June 2002 Volume 2 Issue 6

WebServices JOURNAL

.NET J2EE XML

WSJ2.COM

FULL CONFERENCE PROGRAM Pg.50

RETAILERS PLEASE DISPLAY UNTIL JULY 31, 2002
$6.99US $7.99CAN
06
0 71486 03420 9
SYS-CON MEDIA

FOCUS ON BUSINESS PROCESS MANAGEMENT

WebServices .NET J2EE XML JOURNAL

INTERNATIONAL ADVISORY BOARD
JEREMY ALLAIRE, ANDREW ASTOR, STEVE BENFIELD, DAVE CHAPPELL, PHILIP DESAUTELS, GRAHAM GLASS, TYLER JEWELL, PAUL LIPTON, NORBERT MIKULA, FRANK MOSS

TECHNICAL ADVISORY BOARD
JP MORGENTHAL, ANDY ROBERTS, DAVID RUSSELL, AJIT SAGAR, SIMEON SIMEONOV, RICHARD SOLEY

EDITOR-IN-CHIEF: SEAN RHODY
EDITORIAL DIRECTOR: JEREMY GEELAN
INDUSTRY EDITOR: NORBERT MIKULA
PRODUCT REVIEW EDITOR: JOE MITCHKO
.NET EDITOR: DAVE RADER
EXECUTIVE EDITOR: GAIL SCHULTZ
MANAGING EDITOR: CHERYL VAN SISE
SENIOR EDITOR: M'LOU PINKHAM
EDITOR: NANCY VALENTINE
ASSOCIATE EDITORS: JAMIE MATUSOW, JEAN CASSIDY

WRITERS IN THIS ISSUE
STEVE BENFIELD, ASHISH DESHPANDE, ROB CUTLIP, DAN FOODY, PAUL KAISER, MARK PALMER, GEORGE PAOLINI, SEAN RHODY, SANJIT ROY, MARK RUPORT, GUNJAN SAMTANI, DIMPLE SADHWAMI, HOWARD SMITH, RIC TELFORD

PUBLISHER, PRESIDENT, AND CEO: FUAT A. KIRCAALI
CHIEF OPERATING OFFICER: MARK HARABEDIAN
VP, BUSINESS DEVELOPMENT: GRISHA DAVIDA
SENIOR VP, SALES & MARKETING: CARMEN GONZALEZ
VP, PRODUCTION & DESIGN: JIM MORGAN
VP, SALES & MARKETING: MILES SILVERMAN
VP, EVENTS: CATHY WALTERS
CHIEF FINANCIAL OFFICER: BRUCE KANNER
ASSISTANT CONTROLLER: JUDITH CALNAN
ACCOUNTS PAYABLE: JOAN LAROSE
ACCOUNTS RECEIVABLE: JAN BRAIDECH
ACCOUNTING CLERK: BETTY WHITE
ADVERTISING ACCOUNT MANAGERS: MEGAN RING, ROBYN FORMA
ASSOCIATE SALES MANAGERS: CARRIE GEBERT, ALISA CATALANO, KRISTIN KUHNLE, LEAH HITTMANN

CONFERENCE MANAGER: MICHAEL LYNCH
SALES EXECUTIVES, EXHIBITS: MICHAEL PESICK, RICHARD ANDERSON
LEAD DESIGNER: RICHARD SILVERBERG
ART DIRECTOR: ALEX BOTERO
ASSOCIATE ART DIRECTORS: CATHRYN BURAK, LOUIS CUFFARI, AARATHI VENKATARAMAN
ASSISTANT ART DIRECTOR: TAMI BEATTY
WEBMASTER: ROBERT DIAMOND
WEB DESIGNERS: STEPHEN KILMURRAY, CHRISTOPHER CROCE, CATALIN STANCESCO
CONTENT EDITOR: LIN GOETZ
JDJSTORE.COM: ANTHONY D. SPITZER
CUSTOMER SERVICE REPRESENTATIVE: MARGIE DOWNS

SUBSCRIPTIONS
FOR SUBSCRIPTIONS AND REQUESTS FOR BULK ORDERS, PLEASE SEND YOUR LETTERS TO SUBSCRIPTION DEPARTMENT.
SUBSCRIPTION HOTLINE: 888 303-5282
COVER PRICE: $6.99/ISSUE
DOMESTIC: $69.99/YR (12 ISSUES)
CANADA/MEXICO: $99.99/YR OVERSEAS: $129.99/YR (U.S. BANKS OR MONEY ORDERS)
BACK ISSUES: $10 EA., INTERNATIONAL $15 EA
SUBSCRIBE@SYS-CON.COM

EDITORIAL OFFICES
SYS-CON PUBLICATIONS, INC.
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., 135 CHESTNUT RIDGE ROAD, MONTVALE, NJ 07645
PERIODICALS POSTAGE PENDING
POSTMASTER: SEND ADDRESS CHANGES TO:
WEB SERVICES JOURNAL, SYS-CON PUBLICATIONS, INC.
135 CHESTNUT RIDGE ROAD, MONTVALE, NJ 07645

Face the Music

Written by **Sean Rhody**

Author Bio:
Sean Rhody is the editor-in-chief of ***Web Services Journal****. He is a respected industry expert and a consultant with a leading Internet service company.*
SEAN@SYS-CON.COM

Like many IT professionals, I'm an amateur musician. As such, I know how much effort it can be to get a group of people to work together, start together, end together and make a pleasant sort of noise together. And I play in a small group, so I can imagine the coordination effort of something like a symphony orchestra.

This month we're focusing our attention on Business Process Management or, as some call it, Business Process Orchestration (you were wondering where I was going with the opening paragraph, weren't you?). This particular idea has been around for some time in a variety of forms. The premise is that a business has a set of business rules that define how it acts. These rules, or processes, are best captured in systems that allow them to be edited and changed quickly, by business people, not technologists, in response to business conditions. Or so the theory goes.

