management. BPM is far more than another way to develop applications. BPMS is a platform that will support a raft of new processes, tools, and applications. A sales campaign isn't a software application – it's an application of process management.

How will BPM be assimilated by end-user organizations? There is no doubt that businesses will continue to look to their current software suppliers for BPM innovations, yet they need a true BPMS today even if their preferred supplier cannot deliver, opening the market for new entrants. In addition, companies that survived the turbulent era of reengineering may be tempted to feel that they have already reengineered, reinvented, mapped, analyzed, and improved every aspect of their business processes. The reality is that they know, deep down, that they have barely started and business processes are in a continual state of flux. The reengineering prophecy – "we've not done reengineering"– is indeed true.

Now we are in uncertain times again – a downturned economy, corporate scandals, and the IT Ice Age. Today, companies are experiencing not one broad-based economic reality, but a multitude of process-related problems. They absolutely must be able to do more with less. Early adopters of BPM systems will therefore be those companies that face the most severe process management problems, just as early adopters of relational database systems were those that faced severe data management problems. It is not easy to cast business process–related problems into neat little categories or magic quadrants and no pattern of an "ideal BPMS" has yet emerged. On the other hand, business processes and their management lie at the heart of all business activity. As a result, processes are taking center stage, driving demand for powerful BPM systems with "pi-calculus inside" that take the process complexity outside of stovepipe applications and allow existing applications to be expressed in a form that business people, not just programmers, can understand, evolve, and manage.

Sitting right at the divide between humans and machines – between business and IT – the BPMS represents a paradigm shift in the world of business automation – business process computing – that has a profound impact on the way companies structure and perform work, letting people speak in their native tongue and enabling machines to understand them, not vice versa. Designed top down and deployed directly in accordance with a company's strategy, business processes can now be unhindered by the constraints of existing IT systems.

The implications are equally profound for the IT industry, for it must enter its next phase of evolution and maturity. As CAD/CAM systems enabled computer-integrated and "just-in-time" manufacturing, BPM can facilitate collaborative "just-in-time" business processes and a new era of process manufacturing. Those players in the IT industry that master BPM will share the new wealth with their customers: productivity gains, innovation, and lowered costs like those the industrial design and manufacturing industries have already realized as a result of implementing a direct path from design to execution.

Welcome to the company of the future, the fully digitized corporation – the process-managed enterprise. Welcome to the next 50 years of business and IT.

BPEL Make Your Services Flow

written by Doron Sherman

Composing Web services into business flows

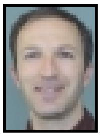
Until now, the options available for implementing business flows in a typical enterprise-computing environment were daunting. IT project managers had to choose between complex high-end EAI/BPM solutions and high-risk application development projects. More often than not, IT decision makers opted to do nothing and wait.

When IBM, Microsoft, and others submitted the BPEL4WS specification to OASIS, a compelling alternative to the traditional options became available. BPEL is to process orchestration what SQL was to data management. Its impact is significant; transforming application integration from a risky and expensive endeavor into a mainstream development practice. BPEL has the potential to create the foundation for new composite, Internet-scale, loosely coupled applications.

Toward Composite Business Flows

Business transactions, often run in isolation and confined to specific line-of-business applications, now require integration across different parts of the organization as well as allowing for manual intervention in support of better exception handling.

AUTHOR BIO:



Doron Sherman is the CTO of Collaxa, Inc., a Web service orchestration server vendor and a BEA Partner located in Redwood Shores, California. He has been involved with Java since its early days and pioneered application server technology while a founder and chief scientist at NetDynamics.

DORON@COLLAXA.COM

This new genre of applications is driving the emergence of an event-driven architecture (EDA) and the development of composite business flows across existing IT assets. It is inspiring renewed interest in EAI, BPM, and workflow solutions, but with radically new requirements. Unlike the stand-alone, monolithic nature of existing solutions, this genre calls for implementation of long-running business processes that coordinate flow and asynchronous conversations while allowing for effective handling of exceptions and performance of manual tasks.

Web services started as the technology answer to the need to ease connectivity to existing IT assets and enable simplified, all-XML application-to-application communication. Business agility, as well as other technical requirements such as scalability, reliability, and flexibility, drives users to assemble Web services into business processes that are themselves exposed as Web services.

An Example

Consider the following composite business flow example (see Figure 1), an availability-to-promise (ATP) application. A chemical manufacturer regularly accepts requests from customers of the form: "I want to buy 15,000 pounds of acrylic acid by September 30th." Processing time for this type of request normally spans several hours. This largely manual process tends to be prone to errors and inaccuracies caused by time latencies and the extensive involvement of humans for its completion. Slow and cumbersome processing of ATP requests often results in unsatisfactory service to customers and entices them to shop for alternative avenues. Reduction of processing time was clearly in the best interest of the chemical manufacturer.

In addition, the ATP process involved exception handling and mandated reporting of processing statistics to facilitate gradual ongoing improvements to the process. Better management of exceptions and greater visibility to decision makers at the chemical manufacturer are important goals. Finally, the inefficient handling of the manual process resulted in high transaction costs. Being able to handle a larger volume of ATP requests, each costing less to process, will increase profitability.

Technical Challenges of Flow Implementation

Automation of the ATP process through

implementation of a composite business flow raises a number of technical challenges – including data mapping – in transforming documents from one format to another to facilitate information exchange among the various applications involved, as well as present information to end users (e.g., customer and sales reps) during execution of the ATP request. Exchanging documents among end users and the various applications and services is inherently asynchronous and poses the technical challenge of handling asynchronous conversations among the cooperating parties.

Controlling proper routing of documents among the various participants also requires flow coordination. For example, an inventory levels check has to occur in parallel to production schedule verification.

The ATP process is a long-running business transaction, where some of its activities may be in need of reversal, or undo, when exceptions occur during execution. Maintaining execution integrity in the context of asynchronous conversations mandates the use of compensating transactions. Hence, cooperating services and applications taking part in the ATP process should provide undo logic to reverse the effect of actions taken during normal processing. This cancellation method is provided in lieu of a conventional atomic transaction mechanism, since resource locking is inappropriate here. Managing compensating transactions requires the use of a transaction coordinator to handle such cancellations.

Emerging Web Service Orchestration Stack

The first phase of evolution of the Web services stack (see Figure 2) focused on making IT assets available. This is the foundation of Web services use, namely publishing, which consists of: Internet-native transport protocols (e.g., HTTP, SMTP), data models based on XML structures, message exchanges over SOAP, and description of service operations and types. Adam Bosworth outlined the requirements for a basic Web services development model in "Developing Web Services" (www.w3.org/2001/03/WSWS-popa/paper53). According to Adam, Web services should support asynchronous messaging (e.g., handle unpredictable processing loads and unreliable communication channels), dynamic hardware scaling (e.g., offer dynamic load distribution and manage distributed state, alongside effective caching), and XML for integration.

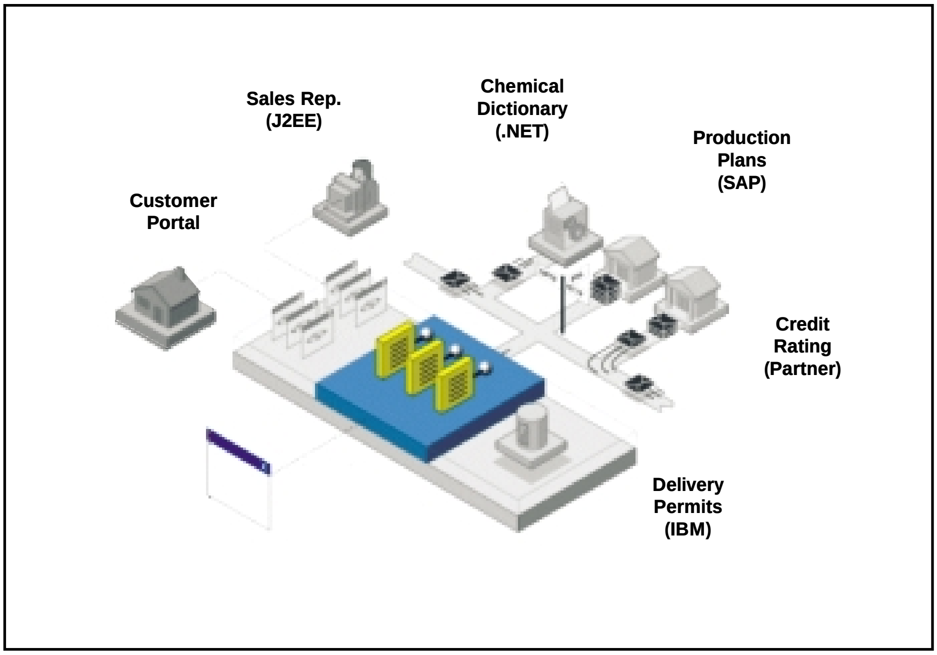


FIGURE 1 Composite business flow example

The current phase of the evolving Web services stack is orchestration; i.e. coordinating interactions among published Web services and composing them into long-running flows. Orchestration is comprised of three pillars: asynchronous conversations, flow coordination, and exception management. In support of these pillars (explained next), and building on its foundation, the Web services stack adds:

- ***WS-ReliableMessaging:*** To guarantee once and only-once delivery of messages
- ***WS-Addressing:*** To define correlation semantics to properly match requests and replies in the context of asynchronous messaging; compensation semantics for undoing of actions in the case of faults, as commanded by application logic
- ***BPEL4WS:*** An execution language for defining service composition and coordinating interactions into business flows

Pillar 1: Asynchronous Conversations

Support for asynchrony is essential for enabling "business quality" Web services that need to take part in integration scenarios. Asynchrony is also mandatory for allowing optimal use of "business time" (e.g., allowing for user intervention within the course of an executing business flow or deferred batch processing for better distribution of processing load). Asynchrony improves scalability by decoupling requests for service from their corresponding responses, thereby avoiding a cascade of execution bottlenecks from spreading throughout the application. Asynchrony also enables uninterrupted execution when services are temporarily unavailable and when clients are disconnected or offline.

The basic interaction pattern involves a business flow initiating a request to an asynchronous Web service. The Web service can take an arbitrarily long time to complete its execution upon receiving a request. Once execution is completed, the service replies to the sender of the request. In order to carry out this interaction, certain information needs to be exchanged between the sender (business flow) and the receiver (async service). The sender needs to provide the receiver with a reply-to address (so that the receiver knows where to send the reply, i.e. invoke a callback once ready).

The sender also needs to provide the receiver with a correlation ID. Post execution, the correlation ID is sent from the receiver to the sender so that the reply from the receiver can be associated with the correct originating request on the sender side. This basic pattern can be extended to a multistep interaction when the business flow needs to invoke the service through more than one entry point or vice versa.

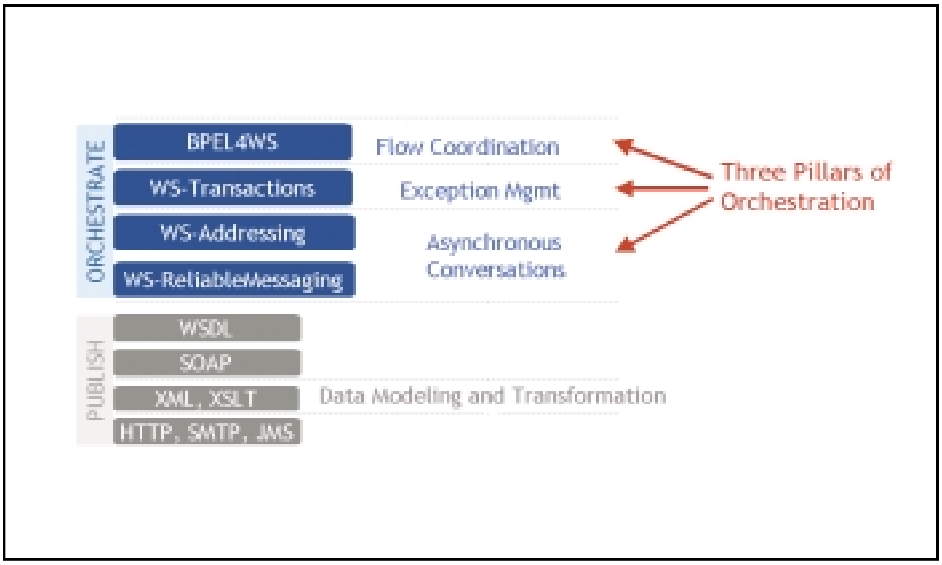


FIGURE 2 **Web services orchestration stack**

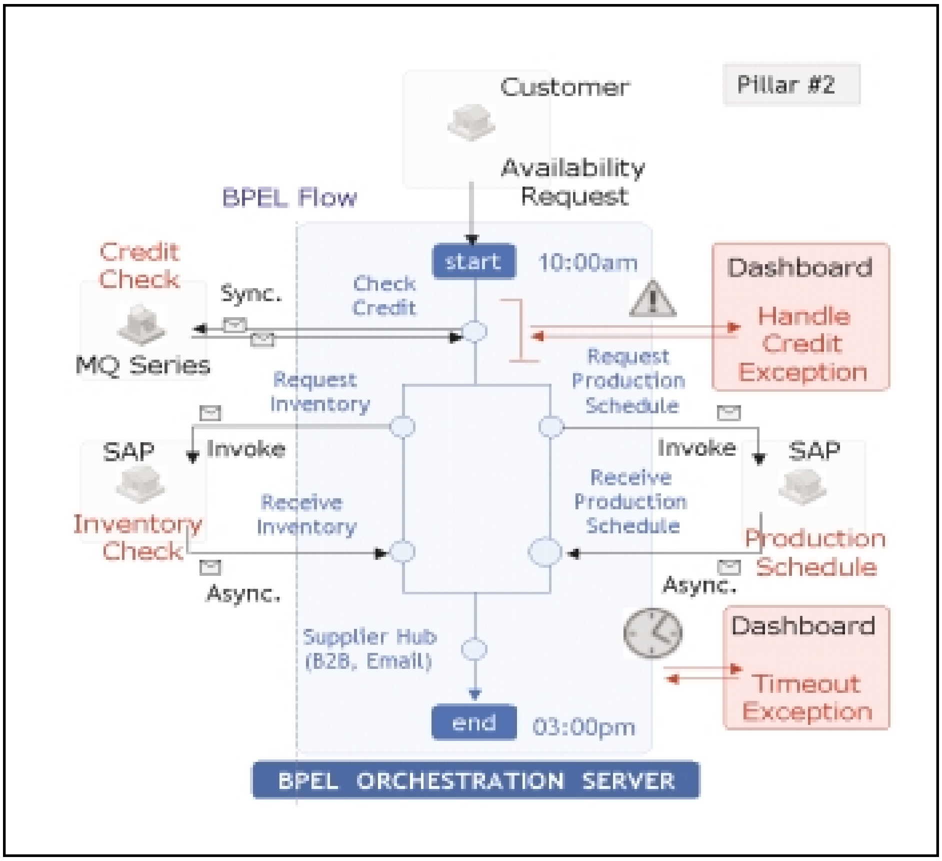


FIGURE 3 **Flow coordination**

The formalism for asynchronous conversations includes WS-Addressing, WS-ReliableMessaging, and BPEL Service Link. WS-Addressing specifies correlation and callback information. WS-Addressing provides transport-neutral mechanisms for addressing Web services and messages by defining XML elements to identify Web service endpoints and to secure end-to-end endpoint identification in transmitted messages.

WS-ReliableMessaging allows messages to be delivered reliably between interacting Web services in the presence of software component, system, or network failures. Its primary goal is to create a modular mechanism for reliable message delivery. BPEL Service Link defines the callback interface.

Pillar 2: Flow Coordination

Business flows in real-world application scenarios may include rather complex interaction patterns (see Figure 3). The ATP process involves the chemical manufacturing company, its applications, and related service providers. The interactions between the business flow and the services include sequencing of activities as well as running such sequences in parallel and joining their execution using a join pattern. The business flow includes both synchronous as well as asynchronous invocations of services and demonstrates exception handling during its long-running execution. (See Listing 1 for the BPEL code of this example; this listing is online at www.sys-con.com/webservices/source.cfm).

Flow coordination is comprised of the WSDL interface, XML variables, partners, flow activities, and compensation handlers. BPEL4WS relies heavily on WSDL descriptions of the services involved in order to refer to exchanged messages, the operations being invoked, and the portTypes they belong to.

The ATP process uses a WSDL interface to integrate, for example, into a customer portal. XML variables are containers for data exchanged between the interacting services. Partners are external services

which participate in the business process. Flow activities are basic and structured commands that define the rules and sequences for interacting with each service. Flow activities will be explained later. Compensation handlers are rules for undoing parts for the business flow in case of exceptions.

The ATP process includes XML variables such as the credit rating, production schedule, and inventory status. These variables are used for exchanging data with the partners (external services), which include a credit check mainframe application fronted by MQSeries, a production plans SAP application, and an inventory check SAP application. The XML variables are used for holding the data used for requests sent to a partner and for replies received from a partner. They are also used to store intermediate information, such as the combined planning information sent to the supplier hub.

Compensation handlers are used within a business process, where it might be necessary to undo one of the steps that have already been successfully completed. Compensation handlers define these undo steps at the level of a predefined scope. A compensation handler contains one activity which is run when a scope needs to be compensated. Since activities can share XML variables, completing scopes with associated compensation handlers need to save a snapshot of the data when the scope is completed. This data is used by the handler in case it is invoked (e.g., upon encountering an exception at some later point in time). Compensation handlers become ready to run once the scope they're associated with is completed successfully.

BPEL Activities

BPEL4WS is divided into basic and structured activities. Basic activities are the simplest form of interaction with the world outside the business flow. They are not sequenced and comprise individual steps to interact with a service, manipulate the exchanged data, or handle exceptions encountered during execution.

Structured activities define the order in which activities execute. They describe how a business process is created by composing the basic activities it performs into structures. These structures entail the control patterns, data flow, fault handling, external event handling, and coordination of message exchanges between process instances.