Workflow was one of the first responses to this theory, and systems that provide workflow automation exist and are in use today. Business Rules engines were another attempt to implement this idea, with a slightly different emphasis – more on rules and grammar, less on process. Regardless of emphasis, this idea has been in play for some time, and many vendors have taken a stab at it.

What makes BPM interesting for Web services is that it provides a portion of the overall picture that is vital to the acceptance of Web services as an operating pardigm. To understand this, picture the use of Web services within an enterprise, as an addition to, or in some cases in place of, an EAI solution. Without a BPM solution, when we create the Web services that encapsulate the internal systems, we are for all intents and purposes creating yet another stovepipe, because we're creating a coded system that deals with the intricacies of our multiple internal systems.

With a BPM capability, we remove the stovepipe by abstracting the logic out of code and into a capability that is integrated into the whole stack of Web services, and intimately aware of them.

Of course, without an easy way to describe business processes, all of this becomes moot. It's one of the uniquely strange truisms of our industry – the more complex a task is, the simpler we have to make a system to describe it in order to be successful.

Our needs regarding BPM are tri-fold. First, we need a language that describes business process well, and meshes with the tools and standards of Web services. Next, we need a studio to design Business Processes in, one that is simple, yet powerful – one that business people can use. Finally, we need an engine that can enact the processes in a distributed, load-balanced fashion, and provide metrics and monitoring.

We actually already have several languages. The Business Process Management Institute (BPMI.ORG) is working on BPML – Business Process Markup Language – a standard that has a number of vendors committed to it. IBM has also put forth a standard – WSFL, or Web Services Flow Language. Interestingly enough, another vendor, Sonic, has already shipped a working implementation of WSFL.

We also have plenty of engines. BEA has an engine in their WebLogic Integration package. IONA, IBM, Sonic, and others also provide engines for the implementation of BPM. Most of these engines are still in the first or second iterations, so while they provide good implementation, they lack some polish, mature metrics and management.

And there are still others building studios. The good thing about a standard language for describing BPM is that a tool maker can use their creative genius to make the tool better, rather than the language. Vendors like Intalio, Proactivity, and RioLabs are concentrating on developing standalone tools that provide for BPM. The big players in the application server arena also provide some slick tools, but with more emphasis on integration to the application server.

Still, BPM in Web services, like the rest of the stack, is an unfinished symphony. We need to settle on a standard language, mature the interfaces, and determine metrics. We also need EAI to be included in the mix, so that Web services represents a truly powerful paradigm for application integration as well as enterprise integration. Which is even more complex. Stay tuned and enjoy the music.

Written by Dan Foody

Accelerating the Adoption of Web Services in Your Enterprise

You've just gone to your CIO with a plan to implement your IT organization's high-profile B2B "Project X" using Web services. Your CIO patiently listens while you explain the benefits of using third-party Web services as part of your mission-critical infrastructure, how contracts will be negotiated electronically without the need for pesky legal departments, how everyone will outsource all their security and management to providers they never need to meet in person, and how applications will dynamically assemble and modify themselves as your needs change.

Start now and have lower risks with long-term advantages

You confidently sit back, smile, and wait for your CIO to give his inevitable nod of approval. It's only natural – after all, this model works just like the electric grid but with software services. After a few moments of thought, your CIO's head begins to nod. But wait, his head is nodding side to side, not up and down! The CIO is rejecting your proposal. How could this happen?

The reality is that adopting an emerging technology like Web services is risky business in the eyes of a CIO. To convince your CIO to adopt Web services, the short-term and long-term rewards have to be clear and the risks have to be manageable. You know it's in the best interest of your organization to move to Web services, but others don't share your foresight. In this article, we'll give you the tools you need to sell your organization on adopting Web services. Just remember though – you need to prove you can walk before you're allowed to run.

AUTHOR BIO:

Dan Foody, CTO at Actional Corp., has extensive hands-on experience in enterprise systems integration of packaged, legacy, and custom applications as well as middleware, platform, and emerging technologies used for integration. He is the author of various application integration standards. Dan holds a BS and MS in electrical engineering from Cornell University.

The first place to deploy Web services is within the enterprise, not across enterprises. This allows you to have total control over every aspect of the project, to manage the risk and ensure its success. What kinds of inside-the-enterprise projects are best suited to showcasing the benefits of Web services? Web services are all about connecting systems together, so integration projects are the right place to focus. Typically, a great starting point is a project with a Web front end that needs to access one or more other IT systems – portal projects, intranet Web sites, call center applications, etc. These projects are a great fit for Web services since they leverage its tight fit with existing Web technologies such as HTTP, XSL-T, and HTML.

The Evolution of Integration

You find an integration project well-suited to the use of Web services. You propose using Web services on this project and are told, "We've been doing integration for years without Web services, why do we need them now?" Best practices in integration have changed rapidly in recent years, and different models for integration have different trade-offs. It's easy for someone to be confused – so let's walk through the dif-

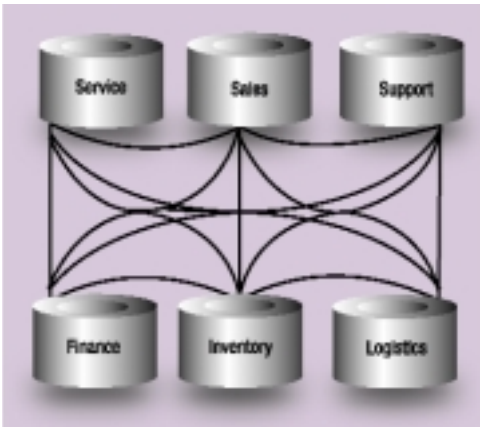


FIGURE 1 | **Point-to-point integration**

ferent models for integration: point-to-point, centralized, and backbone.

Point-to-point: The "Classic Coke" of Integration.

Point-to-point integrations have been going on since IT began. You know the type: "For my project, I need to connect system A's purchasing with system B's inventory management, so we'll just hand-code what we need right now to make this work." Even though point-to-point is the oldest way of integrating, it's still the most common for a few simple reasons: when projects are treated independently of each other, point-to-point is the cheapest solution for any given project and often the fastest way to get that individual project done. In addition, it puts total control in the project owner's hands with in-house resources and tools.

The downside of point-to-point integration is apparent when you look at the "big picture." If all projects are treated independently of one another, how can you possibly optimize your costs and productivity globally, across your projects? The answer is that you can't. Over time, a series of independent point-to-point integration projects breed a fragile spaghetti of application interconnections with no reuse among projects. Over the long run, you end up spending more and taking longer to get the job done while ending up with a brittle enterprise infrastructure (see Figure 1).

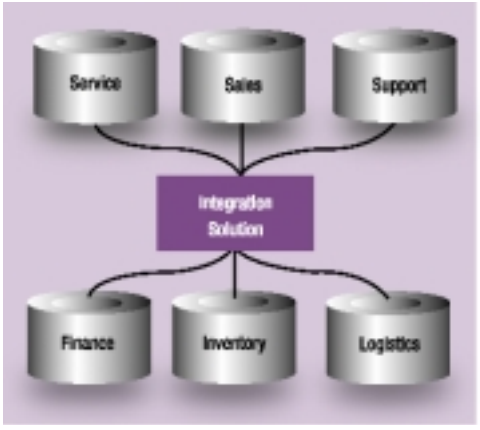


FIGURE 2 | **Middleman "integration solution"**

Centralized Integration: The "New Coke" of Integration

More recently, a few enterprising vendors realized the limitations of point-to-point and proposed a new twist on integration: instead of connecting applications directly together, put in a middleman "integration solution" (see Figure 2). The applications connect in a point-to-point fashion with the integration solution, but never directly to each other. This isolates the applications from changes in one another and provides a single point of control and reuse over all integrations – resulting in lower long-term overall costs.

Unfortunately, the strength of a centralized model is also its crucial weakness. Integration solutions suffer from the "first car builds the road" syndrome: the first integration project to attempt to use the integration solution is saddled with the bulk of the cost in both software and services. And, since you are "buying for the enterprise" rather than "buying for the project," the initial costs are very high. In addition, since the solutions are vendor-specific, the skills needed to build integrations are scarce, and the cost and delivery time of any individual project ends up being significantly more than with a point-to-point solution. And the projects themselves typically are done with outside resources or resources under the control of a central authority (because of the necessary skill sets), significantly reducing your control and increasing your risk.

To compound the problem, to get the long-term benefit from an integration solution you must standardize on one vendor's proprietary solution – a significant point of long-term risk and cost. Imagine the vendor is bought or goes out of business; your infrastructure might need to be totally replaced. Or, imagine you merge with another company; this means converting their infrastructure to your integration solution to gain back the benefits. Implementation of integration solutions often becomes the project that never ends.

For many of these reasons, single-vendor proprietary integration solutions are struggling now that the Internet bubble days of "infrastructure at any cost" are over.

Backbone Integration: The Choice of a New Generation?

Not exactly the new kid on the block, backbone integration has taken place for many years and is the benchmark integration model for large organizations. Most large financial service firms, for example, have application communication backbones. Instead of specifying a single platform or solution through which applications communicate (creating a centralization burden and risk), a backbone is formed by defining a set of standards through which applications communicate – distributing the responsibility for following the standards to each application group, division, or project (see Figure 3).

Typical backbones include standards for transport, protocol, encoding, description, location, and sometimes even data semantics. Any application that obeys these common standards is, by definition, "on the backbone." By having multiple applications obey these common standards, any application on the backbone can now communicate directly with any other application on the backbone.

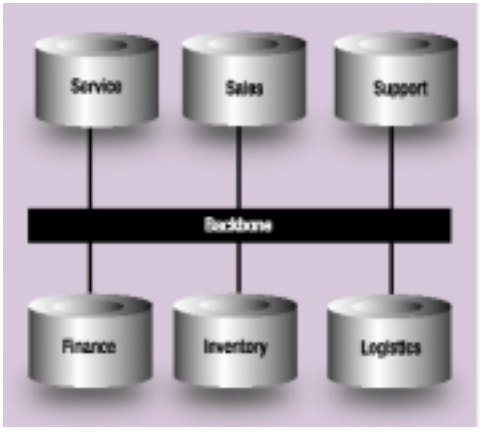


FIGURE 3 | **Backbone integration**

A backbone model has a number of significant advantages. It allows for a clear division of responsibility. It requires no single vendor or point of failure/risk. Backbones provide reuse of integration points across projects. Backbones scale effectively to the largest, most complex IT environments on the planet.

With all of these enterprise advantages, though, how do backbones work at the project level? With the right tools in place, backbone-based integrations are as fast and cost-effective to implement on a project-by-project basis as point-to-point integrations – precisely because backbones are built incrementally on a project-by-project basis.

All the advantages and none of the problems – why doesn't everyone do integration using an integration backbone? The answer is hidden in the last paragraph: "with the right tools in place…." Large IT organizations have the resources to custom-build the tools, frameworks, and infrastructure needed for the organization's custom backbone standards. Once these tools are built they can be effectively leveraged across the entire organization to dramatically reduce integration cost and time – but the cost of building a good custom toolset is typically prohibitive for all but the largest organizations.

What About Web Services?

Web services are a collection of related Web standards for transport (HTTP, SMTP), protocol (SOAP), encoding (XML), description (XSD, WSDL), and location (UDDI). Essentially, Web services form a core set of backbone standards – an "out-of-the-box" backbone.

Web services give you the standards, but the magic equation for forming a backbone is *backbone = standards + tools;* what about the tools? Because of its broad industry adoption, this is where Web services shine. By the end of this year most, if not all, tools and platforms will support Web services. Today, you can choose Web services–enabled tools such as Visual Studio .NET or Borland JBuilder. You can choose Web services–enabled platforms such as .NET Server, BEA WebLogic, or IBM WebSphere. If you want to leverage your existing tools, you can use free or open-source technologies such as the Microsoft SOAP toolkit or Apache SOAP. The choices are limitless.