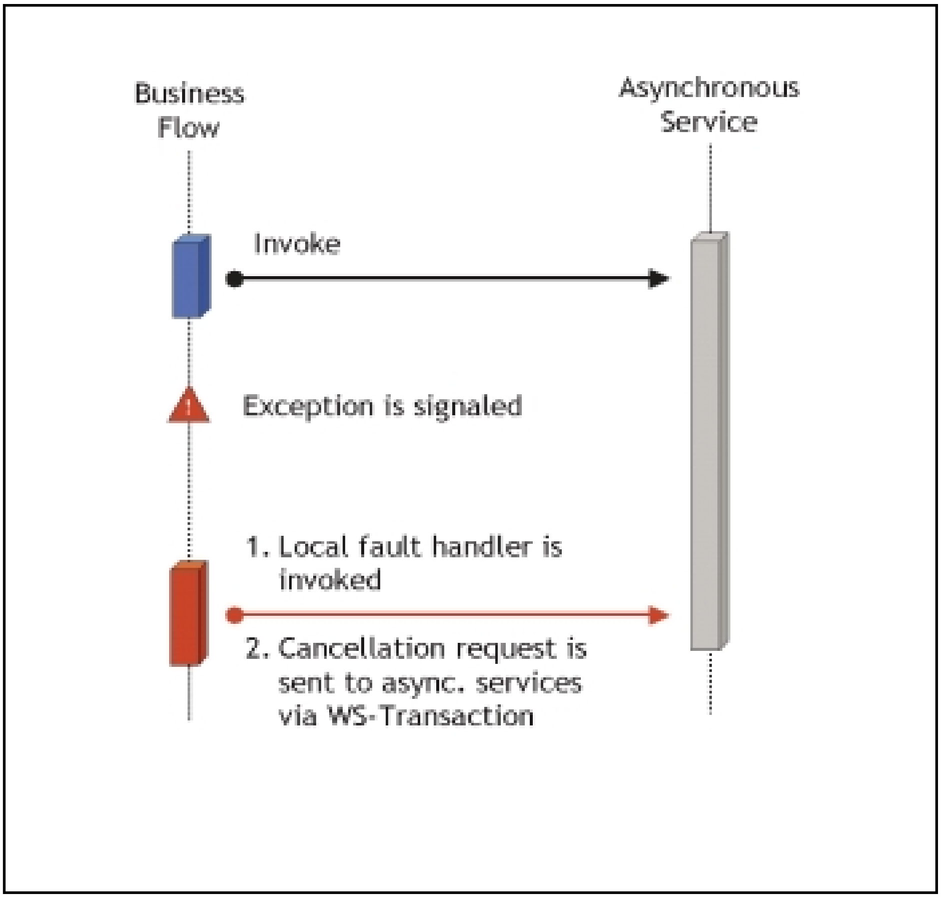

Business Flow
Asynchronous Service
Invoke
Exception is signaled
1. Local fault handler is invoked
2. Cancellation request is sent to async. services via WS-Transaction

FIGURE 4 Exception management

The control flow is a hybrid model halway between block structured and state transitions. It uses links to establish dependencies between block definitions and put their own constraints on how activities are set to run. Links are associated with conditions based on XPath expressions that evaluate to true or false for determining if an activity is to run or terminate (a join condition). It does not have process composition (i.e., a sub-process) but does have service composition (through use of the WSDL interface of a BPEL flow).

BPEL will drive the creation of a new generation of assembly tools that allow enterprises to implement coarse-grain processes by visually composing and tying activities and service interactions into collaborative and transactional business flows. The key benefit is developer productivity for creation of business flows but also the ability to quickly adapt to change, an essential requirement in most real-world implementations of organizations' business flows.

Pillar #3: Exception Management

According to some analysts, nearly 80% of the programming effort in automating business processes is spent in exception management. Therefore, exception management should be treated as an integral part of any specification for Web service orchestration. In contrast, most workflow and EAI/BPM solutions on the market today have implemented exception management as an afterthought. The notion of business transactions is helpful in the context of simplifying exception management across a network of loosely coupled asynchronous Web services.

The basic interaction pattern (see Figure 4) is similar to the one described for asynchronous conversations (i.e., a business flow initiating a request to an asynchronous Web service). The use case here involves an exception signaled during the business flow, before the service completes its execution.

In BPEL, a local fault handler associated with the Web service, is invoked subsequent to the signaled exception. In addition, the asynchronous service is notified of the exception through a cancellation request using WS-Transaction.

> "The core element of this infrastructure is a BPEL server that implements the various layers of the Web service orchestration standards stack"

The formalism for exception management includes the use of fault handlers and compensation handlers. Fault handlers are used when exceptions are signaled during the execution of an invoked Web service. Compensation handlers, on the other hand, reverse or undo the effect of an invocation of a Web service that completed its execution.

Exception management also includes WS-Coordination and WS-Transaction. These specifications provide cancellation requests across a network of services to ensure coordination of interacting services in case of failures, as well as guaranteeing the integrity of the overall execution.

BPEL Infrastructure

The three pillars combine to form the foundation of end-to-end business processes that call for a standards-based infrastructure for deploying and managing composite business flows (see Figure 5). The core element of this infrastructure is a BPEL server that implements the various layers of the Web service orchestration standards stack and executes flow logic that is portable and can execute in any compliant BPEL server.

The BPEL server supports orchestration logic provided in BPEL form for execution of business flows. Although not a mandatory requirement, a BPEL server runtime can utilize a J2EE application server for its underlying execution environment (rather than reinventing the wheel with respect to multithreading, database connection pooling, etc.). It also provides native J2EE integration by leveraging the J2EE application server runtime environment. To ensure reliability of long-running business flows involving asynchronous conversations with Web services and loosely-coupled business transactions, it uses context dehydration for executing flows. The dehydration mechanism uses a persistent store, such as a relational database, to safely store, and subsequently retrieve, flow instances.

The second important element of the infrastructure is the BPEL console, which is designed to provide ease of management for business flows deployed to the BPEL server. It offers essential monitoring and administration facilities to developers as well as operational and business users. These facilities include testing and debugging of business flows, audit trail of flow execution in both textual and graphical form, reporting and statistics, and versioning. The last capability is particularly important for long-running flows since multiple versions of the same business flow must run in parallel at times (e.g., when a new version phases in and an old version phases out).

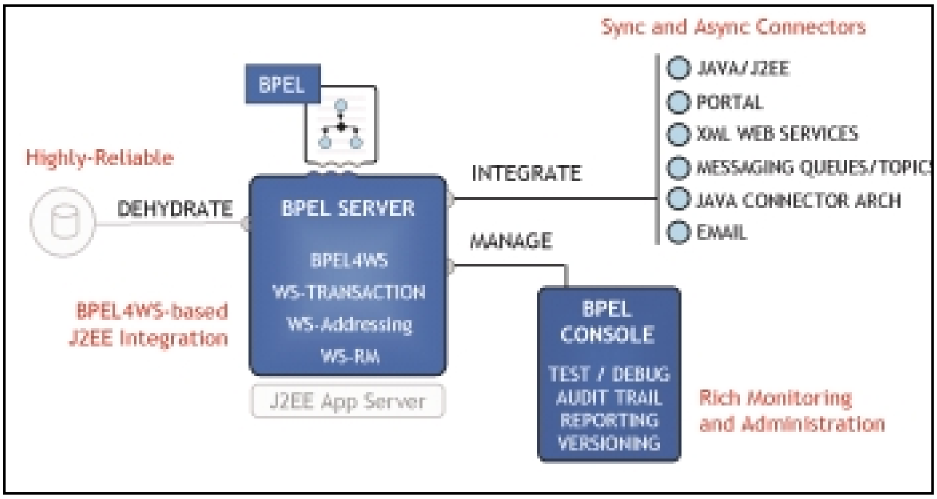


FIGURE 5 BPEL infrastructure

> "This new genre of process-driven applications moves enterprises closer to the realization of the agile real-time enterprise"

The BPEL infrastructure should be equipped with strong support for integration. Integration capabilities are conveniently expressed through connectors that plug into the BPEL server. These connectors should support both synchronous and asynchronous interactions. Common connectors needed for enterprise computing environments include XML Web services, Java/J2EE, .NET, JMS, e-mail, and portal infrastructure (for user interaction). Such connectors should be transparent to flow logic developers and have no effect upon the portability of the BPEL code.

Conclusion

BPEL, a standard process flow grammar, provides a new foundation for integration. It empowers developers to tie transactional services, events, and user tasks into easy-to-adapt business flows. This new genre of process-driven applications moves enterprises closer to the realization of the agile real-time enterprise

Written by David Cameron

BPM to the Rescue

A solution to a process problem

In one day, a CIO receives two dreaded e-mails from the CEO, who is acting under board pressure to change the company's fortunes. The first "asks" for a revised IT budget reflecting a 15% cut for the rest of the year. The second calls for new ways of using IT resources to capitalize on missed customer revenue opportunities.

AUTHOR BIO:

David Cameron is vice president of marketing and product integration for AptSoft Corporation (http://www.aptsoft.com.), developers of software that acts as a real-time enterprise catalyst companies can use to continuously identify, evaluate, and respond to key business events. He manages the marketing and product integration efforts, which include working directly with prospects and clients to better understand the applications and value of AptSoft's enterprise software.

DAVID.CAMERON@APTSOFT.COM

Forgive the CIO for feeling trapped. Suddenly the company's future seems to hinge on two fundamentally conflicting business objectives: cutting costs and increasing responsiveness to cross-selling, customer retention, and other revenue opportunities. The trap for CIOs is that they must do more with the technology already in place with less money to acquire the capability to improve processes. They have yet to recoup the costs of their IT investments, so they can't purchase new technologies to support new initiatives. CIOs must bleed more work out of their existing hardware and software investments to support the increased agility that their line-of-business counterparts require. But how?

CIOs often make the mistake of viewing this as an IT problem first. In fact, it's primarily a process problem. They need to identify their business objectives and determine how best to reconfigure their core processes to support those objectives. Instead of dwelling on the IT plumbing, they need to look at how their companies react to important business events, such as a high-value customer withdrawing half of his investment account. What business processes do they have in place to keep and win more revenue from that customer?

> "Once they've determined their business objectives, CIOs should look to business process management (BPM) technology to make sure their IT investments work together to achieve those objectives"

Once they've determined their business objectives, CIOs should look to business process management (BPM) technology to make sure their IT investments work together to achieve those objectives. BPM relies on Web services to enable new processes that respond quickly to key business events as well as tap little-used business process functionality trapped within existing applications.

Facing the Integration Challenge

Traditionally, CIOs have had two ways to support the kind of requirements described above. First, they could write a lot of custom program code. However, because code is hard to change, they first must poll users to determine exactly what they want to do as well as document and design the code. This type of approach is very costly, and therefore requires an up-front detailed return on investment (ROI) study with a 12-month payback schedule to justify the investment. Most of the cost is due to two factors:

- IT staff must spend time assessing business process requirements and ensuring the code addresses those requirements because it's so expensive to change the code after the fact.
- IT staff must be ready at all times to troubleshoot, modify, and upgrade.

Web services can dramatically reduce the cost associated with these items be-

cause of the flexibility implied in the architecture. Since business rules can be more easily changed after deployment, fewer demands are placed on the up-front design phase and less "user-acceptance" risk exists.

The second approach is the data integration approach. By creating a common database, users and applications can share information and coordinate activity. This, too, is very expensive. It requires complex requirement and architecture efforts, as well as ongoing maintenance that involves refreshing data. Data integration is expensive for several reasons:

- Much of the cost is for the infrastructure to move and store data for potential (unknown actual use) future use by applications. That means most of the data is there for insurance purposes, and probably 20% of the data is accessed 80% of the time.
- Modifying business processes usually requires architectural rework, which drives up maintenance costs.
- The level of granularity of the data (i.e., do we need transaction-level data or just summarized data of monthly transactions?), and the latency of data (how often do I need to refresh it?) drive key cost decisions – decisions typically made data element by data element.
- The cost of having IT staff hide the complexity of the physical data's appearance from the end user who can't understand table layouts also plays a role.

A Web services architecture facilitates the creation of a layer of abstraction, enabling reusable components to be created that hide the complexity of the data from the user and allowing easier modification and enhancement over time.

It is not surprising, then, that when faced with requirements to be more agile (i.e., view change as necessary and good instead of costly and undesirable), the impracticality of these approaches becomes painfully apparent.

Many end users, analysts, and vendors have reached the same conclusion: a new approach is necessary for the types of requirements emerging today in support of business imperatives. Web services that enable siloed IT systems to freely communicate with one another are the foundation of that approach.

A New Approach Takes Shape

For a company to revise its business processes, it must first take stock of its existing systems and methodologies before determining how to use them to achieve certain objectives. A premier financial services company successfully planned how to fine-tune its business processes in the following manner.

> "Web services can dramatically reduce the cost associated with these items because of the flexibility implied in the architecture"

First, it identified strategic business objectives such as increased per-customer revenue and increased competitive differentiation. Then it determined the key business drivers of these objectives, such as meeting increasingly high service expectations, unifying the online and offline channels, and coordinating contact management. Finally, it parsed out the key business processes behind the business drivers, such as making a product offer and responding to a query for product information. Having defined its objectives and the means to achieve them, the firm was set to tackle the technical challenge of how to make it happen.

This process-centric approach to fine-tuning business processes has several benefits. It allows companies to:

- Tap existing infrastructure resources
- Benchmark current state and measure success
- Readily access and manage the "knobs and levers" that control these processes

Tactically, improving the efficiency of business processes involves three areas of focus:

- The "seams" where processes are handed off from system to system or department to department, such as when a loan application filled out on the Web site is passed to the underwriting department
- The areas where companies can modify customer behavior to support business objectives, such as encouraging "self-service" on the Web site, but implementing "chat" to support customers who experience problems
- The manner in which companies differentiate the customer experience, such as treating multirelationship customers differently from single-relationship customers

Addressing these three tactical areas almost always means tighter systems integration, either from system-to-system or system-to-database. It also means connecting "silos" of business logic that operate independently of each other and of any "customer context," or knowledge about the customer that will improve interaction. Until recently, it took custom coding to connect the silos. Custom-coded solutions are expensive, brittle, and often impractical when management demands that the company respond quickly to business drivers. As a result, these types of integrations often fail, or companies don't even attempt them.

Bridging the Gap with BPM

Enter BPM, which industry analysts believe will jump from $570 million in 2002 to more than $1.5 billion in 2005.

BPM coordinates the actions of isolated IT systems — like CRM, SFA, and ERP. Using the built-in intelligence of existing applications and systems, BPM can help companies nimbly react to key business events and capture millions of dollars in potentially missed revenue opportunities while improving customer service. Using a Web services architecture founded on XML, SOAP, J2EE, and similar technologies, BPM can synchronize signals from existing applications, enabling them to work together without changing how they operate individually.

BPM technology overcomes the weaknesses of the current generation of systems integration approaches. It provides a variable cost model that allows for a small initial implementation footprint tied to a three-to-six-month payback period, tightly coupling investment and return. Today, technology must prove utility before vendors can demand license and support fees. Instead of the typical 12+-month implementation and two-year payback model, Web services– based BPM technology scales deployment effort to the initial requirements. As a company extends its BPM implementation over time, it builds upon the work already performed.

> “BPM technology overcomes the weaknesses of the current generation of systems integration approaches”

BPM allows for rapid modification so end users can engage in “test-and-learn” process development and management. Most of the processes that support key business drivers are complex and must continuously evolve in response to internal and external changes. BPM technology that utilizes a Web services architecture enables users to implement, test, modify, and reimplement processes in rapid sequence based on what they’ve learned.

BPM gives end users more involvement so they can change processes on the fly and improve agility and productivity without having to call IT. The current generation of technologies is brittle largely because IT shoulders the bulk of maintenance costs and responsibilities. BPM technology balances the workload between the end user and IT in such a way that users can maintain processes without resorting to IT support.

Companies benefit from BPM’s ability to reuse common data objects such as “Customer” and “Trade” and use open standards to make them available to any application. The complex infrastructure many organizations operate has evolved from the many proprietary database structures and application logic syntaxes from years of systems development. Most valuable components of business processes, including data definitions, business rules, and transformation logic, are replicated in a variety of formats across the enterprise. BPM technology relies on an object model that exposes and abstracts these elements so they may be reused across different systems via components of the Web services architecture, such as emerging XML standards.

BPM provides a context for the business process to allow users to access information at the application level, before it has been saved to a database, dramatically reducing data integration. Most traditional application integration is based upon the movement of “state data,” that is, data that has been saved about a particular event. State data is stored in databases, and then “synchronized” with other databases linked to other applications. But companies don’t save much information needed to support business rules because it is too expensive or too complex.

Companies can use BPM to cultivate existing business logic and integration capabilities to connect existing applications and databases without modifying them. This lightweight connection is the hallmark of Web services. Most of these systems already have connections built into them, either with middleware or published application programming interfaces (APIs). BPM technology uses these connections to link databases and applications while providing a layer of abstraction that hides the variability that exists between systems.

Integration technology has been around for nearly 10 years. However, while it has addressed some specific needs, it still isn’t practical for most companies because of the cost and performance issues. Consider that most integration projects still involve custom code, according to Forrester Research. Businesses that want to streamline how they respond to key events, build revenue, and strengthen customer relationships should instead take a fresh look at integration from the process perspective. What processes do they need to implement to achieve their business goals, and what technology will help them do that affordably? With its roots in Web services that make disparate IT systems communicate, BPM technology can provide the framework CIOs need to capitalize on missed revenue opportunities while still keeping costs under control.

WebServices Journal
.NET J2EE XML

PRESIDENT AND CEO
Fuat A. Kircaali fuat@sys-con.com

VP, BUSINESS DEVELOPMENT
Grisha Davida grisha@sys-con.com

ADVERTISING
SENIOR VP, SALES & MARKETING
Carmen Gonzalez carmen@sys-con.com
VP, SALES & MARKETING
Miles Silverman miles@sys-con.com
ADVERTISING DIRECTOR
Robyn Forma robyn@sys-con.com
DIRECTOR, SALES & MARKETING
Megan Ring-Mussa megan@sys-con.com
ADVERTISING SALES MANAGER
Alisa Catalano alisa@sys-con.com
ASSOCIATE SALES MANAGERS
Carrie Gebert carrie@sys-con.com
Kristin Kuhnle kristin@sys-con.com

SYS-CON EVENTS
PRESIDENT, SYS-CON EVENTS
Grisha Davida grisha@sys-con.com
CONFERENCE MANAGER
Michael Lynch mike@sys-con.com

CUSTOMER RELATIONS / JDJ STORE
CIRCULATION SERVICE COORDINATORS
Niki Panagopoulos niki@sys-con.com
Shelia Dickerson shelia@sys-con.com
JDJ STORE MANAGER
Rachel McGouran rachel@sys-con.com

SYS-CON.COM
VP, INFORMATION SYSTEMS
Robert Diamond robert@sys-con.com
WEB DESIGNERS
Stephen Kilmurray stephen@sys-con.com
Christopher Croce chris@sys-con.com
ONLINE EDITOR
Lin Goetz lin@sys-con.com

ACCOUNTING
FINANCIAL ANALYST
Joan LaRose joan@sys-con.com
ACCOUNTS RECEIVABLE
Kerri Von Achen kerri@sys-con.com
ACCOUNTS PAYABLE
Betty White betty@sys-con.com

SUBSCRIPTIONS
SUBSCRIBE@SYS-CON.COM
1-888-303-5282
FOR SUBSCRIPTIONS AND REQUESTS FOR BULK ORDERS, PLEASE SEND YOUR LETTERS TO SUBSCRIPTION DEPARTMENT
COVER PRICE: $6.99/ISSUE
DOMESTIC: $69.99/YR (12 ISSUES)
CANADA/MEXICO: $89.99/YR
ALL OTHER COUNTRIES: $99.99/YR
(U.S. BANKS OR MONEY ORDERS)

WorldWide Newsstand Distribution
Curtis Circulation Company, New Milford, NJ

SYS-CON MEDIA

Written by Derick Townsend, Ryan Cairns, & Christoph Schittko

Using BPEL

What IT managers need to know

Web services technology is rapidly evolving to meet the complex needs of the enterprise customer. The ability to integrate and assemble individual Web services into standards-based business processes is an important element of the service-oriented enterprise and the overall Web service technology "stack."