But, beyond tools and platforms, most packaged application vendors have announced support for Web services. This means that, once available, these applications will plug almost directly into a Web services backbone with no custom coding – something not possible with custom backbones.

Web services are a set of standards and tools that make backbone integration available to every organization, large and small. With the tools readily available, a Web services backbone can be built up incrementally, on a project-by-project basis, without the need for a "big bang" implementation. This essentially lets you begin deploying Web services as part of point-to-point integration projects, and then seamlessly scale these up to a complete Web-services backbone.

Of course, you have the option to define a custom set of standards for building a backbone, so let's recap why Web services form the most effective choice for a backbone:

- Web services allow you to consolidate B2B, HTML, and application integration projects to use a single set of compatible standard technologies.
- Web services have achieved broad industry adoption. Most future purchases of packaged applications, tools, and platforms will be Web services–compatible. No single vendor is in control of the standards.
- Web services have a low cost of entry because all of the necessary software – TCP/IP, HTTP, XML, etc. – to get started is already available on virtually every machine.
- Because Web services technologies are based on XML, HTTP, and other "text-based" standards, the cost of administration is very low. For example, instead of needing specialized custom tools to monitor Web services (which would be required if Web services were a complicated binary protocol), any simple tool for packet sniffing makes it easy to monitor how Web services interact.
- Web services can be deployed incrementally, on a project-by-project basis, but scale up to a large enterprise backbone. This reduces the risk of adopting Web services, while improving the costs on a per-project basis as well as enterprise-wide.

> **"[Projects to Web-enable your existing applications] are a great fit for Web services"**

- But we already paid for XYZ, why do we need Web services?

There's a lot of hype and misunderstanding in the marketplace about what Web services are. Depending on who you talk to, Web services are the savior of software, the nirvana of networks, the best thing since sliced bread, or just another technology being forced on customers to replace what they bought last year. Web services aren't necessarily any of these, so understanding how Web services already complement your enterprise's existing tools, platforms, and technologies is important. Let's look at some common objections.

- *We run MQSeries, so we don't need Web services.*

MQSeries is a reliable message transport that complements Web services. By analogy, if MQSeries is the telephone network, Web services is the language spoken by callers. An effective conversation can't take place without both. Depending on your needs, you might choose HTTP as a transport for Web services where reliability isn't critical but low cost of deployment is, and you might choose MQSeries as a transport for Web services in other cases where you need guarantees of once-only message delivery. Web services standards are flexible enough to allow you to choose the transport that best suits your needs. Many of the tools already on the market, in fact, support MQSeries as a Web Service transport.

- *We have a business process management system (BPMS), so we don't need Web services.*

BPMS defines business processes that coordinate operations across different business services. But how do you make those services available to the BPM engine? One choice is to use a proprietary set of adapters that the BPMS vendor sells – but this is costly, complicated to imple-

ment, and the only system in the enterprise that will be able to leverage this work is the BPMS itself. Alternately, if you publish the business services as Web services, then not only can the BPMS use the services, but any other application that needs to leverage them can also do so. Using a BPMS in conjunction with Web services is the most cost-effective way, both short and long term, to build your enterprise infrastructure.

- *We have an application server/portal/integration solution, so we don't need Web services.*

As with BPM solutions, Web services are not about how you define, combine, or use services (which is what application servers, portals, and integration solutions are all about), but about how you access services. You always have the choice of accessing services through custom-built code or tightly coupled proprietary adapters with any of these solutions. But these custom vendor-specific solutions are expensive, available skill sets are limited, and they don't easily scale to broader use. Alternately, you can access services using Web services standards, which provide a cost-effective, vendor-neutral, scalable solution for accessing services. Web services are the most effective way to leverage use of your application server, portal, or integration solution.

Web Services Sound Great, but…

With all the advantages of Web services, what are the downsides? There are really two issues to consider. The first is that your enterprise may have already invested hundreds of thousands to billions of dollars in applications that aren't Web services–compatible – and these applications aren't going away. How do you leverage these existing assets without rebuilding them and without truckloads of hand-coded Web service wrappers? Second, even when you get some of your existing applications published as Web services, these applications – not having been built on Web services standards originally – are likely to have their own security and management systems. Even for native Web services, there are multiple competing standards for such things as security. How do you provide a consolidated single model for security and single point of management for your emerging Web services backbone?

Luckily, an emerging category of software, Web services gateways, addresses these issues. Web services gateways allow existing non–Web services applications to be published as if they are natively written Web services. At the same time, these gateways act as a single point of management that transparently adapts between the standards you choose for your backbone (security, reliability, transport, semantics, etc.) and the standards used by the applications themselves – at the application level they serve much the same purpose, in concept, as a multiprotocol network router for low-level network traffic. This allows you to have a well-defined and manageable backbone, while at the same time supporting the diversity of your application portfolio. The best Web services gateways are totally transparent to the applications, requiring no application changes or modifications while providing the infrastructure for mapping between different standards in each area (security, transport, etc.) with the flexibility to accommodate your unique environment, now and in the future, with little or no hand-coding.

> **"Web services form the most effective choice for an integration backbone"**

The Venus Flytrap

Web services have tremendous possibilities when deployed well, but there are some pitfalls that can cripple your ability to derive long-term benefits from them. Even though you're going to start small, you need to think ahead. If the long-term goal is to have a unified Web services backbone, don't make choices now that will hamper your ability to reach this goal in future. The two key traps to avoid here are:

- ***Don't fall for "Web services lip service":*** Some vendors will try to convince you that they support Web services standards, but to gain the real advantages of their solution you need to use their proprietary, nonstandard capabilities, or deploy their proprietary software together with each application (giving them total control of the backbone). This approach locks you into the vendor's proprietary solution – whether it's Web services–based or not. Once you choose to go down this path, there's no easy way to turn back.
- ***"The great thing about standards is that there are so many to choose from."*** Defining a Web services backbone means that you will need to choose from among the various options for security standards, transports, etc. These decisions should not be made on a project-by-project basis; other -wise, you may end up with as many differences as there are similarities – your backbone may be all Web services–based, but no applications can talk to each other. Make sure to choose a cohesive set of standards and stick to them. This will allow the individual projects to, over time, form the basis for your common Web services backbone.

Conclusion

Web services can be deployed initially at low risk and at low cost. At the same time, a thoughtful Web services enterprise architecture plan, begun now, can enable your enterprise to gain significant long term advantages in terms of cost reduction, flexibility, scalability, and responsiveness to changing business needs. Make it happen!

written by Howard Smith

Making Business Processes Manageable

The chance to provide a new solution to an old challenge

What has surprised everyone in the past few years is how challenging it has been to actually do e-business. One of the reasons why this is so is that companies have found it difficult to manage their business processes, especially when they stretch across multiple systems, software applications, companies, and countries. That's about to change.

Howard Smith is cochair of the Business Process Management Initiative (BPMI.org) and CTO of Computer Sciences Corporation in Europe. Howard is coauthor with Peter Fingar of the forthcoming book: *Business Process Management: The Third Wave* (mk press.com).
HOWARD.SMITH@ONTOLOGY.ORG.

It must change, because shareholders still expect companies to fulfill the promise of e-business. As Doug Neal of CSC's Research Services reports, "Companies are under pressure to perform better, faster; to do more with less; and to be super pleasing to customers. This means changing the way they manage their business processes, allowing them to innovate around their own strategic processes while simultaneously collaborating with partners and customers."

Why Is It So Difficult?

Many companies tried to make their business processes more manageable 10 years ago, by reengineering them. At the time, reengineering typically meant designing a new process, then implementing it through a one-time systems and organizational change program. These efforts were more about redesigning processes than about making those processes easy to change and combine with those of partners. Similar problems existed with ERP and other packaged solutions that emerged later. These packages implemented best-practice processes, but did so by ingraining business processes in the software applications that supported them. These solutions had all the flexibility of wet concrete before they were installed and all the flexibility of dry concrete after installation.

Achieving process collaboration inside the enterprise has been difficult enough. Getting processes to collaborate across the networked enterprise is much harder. B2B participants may have informal designs for the processes they need to implement, but as they refine those designs they also have to change the technical implementations that support them. This may be possible in simple cases, but in more complex cases, such as advanced supply chain design, the project may never be completed. According to Intalio, a vendor of standards-based Business Process Management Systems (BPMS), "upgrading applications or adding new suppliers or business units can cause the technical integration activities to escalate out of control." Adding process tools and best practices to existing middleware approaches helps, but it would be preferable if something similar were embodied in the platforms that support applications.

In the networked enterprise, collaboration isn't restricted to any one process domain. Collaboration is now 360 degrees, going on at all points on the compass. This creates a many-to-many integration task, and existing tools and techniques simply are not up to the job. IS departments often try to develop business processes by performing bottom-up technical

What The Analysts Are Saying

In the world of alliances and "virtual companies," where many or even most of a firm's staff may be working on joint projects with other businesses, what makes an appealing partner? There is a quick and easy checklist concerned with products, prices, market access, financial muscle, and track record. But an increasingly dominant question is, what is it actually like to work with these people? How do they do business? Is their way of doing business likely to prove comfortable and compatible for us? A company's processes are becoming a key criterion in the beauty parade for global partners. Is it possible to doubt that the quest for a universal process descriptor language will be successful? It is increasingly hard to doubt that the quest will take place, and will command substantial interest and resource during the next decade.
– CSC Research Services

It's no surprise then that BPM is quickly emerging as the moniker for the next killer app in enterprise software. Few areas of software will receive more attention in the coming months and years than BPM. Yet the greatest challenges to the BPM market are the very forces making it so attractive.
– Delphi Group

Enterprises should begin to take advantage of explicitly defined processes. By 2005, at least 90 percent of large enterprises will have BPM in their enterprise nervous system (0.9 probability). Enterprises that continue to hard-code all flow control, or insist on manual process steps and do not incorporate BPM's benefits, will lose ouat to competitors that adopt BPM.
– Gartner

For the Fortune 2000 companies, the quest to implement the best business process management (BPM) solution is becoming highly desirable – akin to acquiring the "holy grail" in any given industry. BPM promises to streamline internal and external business processes, eliminate redundancies, and increase automation.
– IDC Research

integration, stitching together systems components that were never intended to work together at the business level. They soon find that these projects denude their budgets, and return on investment and delivery-time scales are often unacceptable to the project sponsors. Not only that, but changing processes thereafter, developing custom variants, or optimizing processes on particular business channels is either hard or impossible.

Not all integration problems are technical. Collaboration requires sharing representations for processes that once were proprietary, and this is not an easy step to take. But if companies are under pressure to be better, faster, and cheaper, they will have to do only what they do best; whatever a company doesn't do well has to be done by someone else – hence, the growth in process outsourcing and the sourcing of external services. If commerce is to be truly collaborative, the underlying business processes must collaborate too, both within and across firms. This must be achieved at the business level and from the top down, leveraging existing systems in the enterprise. That is, collaboration must start from the business purpose, not the technical constraints.

What's Needed

What business needs is not a one-time fix for individual processes but a connected systems environment that can flex and recombine as required by changes in the market. Most companies now want more control over their own processes; more interaction between their processes and those of their partners: and some control over, and monitoring of, processes performed on their behalf by partners. Firms are also seeking to expose discrete business competencies as processes they can sell to others or through channel partners. To do all this, firms need to understand the processes that underpin their core business competencies. In short, they need a BPMS capability, not a new suite of enterprise applications.