These loosely coupled business processes, commonly referred to as orchestrated Web services, will be designed, integrated, executed, and managed similar to how proprietary enterprise application integration (EAI) and Business Process Management (BPM) tools operate today. However, business process execution standards and Web services will greatly reduce vendor lock-in to dramatically reduce costs and provide broader interoperability benefits.

The Role of Business Process Execution Language

To address these needs, the Business Process Execution Language for Web services (BPEL4WS or BPEL) has quickly become the dominant specification to standardize integration logic and process automation between Web services. BPEL was jointly created by IBM, BEA, and Microsoft in August 2002. This inception by a coalition of the most influential vendors in the industry practically assured its widespread adoption from the beginning. In April 2003, BPEL was submitted to OASIS to obtain even broader industry acceptance and open standardization. Today, many industry analysts have proclaimed that BPEL is the undisputed standard.

> "Today, many industry analysts have proclaimed that BPEL is the undisputed standard"

The BPEL specification defines the syntax and semantics of the BPEL language, which contains a variety of process flow constructs. You can perform conditional branching, parallel process flows, nested sub-processes, process joins, and other related features. In fact, BPEL represents a convergence of language features from IBM's Web Service Flow Language (WSFL) and Microsoft's XLANG, which is the orchestration language used by Microsoft's BizTalk server. Both WSFL and XLANG have been superseded by the BPEL specification.

Like all the other languages in the Web services arena, BPEL is defined in an XML format. It also leverages other Web service standards such as WSDL to describe available interfaces. BPEL describes the inbound and outbound process interfaces in WSDL so that they can be easily integrated into other processes or applications. This allows consumers of a process to inspect and invoke a BPEL process just like any other Web service.

Just as today's software development tools include Web services in their development capabilities, easy-to-use tools are arriving to design business processes and produce BPEL scripts. If your organization has the capability to integrate Web services, then you will also be able to create and invoke BPEL processes by leveraging existing Web services infrastructure and know-how. This will ultimately enable a broader group of developers to perform business integration and process automation tasks that previously required highly specialized skills.

Why Do I Need BPEL?

Traditional methods for integration and process automation typically involve embedded logic inside of functionally oriented IT applications, such as ERP, supply chain, or CRM. The develop-

Author Bio

Derick Townsend is chief operating officer of OpenStorm Software, a provider of Web service–based integration products. He has over 15 years of experience in software and related technology industries. Derick holds an MBA from the University of Texas at Austin and a BS in mechanical engineering from the University of Arizona.
DT@OPENSTORM.COM

Ryan Cairns is chief technology officer of OpenStorm Software. He has over eight years' experience in software development for companies such as Tivoli, Trilogy, and Security Broadband. Ryan holds a BS in computer science from the University of Texas at Austin.
RYAN@OPENSTORM.COM.

Christoph Schittko is a principal consultant at Momentum Software. He has over 10 years' experience in software development and is a recognized expert on the Microsoft .NET platform. Christoph holds an advanced degree in electrical engineering from the Friedrich-Alexander University Erlangen- Nürnberg in Erlangen, Germany and received a Microsoft MVP award for contributions to various .NET developer communities.
SCHITTKO@MOMENTUM SOFTWARE.COM

Crystal Decisions

www.crystaldecisions.com/cr9/220

ment, testing, and deployment efforts required to change these applications make integration and process changes both costly and complex. These limitations still inhibit organizational flexibility today.

To address these issues, proprietary EAI and BPM products emerged to abstract integration and process automation into a new layer of software tools. These software products liberated integration and process tasks from the underlying functional IT applications so they could be more effectively changed, managed, and optimized.

BPEL and Web services now provide a standardized integration interface and a standardized language for integration and process automation. BPEL, in effect, has the potential to commoditize the capabilities provided by proprietary EAI and BPM solutions. As often occurs in a commodity market, the resulting prices for products and services are certain to fall.

> “BPEL has the potential to significantly disrupt established EAI and BPM vendors and their markets”

What Will Drive Adoption?

The most important case for BPEL is that proprietary EAI and BPM solutions are just too expensive. They are expensive to develop, maintain, and extend across a diverse, heterogeneous environment. Proprietary integration links are often brittle, and the cost to maintain them as organizations continually evolve is a significant burden. The specialized skills required to support these proprietary solutions often create their own cost and availability concerns. The frequent result is that constrained IT budgets end up shifting the majority of their funds toward maintenance issues, with precious little left over to satisfy the needs for innovation and improved flexibility.

BPEL and Web services are technologies with the potential to finally break through this impasse. Web services provide a ubiquitous, standards-based interface that can be readily accessible from inside or outside the corporate firewall. Web service standards for service discovery (UDDI) and self-description (WSDL, WSIL) actively promote and encourage rapid integration and service reuse. With the advent of next– generation BPEL-compliant development tools, the expensive development cycles of the past are replaced with low-cost integration and process changes, enabling a new level of organizational agility through orchestrated Web services. As long as Web service interfaces are available for the target applications and systems, BPEL will provide benefits that proprietary integration solutions will be hard-pressed to beat.

How Will BPEL Be Used?

Within the corporate firewall, BPEL has the potential to standardize application-to-application integration and extend integration to previously isolated systems. As a result of years of proprietary integration efforts, a variety of integration tools and solutions exist in the enterprise today. This remains true in organizations that adopt high-end EAI products, as the cost-benefit analysis of some integration needs cannot justify the use of custom EAI adapters. In contrast, BPEL holds promise as a “lowest common denominator” integration technology that delivers a ubiquitous, platform-neutral solution for lower cost.

Outside the firewall, BPEL can enable a whole new level of corporate agility as it relates to integrating and switching external vendors and services. By using BPEL to define business processes, companies are empowered to select best-of-breed processes and services to incorporate into their operations. This provides flexibility to replace or upgrade certain aspects of a business process without impacting the systems that are working well. For instance, a company can change their warehouse service provider without impacting their order management system, even though both may be participants in several business processes.

When Should I Begin Deploying BPEL?

BPEL remains an emerging technology, with challenges awaiting those interested in near-term deployment. Fortunately, the initial vacuum of BPEL-based development tools has been filled. Many software vendors have recognized the considerable market opportunity and responded quickly with solutions. Vendors like IBM, Collaxa, and OpenStorm offer BPEL-compliant orchestration engines, and a variety of design and development tools have been announced by industry leaders such as Microsoft and BEA.

Regarding specific deployments, BPEL makes sense for environments that already have many exposed Web service interfaces. The greater the number of Web services available, the more valuable BPEL will become. Fortunately, Web service pilot programs and integration efforts are one of the few areas of IT spending that have actually increased during the economic downturn, and the number of publicly accessible Web services continues to grow rapidly.

Finally, BPEL has the potential to significantly disrupt established EAI and BPM vendors and their markets. As a result, established vendors are moving to incorporate BPEL compliance into their proprietary products, and new vendors are leveraging a window of opportunity to create new products and new product categories. If your organization is a current or future customer of EAI or BPM solutions, the time is right to begin BPEL pilot projects and become familiar with the technology. This knowledge will be valuable in evaluating future products, while also exploring ways to leverage BPEL to obtain competitive advantage within your industry.

Web Service Orchestration and Choreography

A look at WSCI and BPEL4WS

Written by Chris Peltz

Web services are rapidly emerging as the most practical approach for integrating a wide array of customer, vendor, and business-partner applications. While many companies have begun to deploy individual Web services, the real value will come when enterprises can connect services together, providing higher value to an organization.

Early experience shows that to make the most of new Web services investments there must be a standard approach to Web services composition.

AUTHOR BIO:

Chris Peltz is a senior software consultant in HP's Developer Resources Organization (devresource.hp.com), providing technical consulting on J2EE and Web services architectures. He brings over 10 years of software development experience in helping customers select technologies, tools, and platforms for building enterprise applications.

CHRIS.PELTZ@HP.COM

IT organizations need the agility to adapt to customer requirements and changing market conditions. But existing business process languages do not directly support Web services standards and, as a result, IT organizations may be tempted to take a short-term approach and create their own proprietary protocols for composing services together. Web services orchestration and choreography standards are efforts that can be long-term solutions for business connectivity. By connecting services through open, standards-based methods, organizations spare themselves the burden of maintaining these proprietary interfaces.

The two standards discussed here – the Web Service Choreography Interface (WSCI) and Business Process Execution Language for Web Services (BPEL4WS) – are designed to reduce the inherent complexity of connecting Web services together. Without them, an organization is left to build proprietary business protocols that shortchange true Web services collaboration. Recently, the terms orchestration and choreography have been employed to describe this collaboration:

- ***Orchestration:*** Refers to an executable business process that may interact with both internal and external Web services. Orchestration describes how Web services can interact at the message level, including the business logic and execution order of the interactions. These interactions may span applications and/or organizations, and result in a long-lived, transactional process. With orchestration, the process is always controlled from the perspective of one of the business parties.
- ***Choreography:*** More collaborative in nature, where each party involved in the process describes the part they play in the interaction. Choreography tracks the sequence of messages that may involve multiple parties and multiple sources. It is associated with the public message exchanges that occur between multiple Web services.

Orchestration differs from choreography in that it describes a process flow between services, controlled by a single party. More collaborative in nature (see Figure 1), choreography tracks the sequence of messages involving multiple parties, where no one party truly "owns" the conversation.

In this article, I'll highlight key technical requirements for Web services orchestration and choreography, and point out key standards used to meet these needs.

Technical Requirements for Orchestration and Choreography

Before introducing the standards, it's important to define the technical requirements for orchestrating Web services. The following requirements are important for both the language and the underlying infrastructure that supports it:

- ***Flexibility:*** One of the most important considerations is the flexibility offered by the language. Flexibility can be achieved by providing a clear separation between the process logic and the Web services invoked. This separation can usually be achieved through an orchestration engine that handles the overall process flow. With this flexibility, an organization can easily swap out services as business needs change.
- ***Basic and structured activities:*** An orchestration language must support activities for both communicating with other Web services and handling workflow semantics. One can think of a basic activity as a component that interacts with something external to the process itself. In contrast, structured activities manage the overall process flow, specifying what activities should run and in what order.
- ***Recursive composition:*** A single business process can interact with multiple Web services. However, a business process can itself be exposed as a Web service, enabling business processes to be aggregated to form higher-level processes.

In addition, both Web services orchestration and choreography must support some basic requirements for managing the overall integrity and consistency of the interactions. These requirements include:

- ***Persistence and correlation:*** The ability to maintain state across Web services requests is an important requirement, especially when dealing with asynchronous Web services. The language and infrastructure should provide a mechanism to manage data persistence and correlate requests in order to build higher-level conversations.
- ***Exception handling and transactions:*** Orchestrated Web services that are long-running must also manage exceptions and transactional integrity. For example, resources cannot be locked in a transaction that runs over a long period of time.

WSCI

WSCI defines an extension to WSDL for Web services collaboration. Initially authored by Sun, SAP, BEA, and Intalio, it was recently published as a W3C note. WSCI is a choreography language that describes the messages exchanged between Web services that participate in a collaborative exchange. A key aspect of WSCI is that it describes only the observable behavior between Web services. It does not address the definition of an executable business process.

A single WSCI interface describes only one partner's participation in a message exchange. As Figure 2 illustrates, a WSCI choreography would include a set of WSCI interfaces, one for each partner in the interaction. In WSCI, there is no single controlling process managing the interaction.

WSCI can be viewed as a layer on top of the existing Web services stack. Each action in WSCI represents a unit of work, which typically would map to a specific WSDL operation. WSCI can be thought of as an extension to WSDL, describing how the operations can be choreographed. In other words, WSDL describes the entry points for each service, while WSCI would describe the interactions among these WSDL operations.

WSCI defines an <action> tag for specifying a basic request or response message. Each activity specifies the WSDL operation involved and the role being played by the participant. External services can then be invoked through the <call> tag. A wide variety of structured activities are supported, including sequential and parallel processing and condition looping. WSCI also introduces an <all> activity, used to indicate that the specific actions have to be performed, but not in any particular order.

Listing 1 is a simple example of WSCI. In this example, a purchasing interface is created containing two activities, "Receive Order" and "Confirm". Note that this is the WSCI document from the perspective of the agent. There would also be a WSCI interface for the buyer and the supplier in the interaction.

BPEL4WS

The BPEL4WS standard represents a convergence of ideas originally proposed by two early workflow languages, XLANG and WSFL. Microsoft, IBM, Siebel Systems, BEA, and SAP authored the 1.1 release of the specification in May 2003. It provides an XML-based grammar for describing the control logic required to coordinate Web services participating in a process flow and is layered on top of WSDL, with BPEL4WS defining how the WSDL operations should be sequenced.

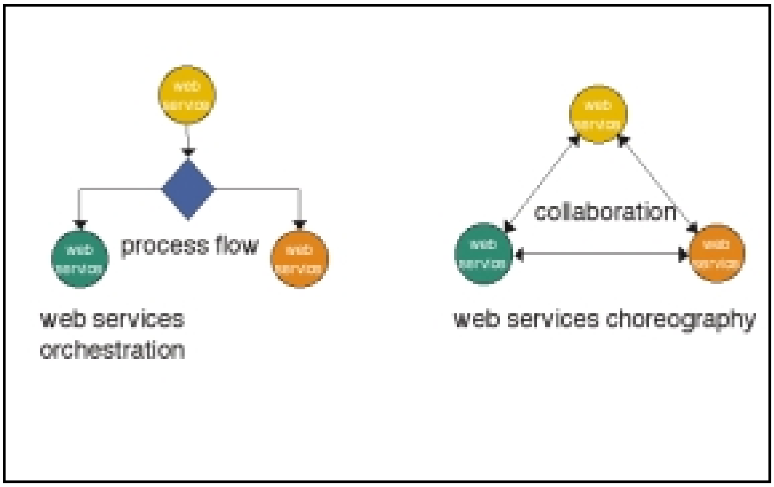


FIGURE 1 Orchestration and choreography

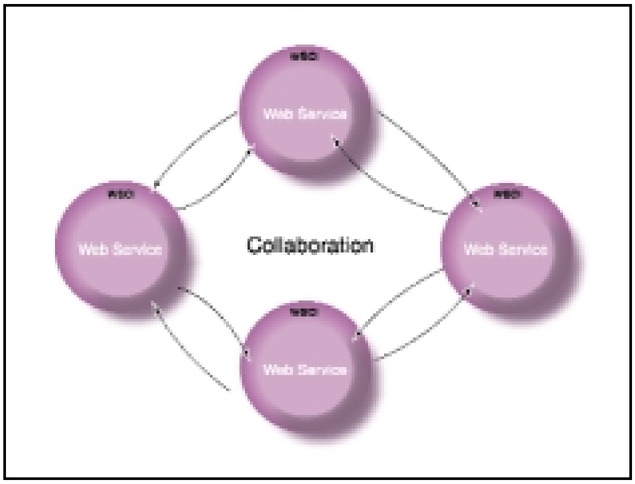


FIGURE 2 Web Services Choreography Interface (WSCI)

BPEL4WS provides support for both abstract business protocols and executable business processes. A BPEL4WS business protocol specifies the public message exchanges between parties. Business protocols are not executable and do not convey the internal details of a process flow, similar to WSCI. An executable process models the behavior of participants in a specific business interaction, essentially modeling a private workflow. Executable processes provide the orchestration support described earlier, while the business protocols focus more on Web services choreography.

The BPEL4WS specification supports basic activities for communicating with Web services. The typical scenario is that there is a message received into a BPEL4WS executable process. The process may then invoke a series of external services to gather additional data, and then respond back to the requestor. In Figure 3, the <receive>, <reply>, and <invoke> messages all represent basic activities for connecting the services together.

BPEL4WS also supports structured activities for constructing the business logic for a process. These activities include sequential and parallel activities, as well support for conditional looping and dynamic branching. Listing 2 is a simple illustration of how a sequential activity would be described.

Variables and partners are two other important elements within BPEL4WS that satisfy the requirements for persistence and correlation.

- ***Variable:*** Identifies the specific data exchanged in a message flow, which typically maps to a WSDL message type or XML schema type. When a BPEL4WS process receives a message, the appropriate variable is populated so that subsequent requests can access the data.
- ***Partner:*** Defines the various parties that interact with the process.

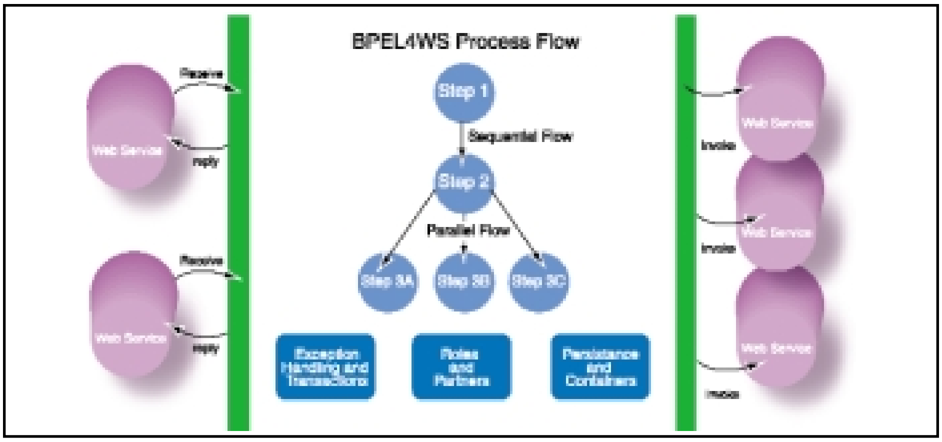


FIGURE 3 BPEL4WS Process Flow

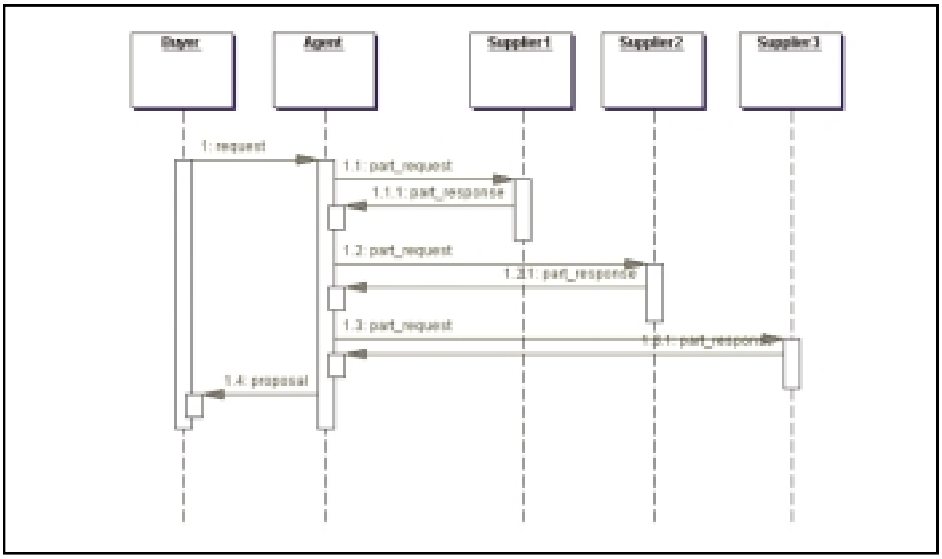


FIGURE 4 Case study sequence diagram

Comparing WSCI and BPEL4WS

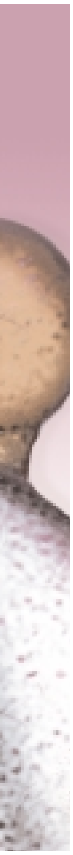
Each standard takes a somewhat different approach to orchestration and choreography. While BPEL4WS supports the notion of "abstract processes," most of its focus is aimed at BPEL4WS executable processes. BPEL4WS takes more of an "inside-out" perspective, describing an executable process from the perspective of one of the partners. WSCI takes more of a collaborative and choreographed approach, requiring each participant in the message exchange to define a WSCI interface.