The situation is similar to the period before the invention of the relational database management system. Business data used to be embedded in applications. As the volume of data grew and the connections between data sets in different applications became important to business, it became obvious that data should be managed outside of the application architecture. For example, managers wanted to analyze the data for business performance indicators. To achieve this, the new methodology for data management was built on a formal model called the relational data model. Today, this is commonplace. By allowing a company to manage its data apart from the applications that use it, the database management system (DBMS) supports a variety of data models and data management tasks and tools. The IT industry as we perceive it today is largely founded on the DBMS. Today's enterprise applications are primarily concerned with reading, writing, and manipulating data tables – in other words, clerical tasks. This has a profound implication – such applications are stovepipes. Business logic, data model, time, and connectivity all exist within the individual application. Creating and managing end-to-end processes has, up to now, depended upon complex middleware solutions. Not only is this expensive, it's overly complex.

New process management systems offer a potentially simpler, more cost-effective, and manageable alternative. One immediate benefit will be the ability to align processes more directly with organizational objectives. Business processes literally define the firm and represent the source of all competitive advantages and market differentiation. Business processes are complex, long-lived, unique, numerous, and constantly evolving. As the drive toward automation, process outsourcing, and collaboration continues apace, processes have the potential to overwhelm the firm. Process management systems are tools for managing that complexity.

In this climate, standard processes deliv-

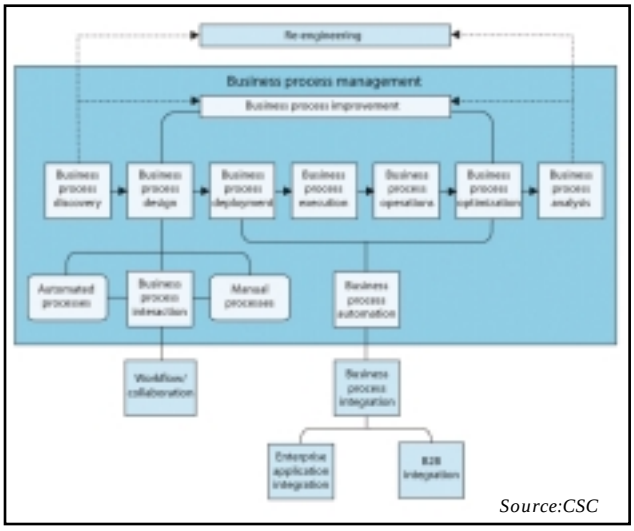


FIGURE 1 | **Process management terminology**

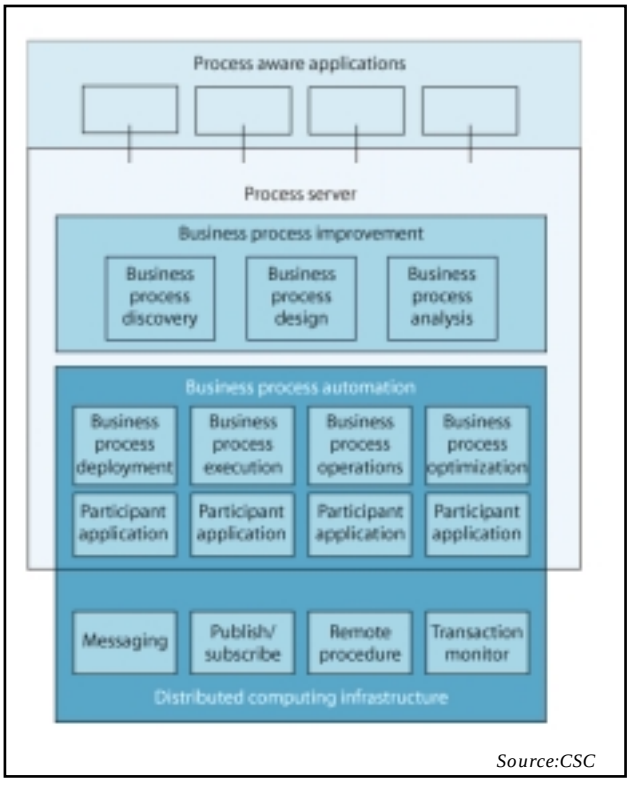


FIGURE 2 | **From BPM layer to BPMS capability**

ered in the form of standard applications that are also available to competitors are less and less attractive. Businesses want to shape their processes themselves, perform continuous and incremental process improvement without impediment from technology, and simultaneously exploit low-cost best-of-breed application components. Powerful new process servers will support this approach, providing a hybrid environment combining the best of component-application engineering and the best of process engineering. The era of stovepipe applications will eventually give way to the era of process manufacturing.

A Process Language

The first step is to make processes explicit by abstracting them from application software. This is hardly new. Decades ago, operating systems were created by abstracting memory management, file access, and graphical user interfaces from applications. Database management systems removed both the management of data and the management of the schema. Today, business rules are held and managed in a separate environment. Process management is the next logical step.

One of the key enabling technologies is Business Process Modeling Language (BPML), published by the Business Process Management Initiative (BPMI.org). The new language – which, like SQL, has a strong mathematical foundation – is a methodological way to represent and interact with business process (see Figure 1). In effect, the process management system abstracts processes out of the application code and by doing so provides a powerful new capability to business: the ability to discover, design, deploy, execute, operate, optimize, and analyze processes independently of the applications built on them. The applications get an end-to-end view of the business, looking down from above. They see the whole process, not just the changes in data as the process executes.

The processes such systems will support will be reliable, transactional, concurrent, and distributed. They will allow collaboration between processes designed independently by different organizations. Like other standards, the value of BPML will be perceived only by demonstrating the power of a process management system to the business.

The Shift From the BPM Layer to BPMS Capability

Many Fortune 2000 companies are deploying BPM point solutions or individual layers in the BPM stack. They find it necessary to deploy EAI (enterprise application integration). Some use workflow extensively, a few are piloting departmental process managers, all use process engineering tools (BPR), and many have employed rules- or process-based development to speed application delivery. It's a complex picture. How this mosaic will migrate toward a standard process-management environment will be a fascinating journey. It won't happen overnight, and it won't happen at all unless a solution emerges that is both as compelling as the RDBMS and SQL and as well-founded as the relational data model.

Plugging all the pieces together is taxing even the most able enterprise IT architects. The hope is that process management will soon be a capability (the BPMS) leveraged by applications, not a layer in an already complex application stack (see Figure 2). It has to be, since the drivers for BPMS are not only technological but economic. The most complex, dynamic processes tend to be those of higher business value and strategic importance. Such processes are difficult to coordinate across multiple partners, are unique to the firm, embody sources of competitive differentiation, and rely upon continual process improvement and improved decision-making. BPMS offers the chance to provide a technological solution to an age-old problem – BPR and organizational change.

Web Services Journal
FOCUS ON BPM

written by Ashish Deshpande

Web Services-Oriented Architecture: A Critical Technology

A graphical interface allows business users to model core business processes

The Web has always been service-oriented. Services such as auctions or ticket-buying services are available today to general Web users. Essentially, what the Web did for interactions between business functions and their users, Web services will do for interactions between functions spread out across multiple business units or even across multiple businesses.

Dr. Ashish Deshpande, CTO and cofounder of Metaserver, is a leading authority on the business process integration space and has more than 15 years of experience with B2B technology, network computing, and distributed parallel processing. He holds a PhD in computer science from Yale University, an MS in computer science from the University of Virginia, and a bachelor of technology in computer science and engineering from the Indian Institute of Technology at Bombay.

Because of this, a number of organizations are trying to design and build a Web service–oriented architecture (WSOA). The good news is that most of them are already on the right path to getting there. This next wave of application development is a natural extension of the move toward component-based standards such as J2EE and COM. Web services are unique in that they rely entirely on Internet-standard technologies. (*Note:* Web services rely on existing standards such as XML messaging [SOAP], Internet standard transport [HTTP, FTP, SMTP], and emerging standards such as WSDL, UDDI, and WSFL.) They build on distributed, component-oriented environments, rather than throwing them away and starting from scratch.

The biggest departure from component-based development (CBD) is that a WSOA is likely to be very process-centric as opposed to the program-centric applications of the earlier waves of application development. That's because organizations will use Web services to provide access to higher-level business functions rather than to their components or legacy systems. For example, car rental agencies want to expose the process of renting a car or checking on its status, and not the inventory system or the order entry system that implements parts of those business functions. Similarly, an insurance company wants to expose the process of renewing an insurance policy or checking the status of a claim, instead of providing access to the legacy systems or databases that contain the required information.

A critical enabling technology is business process integration (BPI). BPI extends integration capabilities by providing a graphical interface that enables business users to model business processes that are core to an organization's operations. BPI allows the use of functionality from existing applications, regardless of the technology with which the applications were developed. BPI also handles issues such as load balancing, fault-tolerance, and scalability, so business processes can be executed at any time by as many customers and partners as necessary.

Essentially, BPI enables organizations to create an internal WSOA that lets them effectively leverage their existing investments and take advantage of Web services. Basic to the WSOA paradigm is the shift away from the traditional approach of building customized systems to focusing on the business process. Organizations have heavy investments in existing systems, applications, and people. A WSOA built on

Richard Hale Shaw Group

www.richardhaleshawgroup.com

the foundations of BPI provides a practical approach to delivering business value while preserving existing investments. BPI enables the WSOA to evolve incrementally and forces it to stay closely aligned with business drivers. It provides an alternative to the endless treadmill of application rewrites and custom coding.

Building a WSOA

In a WSOA, components are aggregated in a simplified manner to accomplish a specific business function. Components are binaries that have a defined interface and perform a specific task, for example "authenticate user," "debit account," or "approve credit card." A service is simply a logical grouping of these components to accomplish a business function such as "renew an insurance policy" or "rent a car" or "process a loan application." This logical grouping of components can then be exposed to the outside world over the Internet as a Web service.

With a WSOA, the focus on the business process enables a much higher level of application development than in the past. It's also more closely tied to business requirements than technical requirements.

For this potential to be realized, the BPI technology must enable processes to be quickly and dynamically created from functionality in multiple, preexisting, back-end systems that are incompatible. It must allow deployment of those processes to a robust execution engine. Finally, it must facilitate the exposure of these processes or various parts or combinations of these processes via the Internet-standard technologies of Web services.

Newer, emerging BPI products enable model-driven visual definition and automation of business processes. Using a sophisticated modeling environment, organizations can define business processes and leverage a powerful connectivity framework that includes numerous point-and-click wizards for interacting with existing legacy business systems as well as newer component-based systems, including J2EE, COM, and Web services themselves. Some even provide a high-performance, high-availability engine for deploying and executing these business processes in an extremely reliable manner. Any such process, portions of the process, or groups of these processes may be exposed to the outside world in the form of Web services.

Building a WSOA with BPI

Using an insurance industry example helps illustrate how to build a WSOA. Many insurance companies have a large number of existing monolithic applications that have been built up over the years using a variety of incompatible technologies. They performed well on the tasks for which they were originally designed, but little thought was given to interoperability between these systems. Many of these organizations now want to deploy a WSOA that provides access to key business functions such as "renewing an insurance policy" or "checking the status of a claim" in a standardized manner over the Internet. They want to build and deploy the WSOA incrementally by automating business functions in bits and pieces, adding to them over time, leveraging existing applications, and building on those successes.

The WSOA is built in layers using existing technology. Below is a description of the layers and then an overview of the combined work at the end.