At the same time, WSCI and BPEL4WS both meet many of the technical requirements outlined earlier. They both provide strong support for persistence and correlation to manage conversations. WSCI and BPEL4WS also describe how exceptions and transactions should be managed. From a usability standpoint, WSCI does have a somewhat "cleaner" interface than BPEL4WS. Some of the difficulties in using BPEL4WS are attributed to the fact that the language includes artifacts from both XLANG and WSFL, each of which took a different approach to workflow.

It's also important to look at overall industry acceptance for each standard. BPEL4WS has a number of major supporters behind it, including IBM, Microsoft, and BEA. Moreover, the companies submitted BPEL4WS to OASIS in April 2003, further broadening its support. Sun, Intalio, and SAP initially submitted the WSCI specification to the W3C, which recently created a WS-Choreography working group to standardize on Web services choreography. While the OASIS BPEL technical committee focuses on standardizing the BPEL4WS specification, WS-Choreography will be defining a choreography language.

The vendor specifications have quickly moved into a number of product implementations. Vendors such as Intalio and Vergil Technologies have products that implement BPML (Business Process Markup Language), which incorporates WSCI. Sun also provides the Sun ONE WSCI editor, which supports the WSCI extensions to WSDL. Vendors supporting or planning to support the BPEL4WS specification include:

- ***Collaxa:*** Offers a complete orchestration platform for BPEL4WS



- ***IBM:*** Provides a BPWS4J runtime/editor for BPEL4WS from their alphaWorks Web site
- ***BindSystems:*** Provides a BPEL4WS modeling/editing tool
- ***Microsoft, BEA, and other vendors:*** Announced they will support BPEL4WS in their products

While the industry appears to be embracing the BPEL4WS initiative, it is still unclear what part WSCI and the W3C Web services choreography working group will play. Clearly, vendor backing and tools support will influence the adoption taken by the software industry.

Case Study

To illustrate some of the capabilities outlined here, let's look at how Web services orchestration and BPEL4WS can solve a typical use-case scenario. It revolves around a purchasing system where a PC manufacturer wishes to build a set of PC machine configurations using a list of available suppliers. In the process, a buyer works through a purchasing agent to fulfill these inventory requests. The purchasing agent then communicates with a number of suppliers, each offering specific components required to build the PC configuration. Once a complete configuration can be built across one or multiple suppliers, a proposal is constructed and sent back to the buyer. The buyer then has the opportunity to place the parts order or cancel the request. Figure 4 shows a simplified view of the process. It shows the initial request from the buyer to the agent, with subsequent requests to each supplier.

Each partner in the process has a WSDL describing the specific input and output interfaces that are being exposed. This example will demonstrate the workflow that is built from the perspective of the purchasing agent, as well as the public interfaces exposed by this workflow.

The first step in creating the BPEL4WS document is to define the process itself. This starts with a <process> tag at the root level. This tag provides a name for the process and lists specific references to XML namespaces used. This is where any WSDL references are placed in the BPEL4WS document. In this example, the xmlns:po (http://acme-manufacturing.com/purchaseorder) will be used to refer to the WSDL definitions.

The next step is to define the specific parties involved in the process. In this example, there are three basic roles: (1) the buyer making the purchase; (2) the purchasing agent working on behalf of the buyer; and (3) a set of suppliers offering computer parts. This is supported in BPEL4WS through the <partnerLinks> and <partner Link> tags.

In Listing 3, a "Buyer" partner link is defined between the buyer (requestor) and the purchasing agent (purchaser). The partnerLinkType (po:requestQuote LinkType) is a reference to a <partner LinkType> tag defined within the WSDL document (see Listing 4).

The partnerLinkType defines the dependencies between the services and the WSDL port types that are used. We will assume there is a WSDL port type defining a request_quote operation that is initiated from the buyer. The purchasing agent will also have a link to the supplier for requesting a quote for a single part.

The process must also manage the flow of information between the partners, modeled as variables within BPEL4WS. In this scenario, a buyer makes the initial request with a configuration number and quantity to purchase, and the agent then constructs individual quote requests to each supplier with a part number and quantity. The requests come back from the supplier with the pricing information. The purchasing agent then constructs a proposal back to the buyer. Here, there are potentially four variables required to model this interaction, two for each request/response interaction. Each variable is declared with a name, followed by a reference to a WSDL message type (see Listing 5).

In this process, there must be a way to correlate the message requests to each other. For example, there might be a unique identifier for the quote that is received back from a supplier. The WSDL document would first define a correlation property for this quoteID, which would then be referenced within the BPEL4WS process. Listing 6 highlights how the correlation property is described in the WSDL document.

A key part of the BPEL4WS document is the definition of the steps required to handle the request. This is where basic and structured activities are used. The process flow consists of an initial request from the buyer, followed by invocations to multiple suppliers in parallel, followed by a reply back to the buyer of the completed proposal. The <sequence> tag is used for executing components sequentially; the <flow> tag is used for parallel execution; and the <receive>, <reply>, and <invoke> tags handle the basic activities required to interact with the services (see Listing 7).

The first step in the process flow is the initial buyer request. Once this request is received, a parallel set of activities is executed using the <flow> tag. Each supplier will be contacted in order to receive quotes for specific PC components. Each references a specific WSDL operation (e.g., request_quote), using variables for input and output. Upon receiving the responses back from the suppliers, the purchasing agent would construct a message back to the buyer. This would involve use of the <assign> tag in BPEL4WS and the XPath language to take the data received from the suppliers and build a final proposal to the buyer.

The final step in this scenario is the management of exceptions. For example, if there is an error in contacting a supplier, the agent may want to send a message back to the buyer. Within BPEL4WS, this would be done with fault handlers (see Listing 8).

You may need to set up compensation handlers for the process. For example, if one of the suppliers can't be contacted while placing the order, there should be a way to roll back the order. To set up a transactional context in BPEL, the <scope> tag is used to group related activities together. In this scenario, the three parallel invocations to the suppliers might be a good candidate for a scope declaration.

In a Nutshell

Orchestration and choreography are terms related to connecting Web services in a collaborative fashion. The capabilities offered by the available standards will be vital for building dynamic, flexible processes. The goal is to provide a set of open, standards-based protocols for designing and executing these interactions involving multiple Web services.

Many vendors have announced support for BPEL4WS in their products, and the OASIS technical committee is looking to move this specification going forward. WSCI is being considered by the W3C for

Web services choreography. While BPEL4WS has defined a notion of choreography through abstract processes, it is still unclear whether this will be accepted over the W3C work. Clearly, market adoption will be driven by the direction taken by vendors and their support of the standards in their product implementations.

As these standards take shape, it will be important to pay close attention to the direction taken by standards bodies such as the W3C and OASIS. There is still some confusion on how these efforts will come together, if at all. And many organizations are concerned over how reliability and security will be addressed. The good news is that much progress has been made by the major vendors embracing these standards, bringing great promise for Web services orchestration and choreography going forward.

Listing 1: **WSCI Example**

```
<process name="Purchase" instantiation="message">
 <sequence>
  <action name="ReceiveOrder" role="Agent" operation=
    "tns:Order">
  </action>
  <action name="Confirm" role="Agent" operation="tns:Confirm">
   <correlate correlation="tns:ordered"/>
   <call process="tns:Purchase"/>
  </action>
 </sequence>
</process>
```

Listing 2: **Illustration of a sequence in BPEL4WS**

```
<sequence>
  <receive partner="buyer" … operation="sendOrder"
     cariable="request"/>
  <invoke partner="supplier" … operation="request"
     inputVariable="itemreq" outputVariable="itemqt"/>
  <reply partner="buyer" … operation="response" variable="
     proposal"/>
</sequence>
```

Listing 3: **Define partner roles in BPEL4WS**

```
<partnerLinks>
 <partnerLink name="Buyer"
partnerLinkType="po:requestQuoteLinkType"
  myRole="Purchaser"/>
 <partnerLink name="Supplier1"
partnerLinkType="po:requestPartQuoteLinkType"
  myRole="Requestor" partnerRole="Purchaser"/>
 <!--Set up other suppliers used in this process -->
</partners>
```

Listing 4: **Partner link types defined in WSDL**

```
 <!-- partnerLinkType defined in the WSDL document -->
 <plnk:partnerLinkType name="requestQuoteLinkType">
 .<plnk:role name="Purchaser">
   <plnk:portType name="po:request_quote"/>
  </plnk:role>
 </plnk:partnerLinkType>
```

Listing 5: **BPEL4WS variables required for the process**

```
<variables>
 <variable name="request" messageType="po:request"/>
     <variable name="part_request
messageType="po:part_request"/>
     <variable name="part_quote" messageType="po:part_quote"/>
     <variable name="proposal" messageType="po:proposal"/>
</variables>
```

Listing 6: **Correlation properties defined in WSDL**

```
<definitions name="properties" …>
<bpws:property name="quoteID" type="xsd:string"/>
</definitions>
 <definitions name="correlatedMessages ..>
  <bpws:propertyAlias propertyName="cor:quoteID"
messageType="po:part_quote" …>
 </definitions>
```

Listing 7: **BPEL4WS process flow for the scenario**

```
<sequence>
     <receive name="receive" partnerLink="Buyer"
operation="request"
   variable="request" initiate="yes">
     </receive>
     <flow name="supplier_flow">
       <invoke name="quote_supplier1" partnerLink="Supplier1"
operation="request_quote"
        inputVariable = "part_request"
outputVariable="part_quote">
       </invoke>
    <!-- invoke other suppliers as part of the process, done
        in parallel -->
     </flow>
     <!-- construct a proposal from the part quotes received
       -->
     <reply name="reply" partnerLink="Buyer"
operation="send_proposal" variable="proposal">
     </reply>
</sequence>
```

Listing 8: **Setting up fault handlers**

```
<faultHandlers>
 <catch faultName="cantFulfillRequest">
  <invoke partner="buyer" operation="sendError"
inputVariable="fault"/>
 </catch>
</faultHandlers>
```

Download the code at

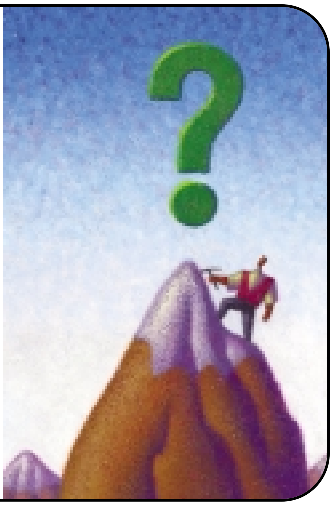
sys-con.com/webservices

Written by Michael Blank

Web Services in the Real World

Why Web Services Work

A positive impact on business



Having been endorsed by virtually every technology vendor on the planet, Web services are now evolving from "feature" to "fabric." They are moving from the latest buzzword (hot new feature) to a mature and accepted technology (fabric of the technology landscape). The hype is fading; it is no longer interesting to develop Web services simply as a proof of technology, or as an end in themselves.

This series explores the use of Web services in real-world situations, with the purpose of identifying usage patterns. The idea behind the series is to help answer questions like: Where and how do Web services deliver value? Where might they be counter-indicated? What works? What doesn't?

For the first time, there is enough data from real-world, business-driven projects to allow us to begin to recognize patterns. The data analyzed here is drawn from production Web services case studies culled from two sources:

- Sand Hill Group's study, "The Web Services Derby" (www.sandhill.com)
- webMethods customers employing Web Service-Based Integration (WSBI)

Where possible, case studies are attributed to specific companies. However, in many cases organizations requested anonymity, generally because their projects are not public and are considered competitive differentiators.

Reasons for Choosing Web Services

This month, we explore the reasons project executives gave for why they chose Web services. Companies that have deployed WSBI projects reported that Web services have such compelling characteristics that they translate into tangible business benefits. Early implementations have shown that Web services can significantly reduce development and maintenance costs while yielding a highly responsive IT infrastructure. Web services are so important that 25% of the companies interviewed indicated that their projects would not have happened without Web services because of the lack of feasible alternative approaches.

Four major reasons for using Web services emerged from the analysis. A few initiatives were too broad or too unique to categorize, but the vast majority – over 70 of the 80 case studies reviewed – coalesced around the following major themes:

- Simplicity
- Interoperability
- Abstraction
- Reuse

According to the companies interviewed for this research, each of these contributed directly to business benefits such as increased revenues, decreased costs, and improved customer satisfaction.

Reason #1: Simplicity

Let's face it: the big draw of Web services is that they're easy to work with. Compared to EDI or programming interfaces, WSDL is much easier for people to understand. Even nonprogrammers can use Web services tools to create useful solutions, and they don't need to be SOAP or WSDL experts. No specialized training is required, and chances are your developers can get ramped up quickly (if they aren't already).

For example, a logistics and transportation company needed to make it easier for developers at partner companies to remotely call the company's shipping functionality from within their applications. In the past, the company distributed software development kits (SDKs) for several platforms, but it was a nightmare to maintain and never gained any traction. Web services were a perfect solution that allowed developers to use whatever development tool they already had. The company believes that its transition to Web services will make its shipping functionality ubiquitous, and will serve as a customer acquisition tool.

Andy Ellicot, technology director for Infinity Pharmaceuticals, notes that he didn't have to hire expensive programmers to implement his Web service projects. "Our team of technical analysts was able to deliver the project without having to get the engineering team involved. This helped us save time and money," adds Ellicott. The technical analysts, who knew the end-user application but weren't programmers, used a Web services–based integration platform to orchestrate individual Web services developed by engineering into business process–specific Web services that were exposed to the portal team. The graphical environment made

Author Bio

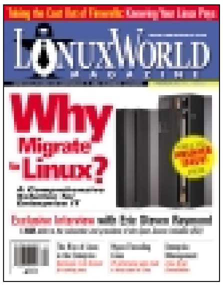
Michael Blank is a founding member of webMethods, Inc., and was its first software engineer. During his tenure, he has started and commercialized several product offerings. As director of developer marketing, he manages webMethods' developer communities as well as the software evaluation program (http://evals.webmethods.com).

MICHAEL@WEBMETHODS.COM

A LIMITED TIME SAVINGS OFFER FROM SYS-CON Media

SUBSCRIBE TODAY TO MULTIPLE MAGAZINES

AND SAVE UP TO $400 AND RECEIVE UP TO 3 FREE CDs!



TO ORDER •Choose the Multi-Pack you want to order by checking next to it below. •Check the number of years you want to order. •Indicate your location by checking either U.S., Canada/Mexico or International. •Then choose which magazines you want to include with your Multi-Pack order.

Pick a 3-Pack, a 6-Pack or a 9-Pack

□ 3-Pack	□ 1YR	□ 2YR	□ U.S.	□ Can/Mex	□ Intl.
□ 6-Pack	□ 1YR	□ 2YR	□ U.S.	□ Can/Mex	□ Intl.
□ 9-Pack	□ 1YR	□ 2YR	□ U.S.	□ Can/Mex	□ Intl.

□ **Linux World Magazine**

U.S. - Two Years (24) Cover: $143	You Pay: $79.99 /	Save: $63 + FREE $198 CD
U.S. - One Year (12) Cover: $72	You Pay: $39.99 /	Save: $32
Can/Mex - Two Years (24) $168	You Pay: $119.99 /	Save: $48 + FREE $198 CD
Can/Mex - One Year (12) $84	You Pay: $79.99 /	Save: $4
Int'l - Two Years (24) $216	You Pay: $176 /	Save: $40 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

□ **Java Developer's Journal**

U.S. - Two Years (24) Cover: $144	You Pay: $89 /	Save: $55 + FREE $198 CD
U.S. - One Year (12) Cover: $72	You Pay: $49.99 /	Save: $22
Can/Mex - Two Years (24) $168	You Pay: $119.99 /	Save: $48 + FREE $198 CD
Can/Mex - One Year (12) $84	You Pay: $79.99 /	Save: $4
Int'l - Two Years (24) $216	You Pay: $176 /	Save: $40 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

□ **Web Services Journal**

U.S.- Two Years (24) Cover: $168	You Pay: $99.99 /	Save: $68 + FREE $198 CD
U.S. - One Year (12) Cover: $84	You Pay: $69.99 /	Save: $14
Can/Mex - Two Years (24) $192	You Pay: $129 /	Save: $63 + FREE $198 CD
Can/Mex - One Year (12) $96	You Pay: $89.99 /	Save: $6
Int'l - Two Years (24) $216	You Pay: $170 /	Save: $46 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

□ **.NET Developer's Journal**

U.S. - Two Years (24) Cover: $168	You Pay: $99.99 /	Save: $68 + FREE $198 CD
U.S. - One Year (12) Cover: $84	You Pay: $69.99 /	Save: $14
Can/Mex - Two Years (24) $192	You Pay: $129 /	Save: $63 + FREE $198 CD
Can/Mex - One Year (12) $96	You Pay: $89.99 /	Save: $6
Int'l - Two Years (24) $216	You Pay: $170 /	Save: $46 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

□ **XML-Journal**

U.S. - Two Years (24) Cover: $168	You Pay: $99.99 /	Save: $68 + FREE $198 CD
U.S. - One Year (12) Cover: $84	You Pay: $69.99 /	Save: $14
Can/Mex - Two Years (24) $192	You Pay: $129 /	Save: $63 + FREE $198 CD
Can/Mex - One Year (12) $96	You Pay: $89.99 /	Save: $6
Int'l - Two Years (24) $216	You Pay: $170 /	Save: $46 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

□ **WebLogic Developer's Journal**

U.S. - Two Years (24) Cover: $360	You Pay: $169.99 /	Save: $190 + FREE $198 CD
U.S. - One Year (12) Cover: $180	You Pay: $149 /	Save: $31
Can/Mex - Two Years (24) $360	You Pay: $179.99 /	Save: $180 + FREE $198 CD
Can/Mex - One Year (12) $180	You Pay: $169 /	Save: $11
Int'l - Two Years (24) $360	You Pay: $189.99 /	Save: $170 + FREE $198 CD
Int'l - One Year (12) $180	You Pay: $179 /	Save: $1

□ **ColdFusion Developer's Journal**

U.S. - Two Years (24) Cover: $216	You Pay: $129 /	Save: $87 + FREE $198 CD
U.S. - One Year (12) Cover: $108	You Pay: $89.99 /	Save: $18
Can/Mex - Two Years (24) $240	You Pay: $159.99 /	Save: $80 + FREE $198 CD
Can/Mex - One Year (12) $120	You Pay: $99.99 /	Save: $20
Int'l - Two Years (24) $264	You Pay: $189 /	Save: $75 + FREE $198 CD
Int'l - One Year (12) $132	You Pay: $129.99 /	Save: $2

□ **Wireless Business & Technology**

U.S. - Two Years (24) Cover: $144	You Pay: $89 /	Save: $55 + FREE $198 CD
U.S. - One Year (12) Cover: $72	You Pay: $49.99 /	Save: $22
Can/Mex - Two Years (24) $192	You Pay: $139 /	Save: $53 + FREE $198 CD
Can/Mex - One Year (12) $96	You Pay: $79.99 /	Save: $16
Int'l - Two Years (24) $216	You Pay: $170 /	Save: $46 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

□ **WebSphere Developer's Journal**

U.S. - Two Years (24) Cover: $360	You Pay: $169.99 /	Save: $190 + FREE $198 CD
U.S. - One Year (12) Cover: $180	You Pay: $149 /	Save: $31
Can/Mex - Two Years (24) $360	You Pay: $179.99 /	Save: $180 + FREE $198 CD
Can/Mex - One Year (12) $180	You Pay: $169 /	Save: $11
Int'l - Two Years (24) $360	You Pay: $189.99 /	Save: $170 + FREE $198 CD
Int'l - One Year (12) $180	You Pay: $179 /	Save: $1

□ **PowerBuilder Developer's Journal**

U.S. - Two Years (24) Cover: $360	You Pay: $169.99 /	Save: $190 + FREE $198 CD
U.S. - One Year (12) Cover: $180	You Pay: $149 /	Save: $31
Can/Mex - Two Years (24) $360	You Pay: $179.99 /	Save: $180 + FREE $198 CD
Can/Mex - One Year (12) $180	You Pay: $169 /	Save: $11
Int'l - Two Years (24) $360	You Pay: $189.99 /	Save: $170 + FREE $198 CD
Int'l - One Year (12) $180	You Pay: $179 /	Save: $1

OFFER SUBJECT TO CHANGE WITHOUT NOTICE

RECEIVE YOUR DIGITAL EDITION ACCESS CODE INSTANTLY WITH YOUR PAID SUBSCRIPTIONS

3-Pack

Pick any 3 of our magazines and save up to $275⁰⁰ Pay only $175 for a 1 year subscription plus a FREE CD

- 2 Year – $299.00
- Canada/Mexico – $245.00
- International – $315.00

6-Pack

Pick any 6 of our magazines and save up to $350⁰⁰ Pay only $395 for a 1 year subscription plus 2 FREE CDs

- 2 Year – $669.00
- Canada/Mexico – $555.00
- International – $710.00

9-Pack

Pick 9 of our magazines and save up to $400⁰⁰ Pay only $495 for a 1 year subscription plus 3 FREE CDs

- 2 Year – $839.00
- Canada/Mexico – $695.00
- International – $890.00

Subscribe Online Today www.sys-con.com/2001/sub.cfm

SYS-CON MEDIA

this easy, and they didn't require expertise with Java, .NET, or WSDL.

Reason #2: Interoperability

The most common technical reason cited for using Web services is platform and technology interoperability. For the first time, an entire industry has agreed on a common messaging format. Almost every software vendor is adding Web services support to its products, allowing "plug and play" interoperability among the portal, desktop, and back-office applications.

Because of Web service interoperability, companies can implement a loosely coupled, global IT infrastructure. Consider the CIO of a large rental car company in Europe, faced with the difficult task of integrating every reservation system in Europe so that a customer can pick up a car in France and drop it off in Germany. This company has grown by acquisition, and each country has their own custom reservation system. Without Web services, his alternative was to standardize on a technology, and then force each country to implement it. A risky venture, even if he wins the political battle. With Web services, though, he just needs to define and gain consensus on the interface. The interface, then, can be implemented by each of the countries with whatever technology they prefer.

Web service interoperability was a key factor for companies that needed to connect their portals to their back office. In fact, over half of the companies surveyed used Web services with a portal or application server. In this scenario, an integration platform is typically used as the central Web services provider that feeds the portal with various back-end data sources. For example, at NEC Electronics America, Web services connect their application server to their back-end databases' integration platform. The project would have taken another 4–6 weeks with an alternative, proprietary approach, such as RMI, and may not have been easily reusable by other business units.

Perhaps the most significant aspect of Web service interoperability is the ability to bridge J2EE and .NET environments. For Infinity Pharmaceuticals, this was the primary driver for using Web services. Andy Ellicot says, "because many applications speak SOAP they can be easily plugged into the architecture. I'm not locked into any one proprietary solution." Various commercial off-the-shelf (COTS) products plug into the integration framework via Web services. This has allowed Infinity to quickly adapt their IT infrastructure to take advantage of market opportunities. A leading mortgage lender and financial services company relied on Web service interoperability to efficiently connect their various portals to their integration platform. Because of acquisitions, the company had to maintain both J2EE and .NET environments. Web services were the only way to feasibly connect these portals to their back office.

Reason #3: Abstraction

Abstraction refers to the process of hiding all but the most relevant data about an object in order to reduce complexity and increase processing efficiency. Companies indicated that Web services abstraction:

1. Eases integration between organizations
2. Reduces complexity
3. Enables application "plug-and-play"

Eases Integration Between Organizations

Nearly all of the companies interviewed are using Web services in projects that require integration with another organization, i.e., another internal team or an external trading partner. The issue with integrating with another organization is that you rarely (a) control them and their technology decisions or (b) have expertise about their systems. With Web services you agree on the interface, not necessarily on the technology, and you don't require domain expertise about the other's back-end systems. Therein lies the power of Web services abstraction.

Future Electronics, a multibillion-dollar electronics components distributor, decided to use WSBI to help ease integration between their e-commerce portal and the company's mainframe system that processed all orders. Without Web services, the only way to accomplish the integration with the mainframe was for the e-commerce team to access the mainframe database tables directly. Understandably, the mainframe team was highly uncomfortable with this. Furthermore, the e-commerce team would have had to acquire mainframe expertise. Instead, they agreed on the (WSDL) interface, and implemented that interface using technology that was already familiar: the e-commerce team used the webMethods Integration Platform and the mainframe team deployed a mainframe SOAP add-on product. Brad Hudon, director of IT Development for Future Electronics, says "Web services allowed us to implement a 'black-box' integration framework that lets each of the teams retain their existing business logic." He adds that the project would not have succeeded without the use of Web services.

Reduces Complexity

Many companies are building "business service hubs" that consolidate and standardize the myriad interfaces, protocols, and data formats they had before. A single Web service may trigger a complex business process that touches several systems and data sources. However, the consumer only invokes a simple Web service and need not be concerned with the intricacies of the back-end systems.

A large financial services company consolidated 50 back-end systems into a simple set of Web services that are consumed by 20 front-office and customer-support applications. One Web service operation, called Update Customer Information, for example, sets off a complex business process that sends the data to 10 different customer systems. However, all of this complexity is abstracted to the consumer of that service, greatly speeding the integration.

Avnet Computer Marketing, a leading provider of enterprise systems, software, networking, and storage, was able to reduce the cost of maintaining and extending their integration architecture by consolidating their FTP, XML, and IP socket interfaces and standardizing on Web services. Now the e-commerce portal team can focus on the presentation layer and no longer needs to know about these various proprietary interfaces and file formats – they just call a Web service.

Enables Application "Plug-and-Play"

With Web services, companies can replace a back-end system without impacting the overall application if they maintain the WSDL interface. An auto manufacturer, for example, had to retire its legacy system as quickly as possible. The company wrapped the system with a Web services interface with the goal of replacing it with newer technology. The interface was designed carefully so that it can be kept stable while migrating the application.

Reason #4: Component Reuse

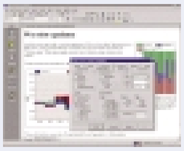
Many companies are building business service hubs not only to consolidate their interfaces but to maximize reuse. These service hubs aggregate individual application capabilities into coarsely grained business services that are exposed for use by multiple systems. Business service projects are very common where multiple front-end systems need to access multiple back-end systems. The biggest benefit of Web services doesn't occur the first time, but on recurring integrations. The speed of execution becomes faster as previously developed services are reused because developers do not constantly have to rewrite code.

For example, a large financial services company implemented Web services because they planned to connect multiple front-office applications that needed to access the same type of data. A service hub exposed their back-end systems as a simple set of Web services. While Web services didn't save time on the first connection, they did have significant impact for subsequent connections. Implementing the first Web services–based business process and integrating it with the customer portal took seven weeks, but the second connection to the automated voice response system took only several days (much to the surprise of the project manager, who had initially budgeted two months!).

Avnet Computer Marketing already had several Web services projects under their belt, and today integration projects are measured in days instead of weeks. Integrating the quote-to-order application with the mainframe took about six weeks, but connecting the order entry e-commerce system to the same Web service took only one day.

Conclusion

The companies that have deployed WSBI projects report that the combination of simplicity, interoperability, abstraction, and reuse make Web services so compelling that they translate into tangible business value. These pioneering companies say Web services will survive because they make a business impact, rather than being adopted for technology's sake. These companies are realizing genuine business benefits and plan to continue on the road to adopting Web services. ⓔ

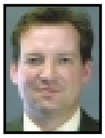


Written by Jon Pellant

“Smart” BPM

The who, what, where, and when for Web services

Companies have been looking for ways to open, or expose, key pieces of their enterprise applications to customers and partners since the Internet took off as a business tool. They want the efficiency, cost savings, and ability to conduct business around the clock that access to key applications offers.



Jon Pellant is the director of technology for PegaRULES, Pegasystems’ patented business rule development and deployment technology. He supports key sales and partnering initiatives through such activities as architectural presentations to customers, speaking at trade shows and conferences, analyst relationships, developing thought leadership papers, and providing market and product feedback to the PegaRULES management team.

JON.PELLANT@PEGA.COM

But so far, Web services have yet to fulfill their potential to automate processes that improve service and drive profit while making life easier for employees, customers, and partners. Self-service change-of-address applications are fine, but they lack the ability to make the kinds of smart business decisions that competitive businesses must make daily. Online customer self-service applications cannot make decisions based on the worth of the customer; for example, such as whether to delay a customer’s order by using batch processing for an address change or to ensure that it takes effect immediately, despite a slightly higher cost to the company. That’s where an older technology with a new twist comes in: “smart” business process management (BPM).

Smart BPM, Web services, and portals are colliding to create an opportunity for companies to take common Web services, such as customer self-service, to the next level of intelligent automation. Portals offer a familiar network environment for shared applications. Smart BPM – BPM driven by business rules – can provide the missing intelligence needed to fully automate complex transactions. Smart BPM allows business people to set business priorities – not the programmers who implement them – to guide how rules influence the way work gets done. The ability to change business rules frequently without rocking the system is the key to agility.

BPM is not a new technology, but traditional BPM alone is not enough to support mature and nuanced transactional automation. Smart BPM, built on sophisticated rules engines, joins process automation and complex Web services transactions through best-practice rules and fine-grained decision intelligence at the point of integration. Business rules engines and process automation can add the intelligence and agility to Web services that companies need if they are to realize the full benefits of exposing key applications.

What’s Not to Like

Smart BPM relies on business rules, workflow, and enterprise application integration (EAI) to automate haphazard and cumbersome manual processes. On their own, these process technologies did little more than push and assign work. Desktop workflow applications streamlined processes, eliminating manual steps and paperwork. EAI linked back-office systems and automated transactions with point-to-point interfaces. As Web services have matured, many organizations have exceeded these efficiencies to expose cost-effective business transactions and services to partners and employees. Companies are using Web portals supported by XML and other Web service technologies to bypass application interfaces entirely, allowing users to submit expense reports without logging into an ERP system, for example. Ramp-up time for Web services is usually only weeks or months, and the ROI equation is proven.

Unfortunately, the Web service implementation model doesn’t scale well in dynamic enterprises. Part of the issue is the classic software development model. IT developers start with a concrete set of use scenarios that they methodically “phase” into the portal. However, IT must juggle new scenarios and capabilities, exception handling in old scenarios, modifications to back-office Web services, and shifts in business priorities. It’s not long before the development team is fighting an uphill battle, the portal stagnates, and the push is on for a more agile solution.

But the classic software development model isn’t the only barrier to rolling out successful, sophisticated Web services. Typically, Web services lack the process intelligence that brings business-level discernment – the use of if/then logic – to make sophisticated choices about how to process transaction input. One problem this causes is that the more Web services proliferate, the more challenging it becomes to ensure that a given transaction applies the proper service (there may be several “change address” Web services available, for example, although one may be for partners and another for customers). The burden of choosing the right one in a given context rests with the application, which adds to the programmer's task.

Moreover, Web services are usually stateless. The management of policies, security, and best practices is placed on the portal developers – an open door for inconsistencies. And when business policies, practices, or rules change, portal developers must code the logic changes into the individual applications because their applications aren't *smart* enough to adapt on their own.

A better approach is to architect a smart BPM layer that abstracts the exposed Web services as a logical business process. Such a software layer applies global policies, business rules, and best practices to the entire portal platform. It manages and executes business processes centrally, tying exposed services into enterprise applications. It's smart because it takes advantage of advanced decision technology to resolve nuanced business rules in real time.

While the *how* of a process falls to Web service developers, smart BPM systems apply *who*, *what*, *when*, and *where* to the business context of a request. It executes the appropriate Web services via an integrated rules engine that models corporate policies, business rules, and best-practice knowledge.

Smart BPM Adds Agility

Smart BPM is crucial to Web services because business rules should be managed by the business, not programmers. Web services governed by smart BPM tailor their actions and responses to the applications calling on them, in effect creating custom services for each user. Intelligent systems leverage inference technologies to choose the correct process based on the context in which it is used. This eliminates the need for IT personnel to aggregate a set of rules for a specific process, and makes the whole system more adaptable to change. Smart BPM systems scale well because IT staff can concentrate on adding new functionality, exceptions, and Web services instead of hand-coding every rule change into each system it touches.

For example, an address change sounds like a simple Web service. But consider a large national bank. A "simple" address change may consume more than 70 process flows and involve close to 40 back-office systems. While many flows concern themselves with the "hows" of changing the address information in the back office, many more will need to deal with elements like correspondence, fraud prevention, and auditing. The rest of the context involves delegated business rules up-selling or cross-selling based on relative customer value, subscription status, demographics, marketing initiatives, change notifications, and so on.

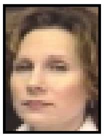
If a smart BPM layer manages all of these flows, the different portals involved don't have to know any of the address change details. If the bank changes a customer value policy, the rule is instantly enforced across the enterprise. There's no need for IT to aggregate the rule and distribute the change to multiple systems. This saves money and time, but its higher value lies in what it can do for Web services as a whole.

Smart BPM's rule-guided intelligence promises to simplify and coordinate Web services for real-world business agility, something that will be critical to all business in the 21st century. While exposing Web services is key to creating automated systems, true agility comes from the intelligent use of business rules, policies, and best practices. Smart BPM is one way Web services are adapting to their own momentum and success. ⓔ

Business Process Automation

A gateway to better business management

Today's economy requires that businesses take a pragmatic approach to managing productivity and costs. The mandate is to optimize efficient processes throughout the organization. In the ongoing search to improve IT productivity, many companies are looking at business process management (BPM) as a means to increase productivity and extend system functionality without requiring rewrites of corporate policies and procedures.



Joan Lynn is the documentation and knowledge manager for Teamplate Inc., which develops the platform for rapid business process automation in Microsoft environments. Prior to joining Teamplate, Joan spent over 10 years as a corporate educator and documentation specialist creating, producing, and delivering customized training solutions to national and multinational organizations.

JOAN.LYNN@TEAMPLATE.COM

The BPM/BPA Advantages

BPM details how a business operates by analyzing the roles of personnel, areas of business, and processes. Business process automation (BPA) goes one step further by automating various aspects of those roles, areas, and processes to ensure consistent and repeatable performance.

An important side benefit of BPA comes from capturing and preserving the business function knowledge of key individuals. This repository of business knowledge rewards organizations with increased productivity, a better understanding of the company's structure, and formalized procedures that stay with the company even if process-founding workers leave.

BPM helps an organization focus on acquiring, storing, and utilizing knowledge for such things as problem solving, dynamic learning, strategic planning, and decision making. It also protects intellectual assets from decay, adding to an organization's overall intelligence and providing increased flexibility.

Using BPM to automate activities also helps to monitor the progress or bottlenecks of a company. Once implemented, automated business processes can assist in the training of new employees, or provide a standardization of processes internally for use with the organization's customers and/or suppliers.

To begin developing business process workflows, an organization needs to analyze how it works and what its people know, and identify areas for improvement. Many organizations discover these areas as a process of continuous analysis, but as a general rule BPA has the greatest economic impact in situations where:

- Processes are currently managed via paper forms that are physically moved around the organization.
- Multiple employees or teams of employees are collaborating on a single document or unit of work.
- Approval steps are numerous and business critical.

Once the organization identifies an ideal area to be automated, the next step is implementation.

About the Project

As with any project, selecting the right team members to create a BPA solution is crucial to its success. Along with the IT personnel needed to design, develop, and deploy an automated process, a workflow solution team should include one or two knowledge workers (also known as Subject Matter Experts), a business analyst, business manager, and technical writer.

As the process requirements are defined, the IT professional will consider whether to build a custom solution or purchase an existing BPM solution. The benefit of a custom solution is its flexibility and customization attributes. The downside is that the IT team will be responsible for the maintenance of the code. Internal development will also require significant resource commitments of time, equipment, and expertise.

If the choice is to purchase a BPM solution, it is important that the organization is clear about the cost and anticipated return on its investment. In addition, the team, including (potentially) the BPM vendor, needs to agree on projected deployment dates.

In either case, the quicker a workflow solution (see Figure 1) is made available, the sooner an organization will realize increased productivity, efficiency, and cost savings. Figure 1 provides an example of a workflow model.

With time and budget concerns in mind, companies will often opt to purchase BPM engines as they typically shorten development and deployment cycles. Many tools associated with a BPM engine's functions capture, model, implement, and manage the components and activities of a business through the use of

Premiering
August
2003

www.LinuxWorld.com

Taking the Cost Out of Firewalls: Knowing Your Linux Pays
LINUXWORLD
MAGAZINE
Why Migrate to Linux?
A Comprehensive Solution for Enterprise IT
SPECIAL PREMIER ISSUE! LWM
Exclusive Interview with Eric Steven Raymond
LWM talks to the cofounder and president of the Open Source Initiative (OSI)
The Rise of Linux in the Enterprise
Hyper-Threading Linux
Enterprise Management

The
Leading Magazine
for Corporate
and IT Managers

LinuxWorld Magazine

There is no escaping the penetration of Linux into the corporate world. Traditional models are being turned on their head as the open-for-everyone Linux bandwagon rolls forward.

Linux is an operating system that is traditionally held in the highest esteem by the hardcore or geek developers of the world. With its roots firmly seeded in the open-source model, Linux is very much born from the "if it's broke, then fix it yourself" attitude.

Major corporations including IBM, Oracle, Sun, and Dell have all committed significant resources and money to ensure their strategy for the future involves Linux. Linux has arrived at the boardroom.

Yet until now, no title has existed that explicitly addresses this new hunger for information from the corporate arena. *LinuxWorld Magazine* is aimed squarely at providing this group with the knowledge and background that will allow them to make decisions to utilize the Linux operating system.

Look for all the strategic information required to better inform the community on how powerful an alternative Linux can be. *LinuxWorld Magazine* will not feature low-level code snippets but will focus instead on the higher logistical level, providing advice on hardware, to software, through to the recruiting of trained personnel required to successfully deploy a Linux-based solution. Each month will see a different focus, allowing a detailed analysis of all the components that make up the greater Linux landscape.

Regular features will include:

Advice on Linux Infrastructure

Detailed Software Reviews

Migration Advice

Hardware Advice

CEO Guest Editorials

Recruiting/Certification Advice

Latest News That Matters

Case Studies

SAVE 30% OFF!

REGULAR ANNUAL COVER PRICE $71.76

YOU PAY ONLY

$49.99

12 ISSUES/YR

*OFFER SUBJECT TO CHANGE WITHOUT NOTICE

SUBSCRIBE TODAY!

WWW.SYS-CON.COM
OR CALL
1-888-303-5282

FOR ADVERTISING INFORMATION:
CALL 201 802.3020 OR
VISIT WWW.SYS-CON.COM

SYS-CON MEDIA
The World's Leading i-Technology Publisher

graphical process design, database access, document assembly, e-mail integration, and other features such as integration with other systems and devices.

There are some important points to consider when purchasing a BPM tool:

- Integration with other systems
- Security
- Device support
- Flexibility
- Scalability

A good BPM solution should provide wizards or an exposed application program interface (API) for integration in order to easily integrate with existing applications used in an office.

Scalability should be considered not only for the current number of users but with consideration for the number of future users that will be added in the future as additional workflow solutions are deployed.

Other factors, such as security, device support, and flexibility, are dependent on the actual deployment. If the workflow solution is strictly internal, it will have different specifications and requirements than one that is Web based.

The organization may start with one automated process for a single department that initially resides within a firewall. Once completed, it could be expanded to include additional groups and then extended beyond the firewall.

About Web Services

Web services are being rapidly adopted as part of workflow solutions. As organizations move towards automated workflow across departments, they need a system to connect contrasting applications and unrelated platforms.

Web services access APIs in order to allow applications to share information. For example, using Web services a developer creates a purchase order workflow that provides live information on the shipping status by incorporating a "Web service" provided by a shipping company. (The implementation may be done through a wizard-based interface.)

If the shipping firm provides the information via a Web service, the information can be accessed and presented automatically and dynamically in the workflow solution. The end users become more efficient as their time is now spent completing tasks rather than looking for information (like shipping status).

Another example would be a developer using Web services who creates a workflow solution that monitors inventory levels as orders are processed. When the inventory falls below a set level, workflow is automatically triggered to send an order to a supplier. Using XML and Web services, the supplier acknowledges the request and processes the shipment without the need for further human intervention.

In essence, Web services allow developers to build workflow with a common language, platform, and protocol. Coupled with .NET, this gives a workflow solution the ability to:

- Manage the process itself, which in turn manages people, relationships, and what they do
- Deal with various applications on disparate platforms
- Provide the end user with dynamic data updates

About .NET

The Microsoft .NET Framework is being adopted by organizations as a means to connect systems and improve the flow of information. This framework makes it easier for companies to share and integrate existing systems and

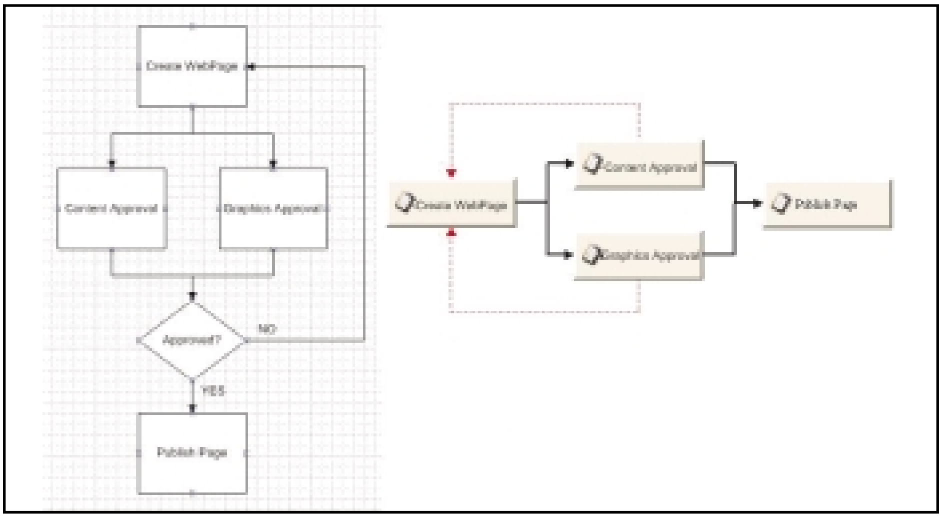


FIGURE 1 Workflow model

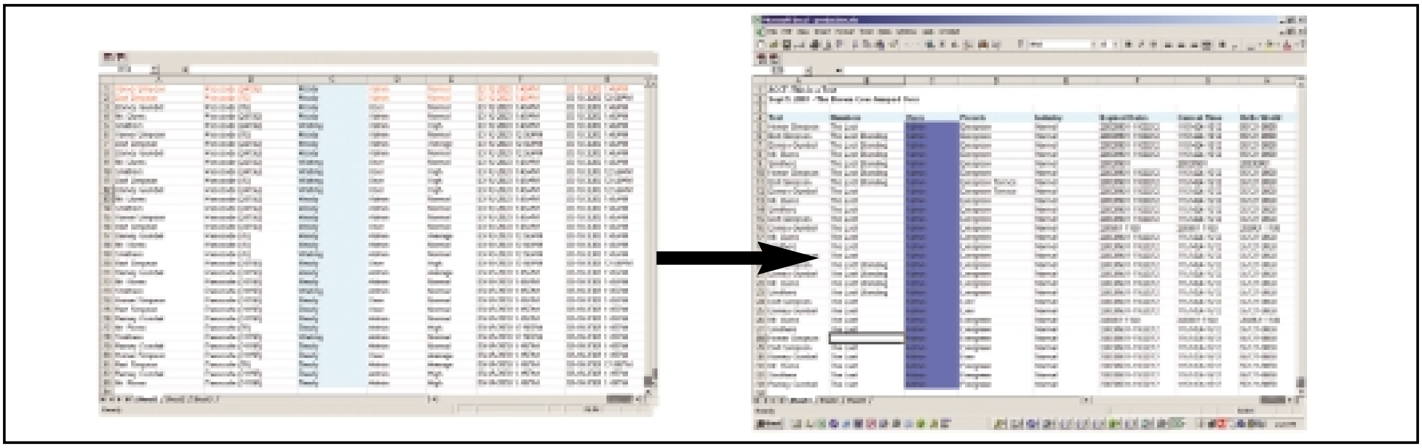


FIGURE 2 Exporting to Excel

information as well as extend functionality by using XML and Web services. As organizations move forward, they will find that information sharing is fundamental to automating a process across platforms and departments or groups.

As a business grows, it consistently adds to the processes and the number of systems it uses. When you consider building workflow solutions, an important point to consider is that .NET provides an ideal platform for interoperability and information sharing. With .NET, passing data from one system to the next becomes seamless and requires minimal programmatic manipulation.

Just as the .NET Framework provides organizations with the ability to link systems beyond a company's walls by providing users with Web and wireless-based computing, XML and Web services enable systems and devices to connect and interoperate with one another. XML is not only used as a mechanism to pass data from system to system within an organization, but also from business to business.

These technologies help automate workflow by leveraging interoperability and ensuring that a user does not have to be in proximity to the central enterprise to complete tasks. Workers can be anywhere in the world, using the Internet to complete work. Certain BPM engines also support wireless devices like PDAs.

Reporting Best Practices

Since an important element in setting up a workflow process is the selection of the correct BPM tool, organizations should further consider what that tool will provide. It is important that organizations select a tool that provides the ability to model the process as well as to integrate the technology.

Workflow solutions should incorporate reporting and monitoring capabilities. Management will require at-a-glance reporting. Even better, these reports should be easy to export to other technologies, such as Microsoft Excel (see Figure 2).

The Benefits

Initially, as with any project, implementing workflow solutions is not without its growing pains; however, the benefits are numerous. Organizations point out that BPM/BPA have helped in the automation of administrative tasks as well as reducing the costs of transactions or business events. These solutions have also improved coordination of activities and information/knowledge sharing across departments and locations.

Once processes have been modeled and therefore automated, organizations have also been able to reduce the number of tasks required in a process or the number of users needed to complete a transaction. By eliminating redundancies, efficiency and productivity are increased. This means quality improvement, fewer errors, higher productivity per person, as well as a reduction in the time required to complete a process. The likelihood of "something falling through the cracks" is greatly diminished.

Conclusion

BPM is not just a matter of decreasing or refining the number of tasks to complete a process. It is a business transformation project. It is a way of implementing new technology and tools, reutilizing existing systems, and leveraging the human factor to help evolve an organization as a more successful business. ⓔ

Perspectives On Third-Party Integration
BEA WebLogic DEVELOPER'S JOURNAL
DELIVERING SOA

Planning Your Moves: Focus on Migration
BEA WebLogic DEVELOPER'S JOURNAL
Upgrading to WebLogic Server 10

Where Open Minds Meet

- **Learn** how companies have achieved higher profits and increased their productivity by utilizing Linux
- **Participate** in LinuxWorld's **world-class** education program and benefit from interactive training in the **all-new Hands-on Labs!**
- **Discover** the latest innovations and technologies from the hottest companies around
- **Hear** the latest developments and updates on the state of open source at our analyst roundtable discussion

CORNERSTONE SPONSOR

hp invent

PLATINUM SPONSORS

Computer Associates DELL IBM intel Novell ORACLE pogo [linux] Merging Business with Linux redhat Sun microsystems We make the net work. VERITAS

GOLD SPONSOR

Hummingbird

SILVER SPONSOR

APPRO HPC Cluster Solutions

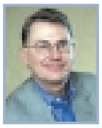
LINUXWORLD™ CONFERENCE & EXPO

Conference: August 4-7, 2003
Expo: August 5-7, 2003

The Moscone Center
San Francisco, CA

Join us this August and see why Linux is thriving! Government agencies and companies in the telecommunication, financial services, retail and manufacturing industries are turning to Linux to save money. Isn't it time you did? **LinuxWorld. Where Open Minds Meet.**

Attend Keynotes *and learn from inspiring visionaries who are devoted to Linux and open source.*

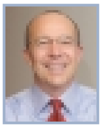


Tuesday, August 5th
10:30am-11:30am

Linux: The Next Step
Peter Blackmore
Executive Vice President
Enterprise Systems Group
Hewlett-Packard

Tuesday, August 5th
1:30pm-2:30pm

Jonathan Schwartz
Executive Vice President
Software Group
Sun Microsystems, Inc.



Tuesday, August 5th
4:30pm-5:30pm

Matthew Szulik
Chairman, Chief Executive Officer and President
Red Hat

Wednesday, August 6th
10:30am-11:30am

Linux and the Evolution of the Internet
Irving Wladawsky-Berger
General Manager
IBM Corporation

Wednesday, August 6th
3:30pm-4:30pm

Charles Rozwat
Executive Vice President
Server Technologies Division
Oracle Corporation

Special Presentation

Open to All Registered Attendees

Tuesday, August 5th
12:00pm-1:00pm

The Golden Penguin Bowl
Host: Chris DiBona, Vice President – Marketing; Co-Founder, Damage Studios, Inc.

Analyst Roundtable

Open to All Registered Attendees

Wednesday, August 6th
1:30pm-2:30pm

State of Open Source Roundtable
Moderator: Larry Augustin, Partner, Azure Capital Partners

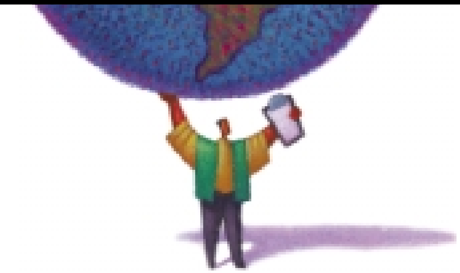
Panelists: Pierre Fricke, Executive Vice President of Web Application Infrastructure and Product Lifecycle Management (PLM) Infrastructure, D.H. Brown Associates, Inc.; Daniel Kusnetzky, Vice President, System Software Research, IDC; Ted Schadler, Principal Analyst in Software, Forrester; George Weiss, Vice President and Research Director, Gartner

IDG
WORLD EXPO

www.linuxworldexpo.com

Written by Harvey Reed

Web Services Journal
FOCUS
BPM



Managing the Reach and Range of Your Business Processes

The enterprise service bus and BPM

Business processes reach across enterprises and partners, and require a range of complex functions. As the reach and range of your business processes increase, consider (a) moving these functions into an integration network, such as an enterprise service bus (ESB); and (b) recursively encapsulating your business processes as services. The resulting architecture is agile without redundant and confusing technology.

AUTHOR BIO:

Harvey Reed is a technical product manager with Sonic Software (www.sonicsoftware.com). He joined Sonic after their acquisition of eXcelon, where he was product manager of eXcelon BPM. Prior to eXcelon, he was chief architect for several Release 1 software products, as well as a practice director for C-bridge, a systems integrator. Harvey is also chief inventor for a U.S. Patent in object testing technology.

HREED@SONICSOFTWARE.COM

'Reach' and 'Range'

The metric of distance describes how far the business process *reaches* to interact with the entities it orchestrates. Distance is relevant to two resources or services that are orchestrated by a business process management (BPM) tool. Consider what is important when a business process is deployed, managed, or updated.

These activities are performed by company employees and systems in a coordinated fashion, so *distance* correlates to organizational boundaries and diverging of commonality of infrastructure, management, and operations.

Figure 1 summarizes how reach is defined using organizational boundaries as the metric. As the distance increases, the challenges to achieving the goals of deploying, managing, and updating increase.

The metric of complexity describes the *range* of sophistication of the business process. Complexity is relevant to resources that are orchestrated by a business process. Consider when a business process starts, stops, tries to undo, and is long running.

Transactional support enables activities to be orchestrated in a coordinated fashion. With transactional support, units of work become complex:

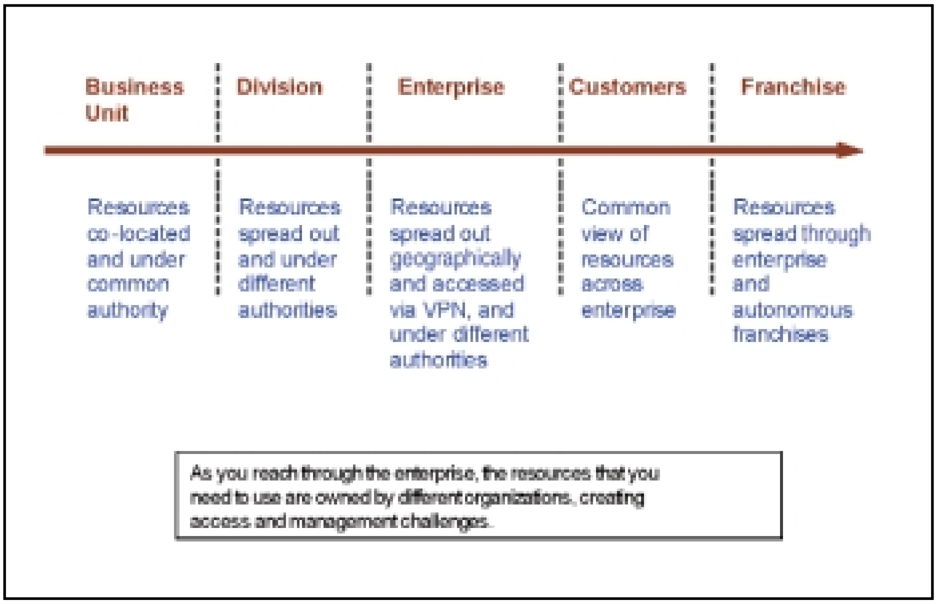
- What is the unit of work?
- Can it be undone?
- How long running?

Figure 2 summarizes how range is defined using transactional complexity as the metric. As the complexity increases, the challenges to completing units of work increase.

Messaging is the simplest level of complexity for business process implementation. The transactional complexity is low, since sending and responding to messages are separate transactions. The business process is expressed in separate pieces of code.

Content-based message routing is the next level of complexity. The overall process is stateless, but messages have guaranteed delivery for recoverability. The core content of these messages is considered to be documents.

BPM is the next level of complexity. The "Ten Pillars of Business Process Management" (McDaniel) is an excellent summary of BPM tools. A BPM tool executes a business process described in a model. As the BPM tool executes the



Business Unit	Division	Enterprise	Customers	Franchise
Resources co-located and under common authority	Resources spread out and under different authorities	Resources spread out geographically and accessed via VPN, and under different authorities	Common view of resources across enterprise	Resources spread through enterprise and autonomous franchises

As you reach through the enterprise, the resources that you need to use are owned by different organizations, creating access and management challenges.

FIGURE 1 Reach defined

instance of the process, and records progress, we have state and can recover from failures. A BPM tool can execute long-running business processes, and is capable of rich exception handling.

We measure the progress of business processes by collecting business events. For messaging and content-based implementations, all of the business events have to be correlated. For BPM, the business events are naturally correlated to the instance of business process.

The highest level of complexity requires a combined view of business event and business object data. This is a powerful view of the operations of the enterprise, revealing hotspots and trends, and suggests how to optimize your business processes. This is *awareness*, which feeds suggested optimizations back into business processes.

Why Are There Data and Applications Everywhere?

An application encapsulates domain-specific procedures and policies. Each application is a subset of what an organization needs since no single product has all domain knowledge. Each application saves state, and remembers data by using a related database. Thus, we have multiple databases for multiple applications for reasons of domain knowledge. Within an enterprise, people with domain knowledge cluster into departments and business units, roughly along the lines of operational responsibility and authority to change.

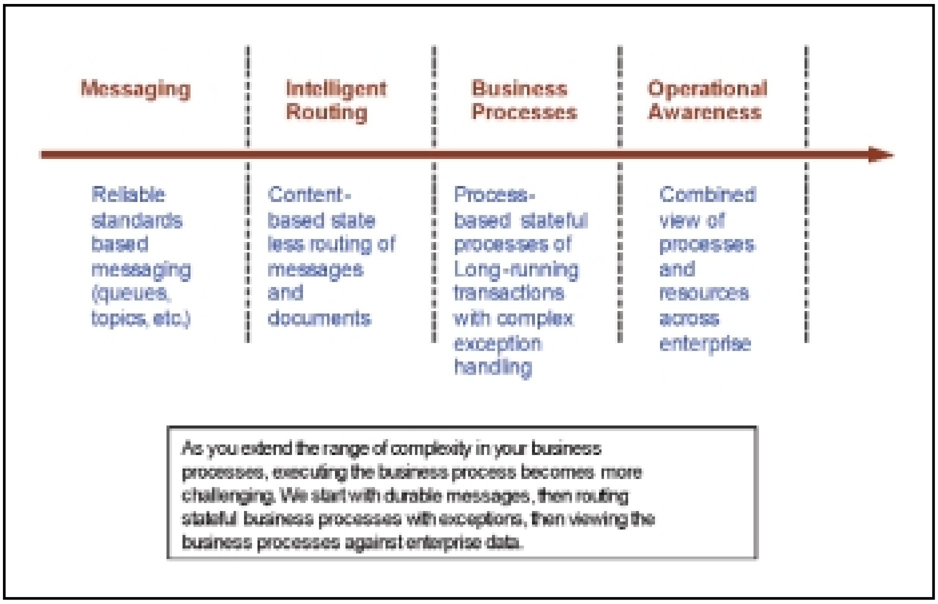
A database supports transactions with ACID properties: atomic, consistent, isolated, and durable. An ACID transaction requires locking relevant and related data for the duration of the transaction. The wider the *reach* of a transaction, the more database resources are locked. This limits the scope of databases, to avoid locking all of our data all the time, for each transaction.

Thus, an enterprise uses many applications and databases. The databases may be able to participate in distributed ACID transactions, but the degree of locking and resource contention will restrict this on an enterprise scale.

Business processes need a different approach for transactions. Business processes span the enterprise, are long-running, and frequently have complex exception handling. A business process is a series of activities to update applications and databases, such that:

- Forward progress is always made
- Appropriate systems of record are updated

We will always have islands of domain knowledge embedded in applications and databases. For an organization to create enterprise-wide business processes, these islands have to be integrated and participate in orchestration.


Messaging
Intelligent Routing
Business Processes
Operational Awareness
Reliable standards based messaging (queues, topics, etc.)
Content-based state less routing of messages and documents
Process-based stateful processes of Long-running transactions with complex exception handling
Combined view of processes and resources across enterprise

As you extend the range of complexity in your business processes, executing the business process becomes more challenging. We start with durable messages, then routing stateful business processes with exceptions, then viewing the business processes against enterprise data.

FIGURE 2 **Range defined**

Use Cases

Two typical use cases for applying BPM in your business are to orchestrate:

- Business processes that reach across your enterprise, including franchises
- Business processes that reach out to your partners

> “Business processes reach across enterprises and partners, and require a range of complex functions”

In the first use case, there are many resources (applications, databases, and people) to integrate and orchestrate. These resources are managed by different organizations, and are typically “paper-driven.” There are usually two high-level, dominant business processes, CreateProduct and SellProduct. The implementations reach all resources of the enterprise, and pose significant challenges in negotiating agreements between departments and business units of a large enterprise.

In the second case, there are fewer resources to integrate and orchestrate. However, your degree of control over how a partner interfaces to your enterprise is low, and these resources range in sophistication, from FileTransfer to WebServices. The business processes between partner and enterprise collaborate with the enterprise’s CreateProduct or SellProduct business process.

An enterprise BPM tool *executes* a business process across a wide reach and range. We *measure* to remove operational blindspots, and determine business process improvements. And we *change* a business process to adopt improvements, and to leverage high-change areas of the business. This is the foundation of *agility*. Next, we’ll see how reach and range impact the ability to execute, measure, and change a business process.

Reach and Range Impact

Reach and range require complex functions to keep your business processes agile across your enterprise, partners, and ecosystem.

When a business process *executes*, it depends on many complex functions to work over the entire reach and range:

- ***Addressability:*** Required for even the smallest reach and simplest range of complexity. There are challenges, such as namespaces, for resources that are far apart and hosted on disparate servers. For some integration techniques, such as clustered app servers, namespaces pose a significant challenge.
- ***Messaging:*** Base requirement for low range of complexity. Message-oriented middleware (MOM) provides asynchronous messages and durability, as well as Quality-of-Service semantics of "delivered at most once," "delivered exactly once," "delivered at least once."
- ***Enterprise and Web services:*** Standards-based service-oriented architecture avoids point-to-point proprietary connections. Service-oriented standards leverage XML for a common form of message content.
- ***Transformation:*** This is a fundamental function. It allows two services to communicate even if they speak two dialects of XML.
- ***Security:*** A service must be secured from use by unintended users; and more important, allow appropriate users to use it.

When the business process is *measured* it depends on this complex function to work over the entire reach and range:

- ***XML data collection:*** Collect business events of your business processes in one place to enable XQuery reports against it. To achieve the most sophisticated range, you need to correlate your business events against enterprise business object data that is throughout the reach of your enterprise. This is a prerequisite to achieving operational awareness.

> "Reach and range require complex functions to keep your business processes agile across your enterprise, partners, and eco-system"

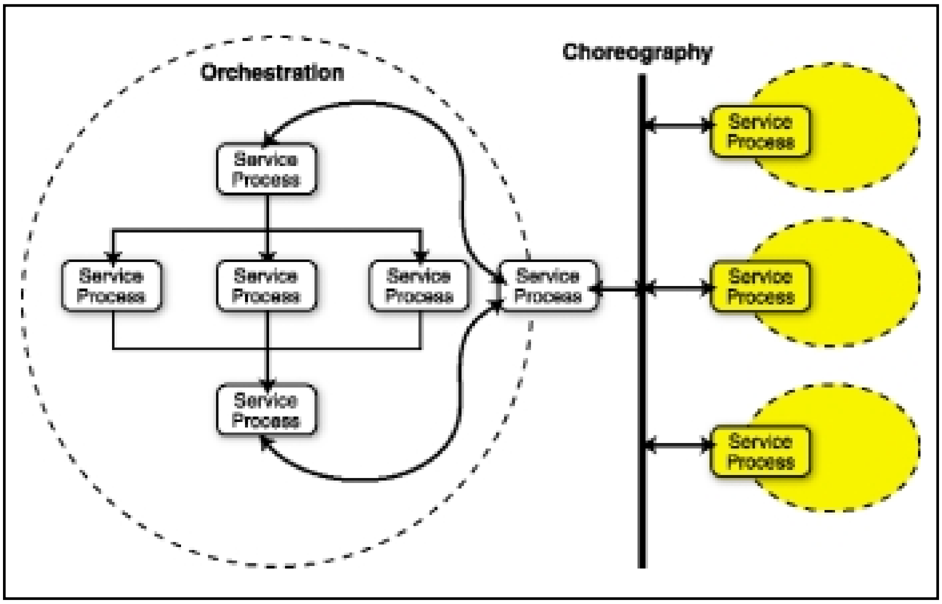


FIGURE 3 Business process as service (Adapted with permission from ZapThink)

When the business process is *changed*, it depends on many complex functions to work over the entire reach and range:

- ***Transparent management:*** Implement once and manage forever. Your ability to manage orchestrated resources may be limited to subsets or clusters of resources, where you are forced to manually coordinate separate management efforts. Some enterprise integration software is limited in this capability due to clustering limitations. In particular, consider the investment a typical enterprise makes in these management activities:
  - *Configuration:* If anything changes, configuration must happen. There is a big difference in operations between managing configuration from one point rather than manually coordinating several acts of configuration.
  - *Deployment:* The promise of business processes is that they can be measured, optimized, and redeployed. A large enterprise is likely to have multiple implementations of business processes. Coordinating rollouts, tests, and rollbacks of business process implementations is onerous without transparent management.

Reach and range of orchestrated resources in a confined area such as a Web service is easy. Reach and range over the enterprise and ecosystem, with multiple owners of systems, is hard. The reach of crossing over multiple organizations geometrically increases the difficulty of the range of process sophistication. The difficulty is similar to doing business between companies in different countries where there is a lack of trading agreements. With differing legal and monetary structures, there is much manual intervention, translation is imperfect, and you hope for the best.

The BPM tool is not the best place to implement reach and range functions. A business process implementation requires the reach and range functions above, but the appropriate division of labor is to put reach and range functions into an integration network. The work of orchestrating resources in a business process becomes simple, and the promise of reuse is realized.

The Integration Network

The contemporary integration network is best typified by the enterprise service bus.

In its 2003 Predictions series, industry analyst firm Gartner, Inc., said:

> *A new category of integration middleware called the enterprise service bus (ESB) has emerged to support the proliferation of service-oriented interactions between enterprise applications. An ESB is a standards-based integration backbone that combines messaging, Web services, transformation and intelligent routing to reliably connect and coordinate the interaction of hundreds of application endpoints spanning a global organization.*

In the report, Gartner predicts that "a majority of large enterprises will have an ESB running by YE05."

Using an ESB leverages its inherent reach and range functions, saving implementation and management costs. The ESB is a new product concept, similar in concept to the modern office building. A person arrives at work, goes to their room. In their room is fresh air, desk, phone, data, and so on. When one person talks to another, they don't install a custom phone and new wires to the people they expect to talk to. They just pick up the phone, dial a standard extension number, and talk to anyone.

Prior to the ESB, resources were manually glued together one by one, by plugging them into either an integration broker hub or an application server hub. If those hubs themselves need to integrate, the process is repeated, yielding a brittle hierarchical structure.

Using an ESB, the implementation of the business process is minimal, saving on development and QA costs. This is true whether the business process is basic, as with messaging, or sophisticated, as with a BPM tool.

The ESB approach is the cleanest way to implement a business process as a service, as described in ZapThink's April 2003 *Service Oriented Process*. Your inventory of applications and databases are your atomic services. Business processes are built, and encapsulated as services, and installed on the ESB as a resource. Top-level business processes, such as CreateProduct, are then composed of sub-processes, perhaps DesignProduct, ProcureParts, and BuildProduct. Each of these sub-processes can use the atomic resources, or be further decomposed.

The end result is a clean hierarchical grouping and usage of business processes, as services, without the redundant and confusing technology of prior approaches. In Figure 3 we see one service on the left, which is transparent, and composed of other services which are orchestrated by BPM, all on the ESB. This larger process is itself encapsulated as a service and is available on the ESB, within the reach of the enterprise. Further, some processes are opaque, as we see on the right of Figure 3. This can be the situation when integrating with partners. In this situation the interactions between services is via collaboration, for example as expressed in ebXML.

> "The ESB approach is the cleanest way to implement a business process as a service"

Using BPM in an ESB

The ESB provides all reach functions, and the basic range functions of messaging and content-based routing. The BPM tool must provide facilities to:

- ***Model a business process:*** In an ESB context, modeling is simplified, since the BPM tool only needs to know a resource's service name and its API messages. The BPM tool uses the ESB to:
  - Connect to a service
  - Transform each service XML to common form
- ***Execute the business process:*** In an ESB context, execution is simplified because the BPM tool can uniformly interact with any service on the ESB. An executing business process can be encapsulated as a service on the ESB. Thus, a business process can be composed of other business processes, all without additional plumbing or coding.
- ***Monitor the business process:*** In an ESB context, we capture business events into a business event repository service to be queried anywhere on the ESB.

Can we use multiple BPM tools in an ESB? Yes! By factoring out the ESB from the BPM reach and range functions, we encapsulate executing business processes as services deployed on the ESB. The actual BPM tool that executes the business process is hidden. We can have one or several BPM tools spread out over the enterprise. We can have BPM tools from different vendors, and different BPM tools from the same vendor.

The ESB highlights the difference between the two use cases. In the first use case the ESB spans the enterprise, including franchises. Spanning franchises is attractive because of the inherently cost-effective nature of the ESB. In the second use case we can't assume tight ESB integration to the Partner, so we use collaboration, as defined, for example, in the ebXML standard. This is a radically different style of business process and may require a different BPM tool than the one used within the enterprise.

Summary

To summarize, when implementing a business process ask:

- What is the reach of the business process?
- What is the range of complexity of the business process?
- Are we ready to factor out the reach and range complexity of your processes, into an ESB?

Then implement the business process or processes that are right for you!

- Within your enterprise
- To your partners
- Decompose into smaller business sub-processes and encapsulate as a service?

References

- Zapthink, LLC. Service Oriented Process. Report #ZTR-WS108, April 14, 2003.
- McDaniel, Tyler, "Ten Pillars of Business Process Management," eAI Journal, November 2001: 30-34.
- Gartner, Inc. "Predicts 2003: Enterprise Service Buses Emerge." December 9, 2002 (DF-18-7304)

International Web Services Conference & Expo

Web Services Edge WEST 2003

web services EDGE conference & expo

SEPT. 30 - OCT. 2, 2003

Santa Clara, CA

EXTENDING THE ENTERPRISE WITH WEB SERVICES THROUGH JAVA, .NET, WEBSPHERE, MAC OS X AND XML TECHNOLOGIES

Featured technologies and topics will include:

- Focus on XML
- Focus on Web Services
- Focus on Java
- Focus on .NET
- Focus on Mac OS X

XML

WebSphere

Microsoft .net

Mac OS X

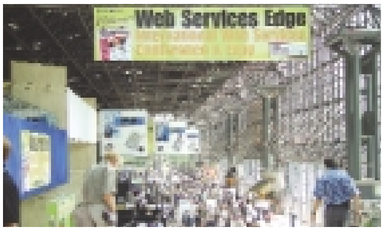
COMING SOON!

LINUX EDGE conference & expo

web services EDGE conference & expo

BOSTON

February 24-27, 2004



Over 100 participating companies will display and demonstrate over 300 developer products and solutions.

Over 2,000 Systems Integrators, System Architects, Developers, and Project Managers will attend the conference expo.

Over 60 of the latest sessions on training, certifications, seminars, case studies, and panel discussions will deliver REAL World benefits, the industry pulse and proven strategies.

Contact information: U.S. Events: 201 802-3069 or e-mail grisha@sys-con.com

Java is a registered trademark of Sun Microsystems, .NET is a registered trademark of Microsoft, Mac OS X is a registered trademark of Apple Computer, Inc., WebSphere is a registered trademark of IBM. All other product names herein are the properties of their respective companies.

Web Services Edge WEST 2003

WEB SERVICES EDGE CONFERENCE & EDUCATION

web services EDGE conference & expo

SEPT. 30 - OCT. 2, 2003

Santa Clara, CA

.NET TECHNOLOGY

Presentations will explore the Microsoft .NET platform for Web services. To the average developer, it represents an entirely new approach to creating software for the Microsoft platform. What's more, .NET development products - such as Visual Studio .NET - now bring the power of drag-and-drop, GUI-based programming to such diverse platforms as the Web and mobile devices.

Sessions will focus on:

- ASP.NET
- Security
- VB.NET
- .NET and XML
- Smart Device Extensions for VS.NET
- Best Practices
- Shared Source CLI
- .NET Remoting
- Smart Devices in Health Care Settings
- Mobile Internet Toolkit
- ROTOR
- Portable .NET
- ASP.NET Using Mono
- Using WSE with IBM's WSTK
- GUI applications Using Mono
- Portals – Windows Sharepoint Services/Sharepoint Portal Server
- Windows Server 2003 and IIS 6
- .NET and Java Interoperability
- Distributed .NET for Financial Applications
- Developing C# with Eclipse

Microsoft .net

JAVA TECHNOLOGY

The Java Track features presentations aimed at the beginner, as well as the seasoned Java developer. Sessions will explore the whole spectrum of Java, focusing on J2EE, application architecture, EJB & J2ME. In addition the Track will cover the latest in SWT, Ant, JUnit, open source frameworks, as well as an in-depth look into the vital role that Java is playing in building and deploying Web services.

Sessions will focus on:

- Enterprise Java 1.4
- Ant Applied in "Real World" Web Services
- Developing Application Frameworks w/SWT
- Empowering Java and RSS for Blogging
- JUnit: Testing your Java w/JUnit
- JDK1.5: The Tiger
- Simplifying J2EE Applications
- Using IBM's Emerging Technologies Toolkit (ETTK)
- Apache Axis
- Meeting the Challenges of J2ME Development
- Integrating Java + .NET
- Squeezing Java

WEB SERVICES TECHNOLOGY

Presentations will include discussions of security, interoperability, the role of UDDI, progress of the standards-making bodies, SOAP, and BPM. Case studies cover the design and deployment of Web services in the marketplace.

Sessions will focus on:

- Interoperability
- Enterprise Networks
- Web Services Management
- Web Services Standards
- Web Services Orchestration
- Security (WS-Security, SAML)
- BPEL4WS
- UDDI: Dead or Alive?
- ebXML & Web Services
- EAI & Web Services
- RPC vs. Messaging: Uses and Differences
- User Interfaces for Web Services
- Web Services Best Practices
- Service Oriented Architecture

MAC OS X

OS X represents a new wave of operating systems. It combines the ease of use of a Mac with the power of Unix. Sessions in this track will highlight the use of the Mac OS X platform in applications and Web services development, deployment and management.

Sessions will focus on:

- Introducing OS X (Panther): What's New?
- Quick Applications using AppleScript
- Enterprise Java and OS X
- Developing Web Services Using WebObjects
- Xserve: Ease of OS X and Power of Unix
- Introducing Quartz: 2D Graphics for Apple
- OS X for the Unix Developer
- Securing OS X Applications
- Java and OS X: A Perfect Marriage
- Programming Rich User Interfaces Using Cocoa

X Mac OS X

XML TECHNOLOGY

Presentations will focus on the various facets of XML technologies as they are applied to solving business computing problems. Sessions will include emerging standards in XML Schemas, XML repositories, industry applications of XML, applying XML for building Web services applications, XML/XSLT/XQuery-based programming using Java/.NET, XML databases, XML tools and servers, XML-based messaging, and the issues related to applying XML in B2B/EAI applications. The XML Track is geared for audiences ranging from beginners to system architects and advanced developers.

XML

Sessions will focus on:

- XML Standards & Vocabularies
- Introduction to XForms
- Securing Your XML and Web Services Infrastructure
- XQuery Fundamentals: Key Ingredient to Enterprise Information Integration
- XML and Enterprise Architecture: Technology Trends
- Standards-Based Enterprise Middleware Using XML/Web Services
- XML and Financial Services
- Canonical Documents for Your Business: Design Strategies
- XPath/XSLT 2.0: What's New?
- XML Schema Best Practices
- XML in EAI, Enterprise Portals, Content Management

For more information visit
www.sys-con.com
or call
201 802-3069

PRODUCED BY SYS-CON EVENTS

Reviewed by Joseph A. Mitchko

Mindreef SOAPscope 1.0

Bring SOAP protocol into view with handy diagnostic tool

Author Bio:
Joe Mitchko is a technology specialist working for a leading consulting company and is product review editor and a contributing writer for Web Services Journal.
JMITCHKO@PATMEDIA.NET

While carefully sorting out junk mail in my inbox late one evening, I came across something that caught my attention. First of all, as product review editor, it is my duty and responsibility to give the subject line of any e-mail a quick once-over for Web service buzz words before I give it a heave-ho into the virtual trash bin. But this unsolicited parcel had something about it that made it a bit different than the others. Something about Web services, diagnostic tool, easy to use, please give it a try. So far, so good. In addition, an honest-to-goodness permanent license is included for free, along with a simple promise that you will like what you see. Okay, now you're talking! But it was the simplicity and genuineness of the e-mail that really inspired me to hit the download button and give it a try. The message regarding Mindreef SOAPscope basically lived up to its promises and more.

Key Features

Mindreef SOAPscope 1.0 is a Web services diagnostic tool, designed to provide toolkit-independent logging and monitoring of SOAP network traffic. SOAPscope is composed of two components, a network sniffer and a browser-based message viewer (see Figure 1). The sniffer component is designed to capture SOAP request and response messages within the HTTP protocol traffic and persist the information to an embedded relational database. The message viewer component is a browser-based Web application that allows a user to view the persisted SOAP request and response messages and more. Since it is browser-based, the viewer opens the door for remote and collaborative debugging sessions.

The SOAPscope viewer provides a pseudocode and XML view of message details, and two ways to monitor SOAP traffic – log view or live view. The log view provides message history and search capabilities while the live view allows for real-time debugging. In addition, a handy WSDL viewer allows you to punch in a WSDL URL and view it in either native XML or in pseudocode mode.

Some of the more advanced features of the tool allow you to modify and resend previously captured SOAP requests – handy for on-the-fly debugging.

Mindreef
web service diagnostics

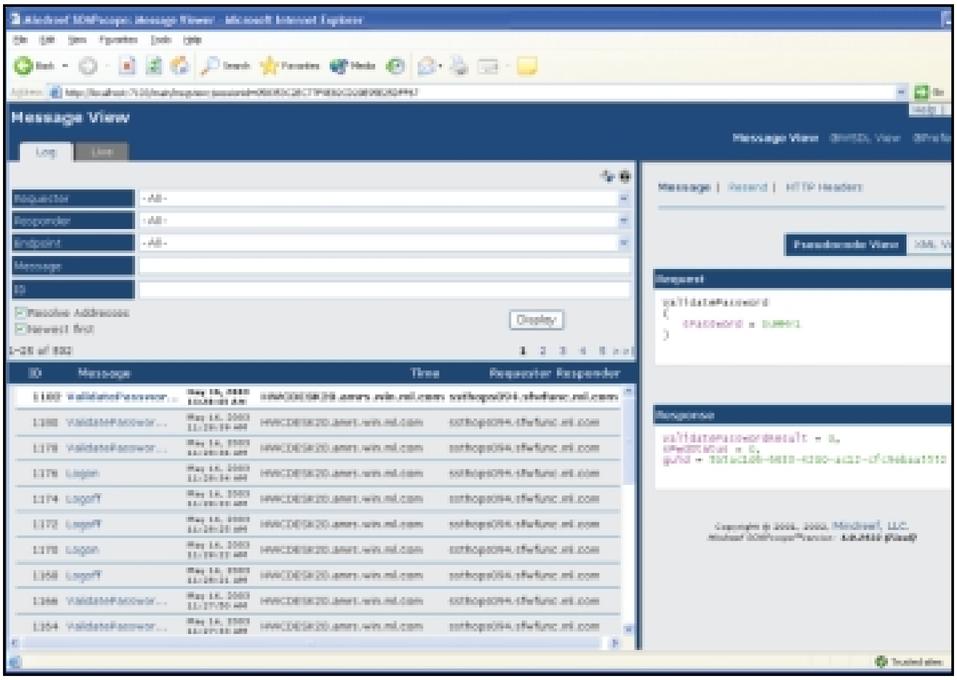
COMPANY INFO
Mindreef, Inc.
22 Proctor Hill Road
Hollis, NH 03049
Tel: 603 465-2204
Fax: 603 465-6583
Web: www.mindreef.com
E-mail: mrsoapscope@mindreef.com

DOWNLOAD INFORMATION
Web site: www.mindreef.com/order/eval.php

LICENSING INFORMATION
$99

TESTING ENVIRONMENT
OS: Windows-XP
Hardware: 1GHz Athlon, 1GB RAM

Web Services Journal MAGAZINE WSJ World Class PRODUCT AWARD



ID	Message	Time	Requester	Responder
1182	ValidatePasswor...	May 14, 2003 [illegible] AM	HWCDESK20.amer.win.ml.com	sothops094.sfwfunc.ml.com
1180	ValidatePasswor...	May 14, 2003 [illegible] AM	HWCDESK20.amer.win.ml.com	sothops094.sfwfunc.ml.com
1178	ValidatePasswor...	May 14, 2003 [illegible] AM	HWCDESK20.amer.win.ml.com	sothops094.sfwfunc.ml.com
1176	Logon	May 14, 2003 [illegible] AM	HWCDESK20.amer.win.ml.com	sothops094.sfwfunc.ml.com
1174	Logoff	May 14, 2003 [illegible] AM	HWCDESK20.amer.win.ml.com	sothops094.sfwfunc.ml.com
1172	Logoff	May 14, 2003 [illegible] AM	HWCDESK20.amer.win.ml.com	sothops094.sfwfunc.ml.com
1170	Logon	May 14, 2003 [illegible] AM	HWCDESK20.amer.win.ml.com	sothops094.sfwfunc.ml.com
1168	Logoff	May 14, 2003 [illegible] AM	HWCDESK20.amer.win.ml.com	sothops094.sfwfunc.ml.com
1166	ValidatePasswor...	May 14, 2003 [illegible] AM	HWCDESK20.amer.win.ml.com	sothops094.sfwfunc.ml.com
1164	ValidatePasswor...	May 14, 2003 [illegible] AM	HWCDESK20.amer.win.ml.com	sothops094.sfwfunc.ml.com

```
validatePassword
(
   sPassword = [illegible]
)
```

```
validatePasswordResult = 0,
[illegible] = 0,
guid = [illegible]
```

Copyright © 2002, 2003, Mindreef, LLC.
Mindreef SOAPscope Version: [illegible]

FIGURE 1 The Mindreef SOAPscope Message Viewer

Sniffer Component

The SOAPscope HTTP sniffer-interceptor is responsible for monitoring a network interface for SOAP-related traffic, relaying that information to an open message viewer, and logging the information in an internal embedded database. The sniffer interceptor executes in a DOS window on the desktop, where on startup it fires up an embedded database and an Apache Tomcat server (for hosting the message viewer application). Then, of course, it starts sniffing.

Preferences in the viewer allow you to choose a specific network adapter for multihomed machines and various default settings. In cases where you cannot install the network sniffer on a target machine or in cases where client and Web services reside on the same machine, SOAPscope includes support for network proxy services. In this case, the proxy will consume the SOAP request, log the request, and forward the message to the true endpoint.

The installation also includes support for .NET proxy services that includes a .NET assembly that can be configured to instruct the client Web reference to reference a proxy endpoint. Those that are really into this kind of stuff will further appreciate the "ssconfig" utility that comes with SOAPscope. It helps you set up and configure proxy chaining and port forwarding.

Message Viewer

As I mentioned earlier, the message viewer is browser based and served up at port 7103 by the sniffer-interceptor service. Once started, you choose to view live SOAP traffic or search through previously logged SOAP messages by selecting the appropriate tab in the message viewer.

Live view provides real-time monitoring of SOAP transactions. You can just keep the viewer open on the desktop and watch the SOAP protocol activity fly by. Errors are easily spotted with a red "X" next to the SOAP transaction entry, while successful transactions are given a green check mark. You can drill down for more detailed information at any time simply by clicking the SOAP transaction line.

The log view, on the other hand, allows you to search and view message history. You can filter SOAP messages according to requester, responder, endpoint, message, and ID. This is great for searching through past history in search of Web service errors that have occurred under certain conditions.

> "It's not often that you find a tool that is so well thought out and designed"

You have your choice of viewing the SOAP request and response messages as straight XML or in pseudocode mode. In the pseudocode mode, the request and response messages are represented in an easy-to-read, 3GL-like format. Most of the XML "noise", including headers and namespaces, is taken out.

Another great feature is the HTTP header viewer. This tracks "out-of-band" header data processed by the HTTP protocol during SOAP transactions. To further help in diagnosing a problem, SOAPscope provides a very useful WSDL viewer. Again, you can view the Web service in SOAP XML (straight XML) or in pseudocode mode. And if you really want to get fancy, you can embed (in HTTP headers) service-based information that is not readily available in the SOAP response message by including debugging annotations in your Web service calls.

Installation

Windows 2000/XMP installation is essentially a no-brainer, and in no time you're viewing SOAP messages. The installer was able to identify the network card on my workstation and configure the sniffer to use it. Unless you have special network proxy needs or other network interface issues, it is essentially ready to run after installation.

Test Drive

While doing some serious Web service testing on a project I'm currently working on (I do other things besides writing in my spare time), I had the opportunity to give SOAPscope a good workout. One particular test bed, the mother of all client programs, references literally dozens of Web service calls while maintaining session and transaction state. SOAPscope for the most part accurately logged SOAP-related activity, but in one particular situation the live view would freeze up due to a "long-running" SOAP transaction – more like it doesn't respond at all. In order to get around it, I had to shut down the sniffer service and restart. I was finally able to narrow in on the problem (a UDDI thing), but I think that the sniffer could have handled it a bit better by using a timeout.

I found the viewer's user interface to be very clean, easy to read, and relatively uncluttered. The information displayed was basically accurate and bug free. In addition, both the XML and pseudocode views have color-coded text, making it easy to see SOAP-specific tags, namespace information, and message request and response content.

All SOAP message content and log information is stored in an embedded database. Although it is basically transparent, you will need to do a little database management in order to purge or back up the database. Nothing in the way of log maintenance is provided in SOAPscope for this release. Luckily, database maintenance instructions are included in the documentation and are relatively easy to follow.

Conclusion

It's not often that you find a tool that is so well thought out and designed. SOAPscope really lives up to its promises of being "here to help." Equally versed for development and testing situations, once you start using it, you will never put it down. The amount of functionality provided is just right, neither overloading the GUI with seldom-used features nor leaving you to find some other diagnostic tool because it doesn't do enough. Like a Swiss Army Knife, it is a trusted tool that you will use again and again. ⓔ

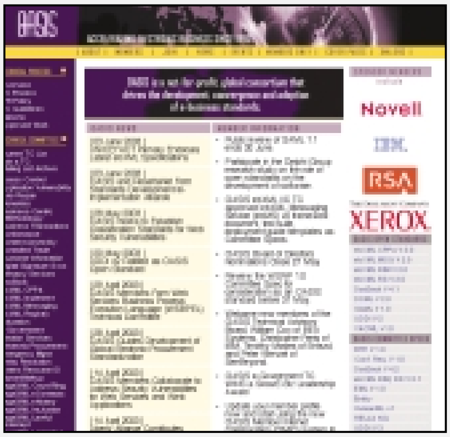
WSJ News

OASIS and RosettaNet Form Alliance

(Boston and Santa Ana, CA) – OASIS and RosettaNet, the industry standards consortia, have formalized plans for a coordinated approach to standards development and implementation that will streamline business-to-business (B2B) integration practices for global supply chain companies. The new alliance leverages the supply chain expertise of RosettaNet with the broad, interoperability focus of OASIS.

The organizations have established respective liaison memberships that allow representatives of each consortium to actively participate in the technical work of the other.

www.oasis-open.org, www.rosettanet.org



W3C Approves Patent Policy

The World Wide Web Consortium has approved the W3C Patent Policy. Written by the Patent Policy Working Group, it received more support from the membership than any recommendation in recent history. The policy encourages royalty-free Web standards and governs the handling of patents in the process of producing and implementing W3C Recommendations.

www.w3.org

Intalio Enters Alliance with CSC

(San Mateo, CA) – Intalio, Inc., the business process management company, has announced an alliance with Computer Sciences Corporation, a leading information technology services company. The alliance integrates Intalio|n^3 2.0, Intalio's Business Process Management System (BPMS), with CSC's e4 enterprise architecture. The two companies will jointly pursue the design, development, and delivery of comprehensive BPM solutions for their customers.

Intalio|n^3 is a standards-based, platform-neutral BPMS that supports the design, deployment, execution, maintenance, and optimization of business processes that involve distributed transactions with packaged applications, databases, and heritage systems, as well as complex workflow interactions with end users.

www.csc.com, www.intalio.com.

Service Integrity Unveils SIFT

(Newton, MA) – Service Integrity Inc., a developer of Web services monitoring and analysis software, has announced the availability of SIFT, their Web services management solution. SIFT is designed to reduce the risks and costs of operating production Web services and enhance business performance.

SIFT provides monitoring and analysis capabilities that enable companies to isolate and resolve production problems, ultimately reducing downtime. The technology also provides real-time visibility into any business transaction using a Web service.

www.serviceintegrity.com

Framework Helps Companies Connect C++ to the Enterprise

(Boulder, CO) – Rogue Wave Software has launched a new framework for C++ applications to access Microsoft .NET-connected software.

The Lightweight Enterprise Integration Framework, or LEIF, is a structured set of tools with which developers can expose applications created in C++ to Web services and to the functionality of software developed to run on the Microsoft .NET Framework.

www.roguewave.com/LEIFrw01

Sforce Adopted for Software-As-Service Development

(New York) – Salesforce.com has launched sforce, the salesforce.com client/service application development utility, and announced more than 25 founding developers, including BEA Systems, Borland, Microsoft, and Sun Microsystems, that are pioneering its use to create new software-as-service applications.

Using the same utility computing model as salesforce.com, sforce helps developers build business applications directly over the Internet. It will enable enterprises to drive sophisticated business processes as a Web-based service for a monthly per-user fee.

www.sforce.com

Factiva Enhances Its Web Service API

(New York) – Factiva has announced the release of new features for its Web service API, enabling developers to take advantage of automated programming tools to develop and deploy new content applications quickly and cost-effectively.

Factiva Developer's Kit offers a second-generation XML-based application-programming interface (API) that provides developers with flexible tools to easily integrate new content into business applications. This release includes Factiva Developer's Kit transactions that provide developers with access to additional content and functionality in the Factiva Platform. The addition of WSDL allows developers to use widely available Web service tools to add a high level of automation to the production of programming objects.

www.factiva.com/developerskit

WSJ ADVERTISER INDEX

ADVERTISER	URL	PHONE	PAGE
Altova	http://xmlj.altova.com/code		19
Choreology	www.choreology.com		8, 9
Crystal Decisions	www.crystaldecisions.com/cr9/220	800-877-2340	27
Ektron	www.ektron.com/ws		59
IBM	www.ibm.com/websphere/seeit		60
JDJ Store	www.jdjstore.com	888-303-5282	39
LinuxWorld Conference & Expo	www.linuxworldexpo.com		46, 47
LinuxWorld Magazine	www.sys-con.com	201-802-3020	53
Macromedia	www.macromedia.com/go/certification1/		23
Merant	www.merant.com	800-547-7827	29
Mindreef	www.mindreef.com	603-465-2204	5
Mindreef News	www.mindreef.com/nl0307		4
Parasoft	www.parasoft.com/wsj07	888-305-0041	2
Quest Software	http://java.quest.com/jcsc/ws	800-663-4723	33
Quest Software	www//java.quest.com/jcsr/ws	800-663-4723	15
Sonic Software	www.sonicsoftware.com		11
SYS-CON Media	www.sys-con.com/suboffer.cfm	888-303-5282	46, 47
SYS-CON Media	www.sys-con.com	888-303-5282	41
Web Services Edge West 2003	www.sys-con.com	201-802-3069	51
Web Services Journal	www.wsj2.com	888-303-5282	43

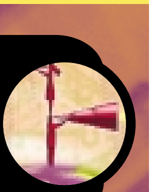
Advertiser is fully responsible for all financial liability and terms of the contract executed by their agents or agencies who are acting on behalf of the advertiser. This index is provided as an additional service to our readers. The publisher does not assume any liability for errors or omissions.

IN THE NEXT ISSUE OF WSJ...

Focus: Grid Computing

Introducing Open Grid Services

The Grid community is building its infrastructure on top of Web services technologies, making the Grid a big user of the emerging standards and an excellent evaluation platform for all the specifications and tools.



Grid Services Extend Web Services

A Grid service is a Web service that conforms to a particular set of conventions. Grid services marry important concepts from the Grid computing community with Web services.

Plus:

Infrastructure-Level Web Services

Infrastructure-level Web services are Web services that implement part of the distributed computing infrastructure - and can solve a huge number of problems.

Introducing BPEL4WS 1.0

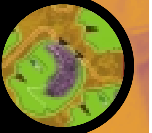
The ultimate goal of business process languages like BPEL4WS is to abstract underlying Web services so that the business process language effectively becomes the Web services API.

The Change Management Balancing Act

Done right, change management reduces complexity and maximizes efficiency. Done wrong, change mismanagement can bust budgets, trash schedules, and undermine team efforts.

Building a Business Logic Layer Over Multiple Web Services

Leverage your multiple Web services to build a truly distributed Web services architecture.



WebServices .NET J2EE XML JOURNAL

Unleashing the Power of BPM Through Web Services

Alejandro Danylyszyn & Cesare Rotundo

Alejandro Danylyszyn is a senior manager in Deloitte Consulting's Technology Integration practice. He has worked for over 15 years as a consultant to large high-technology manufacturing, telecommunication carriers, and financial services companies in the areas of strategy, operations/process improvement and solution design/implementation, with a focus in systems integration, enterprise portals and web services.
ADANYLYSZYN@DC.COM

Cesare Rotundo is a senior manager with Deloitte Consulting. He has expertise in the enterprise IT infrastructure, particularly middleware such as EAI, B2Bi, enterprise portals, Web services, and J2EE, with a specific focus on real-time business integration. He has focused in the last year on building enterprise integration solutions around CRM.
CROTUNDO@DC.COM

Long touted as a revolutionary technology, Web services promise to make IT infrastructures obsolete. They enable the free flow of information across systems, lowering the cost and complexity of integration across entire enterprises.

They have moved to the forefront of IT and business executives' minds as an increased focus on real-time transaction processing, collaborative manufacturing, and e-commerce drive a strong demand for external integration (B2B) and Business Process Management (BPM). After all, the success of any business rests with process management since companies realize that broken or badly managed processes drain resources and profits.

BPM solutions define structured, efficient and easily reconfigurable processes to support a company's operations. Optimally, they allow an organization to change its processes quickly when conditions shift in the business environment. Bottom line: BPM solutions make a company "nimble" to adapt quickly and gain – or maintain – competitive advantages.

Enterprise application integration (EAI) has evolved from its traditional role as "middleware" to an advanced form of business integration. It is at an evolutionary point that promises to create a new breed of business integration, one where disparate platforms and heterogeneous environments work together. It focuses on executing new and enhanced business processes within the integration software.

Web services could offer the key to unleashing the power of BPM. They enhance its benefits by providing lower integration costs, accelerating the time to market, through shorter implementation cycles, and increasing the ability to adapt (e.g., dynamic binding of process steps.)

Now that companies can adapt more quickly through this next generation of BPM-Web services, many competitive advantages emerge. In *The Power of Corporate Kinetics*, authors Michael Fradette and Steve Michaud suggest that self-adapting, self-renewing, and instant-action enterprises can:

- ***Serve a single customer:*** Provide the sales relationships, products, and services to match the infinitely diverse and changing demands of individual customers, one by one.

- ***Act in zero time:*** Meet demands and exploit market opportunities instantly by means of simultaneous enterprise collaborations and actions.

However, a number of challenges exist that companies must pvercome before they can harness the power of Web services and BPM successfully. For example, Web services lack adequate security. Current standards cover only basic connecting services. Such fundamental integration services as orchestration, universal data definitions, and transaction management, among others, are in their infancy, and it will take time before their completion and adoption.

Therefore, it is important to recognize that Web services do not replace EAI. Rather, they work with it to create a new and improved BPM, thereby moving enterprises toward a higher level of business process integration that is more dynamic and reconfigurable.

Driven by Web services, BPM offers a prime opportunity to create new types of business solutions that were impossible before. Additionally, as Cathleen Benko and F. Warren McFarlan point out in *Connecting the Dots*, enterprises that can adapt quickly (that is to say, align their IT solutions with their strategic and operational objectives) not only survive, but also thrive in the "information frontier" that we live in today.