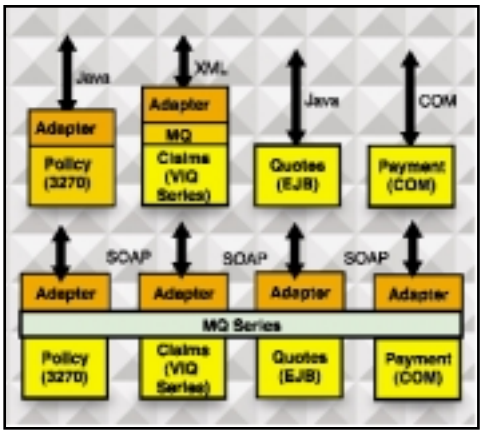


FIGURE 1 **Components access existing back-end systems**

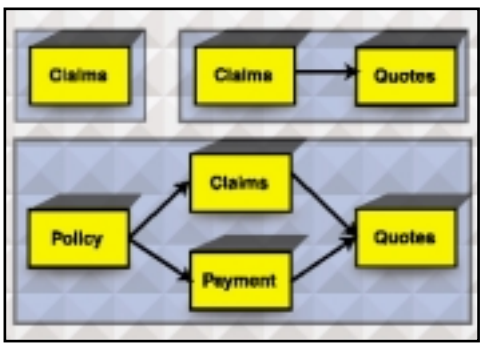


FIGURE 2 **Services represent business functions**

Components

A component is a binary piece of code whose sole purpose is to do one and only one thing. In the example, there are four components, each having its own purpose:

- ***Policy:*** Provides access to policies that have been issued, including detailed policy information. It will need to access a legacy IBM mainframe system in order to provide this functionality.
- ***Claims:*** Provides access to the company's claims system, including the status of any claims or the list of outstanding claims against a particular insurance policy. The claims system is also implemented via a legacy IBM mainframe system but is accessed via MQSeries.
- ***Quotes:*** Uses the latest information avail-

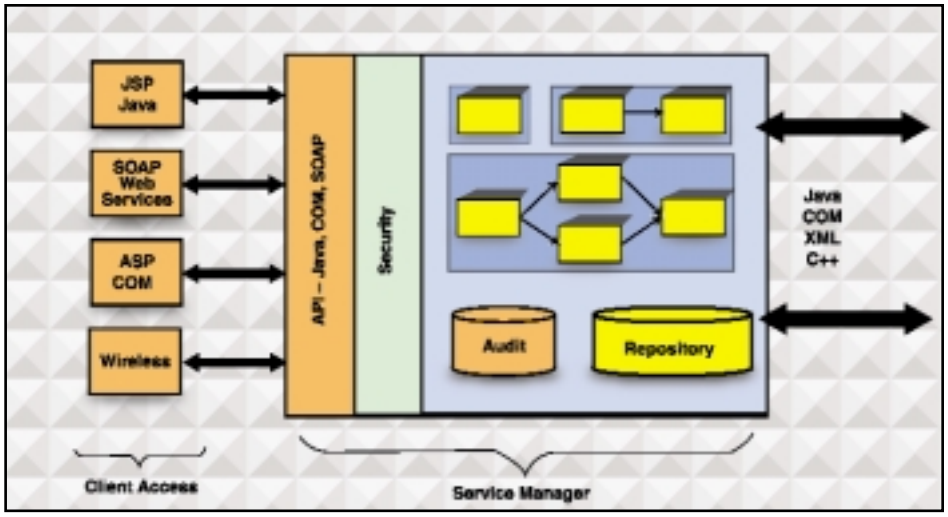


FIGURE 3 | **A "service manager" completes the WSOA**

able to generate a quote for a particular insurance policy. Given existing policy information, new policy requirements and other information about an applicant, this component can compute the premium for a new policy. The quotes system was recently rebuilt using EJB technology and runs in the WebSphere environment.

- ***Payment:*** Provides access to payment information, including premiums that have been paid for a particular policy, payment dates, etc. This system runs in a Microsoft environment.

In a WSOA the components may utilize the same Internet-standard technologies of Web services. The required back-end functionality may be accessed via SOAP messages sent over a messaging layer such as MQSeries. However, this is not required and may not add any business value, particularly since some BPI products can already communicate with a variety of back-end systems to automate the business process.

As Figure 1 shows, the most convenient way to access each component is different. BPI technology allows each component to be created using the most appropriate technology for that platform. However, it doesn't preclude the creation of internal Web services to expose the functionality in each back-end system.

By and large, most organizations are already well down the path toward componentizing using one or more technologies. BPI technology can leverage these efforts and build on top of existing component strategies to enable the WSOA.

Services

A service is simply a logical grouping of these components to accomplish a business function such as "renew an insurance policy" or "check the status of a claim." This logical grouping of components becomes a Web service if it is exposed to the outside world using Internet-standard technologies such as SOAP and WSDL.

In this case, the business requirements are to provide access to several business functions. Customers should be able to check the status of a pending claim and agents should be able to generate quotes for new policies or renewals. Executives would also need reports describing the number and kind of new quotes that were generated and policies that were issued in any given time period. The following services are a subset of these business requirements (see Figure 2):

- ***Check Claim Status:*** This service is utilized by end customers or by insurance agents to check the status of a pending claim.
- ***Renew Policy Quote:*** Agents use this service for quoting a premium for a policy renewal.
- ***Daily Report:*** Managers can obtain reports about daily activity.

BPI technology allows business analysts to rapidly model business processes that combine components to create the above services. Typically, the process model is very rich and includes support for business rules, data transformations, and automatic error-handling.

It doesn't require all components to be created using one particular standard such as EJB, COM, or XML. The process can mix and match components created using incompatible technologies.

Service Manager

The service manager is the runtime engine that enables client applications to access and execute the services that have been created above. The engine must deliver high performance and high reliability to the WSOA without requiring the organization to invest significant expertise on infrastructure and middleware.

A BPI engine can provide a very high-performance, scalable, robust execution environment for internal as well as collaborative business processes. The engine is equally capable of executing long-running processes that may run for weeks as well as short-running straightthrough processes that are fully automated and may run for a few milliseconds.

Client Access

The WSOA provides services that are designed for programmatic access and does not include a user interface layer (see Figure 3). This provides maximum flexibility. The same service may be accessed from a user interface such as a Web browser via a JSP or from a business partner application via an XML SOAP message sent over the Internet. BPI provides the ability to expose the business process via a variety of technologies. The same business process can be accessed:

- As a Web service using XML messaging over HTTP

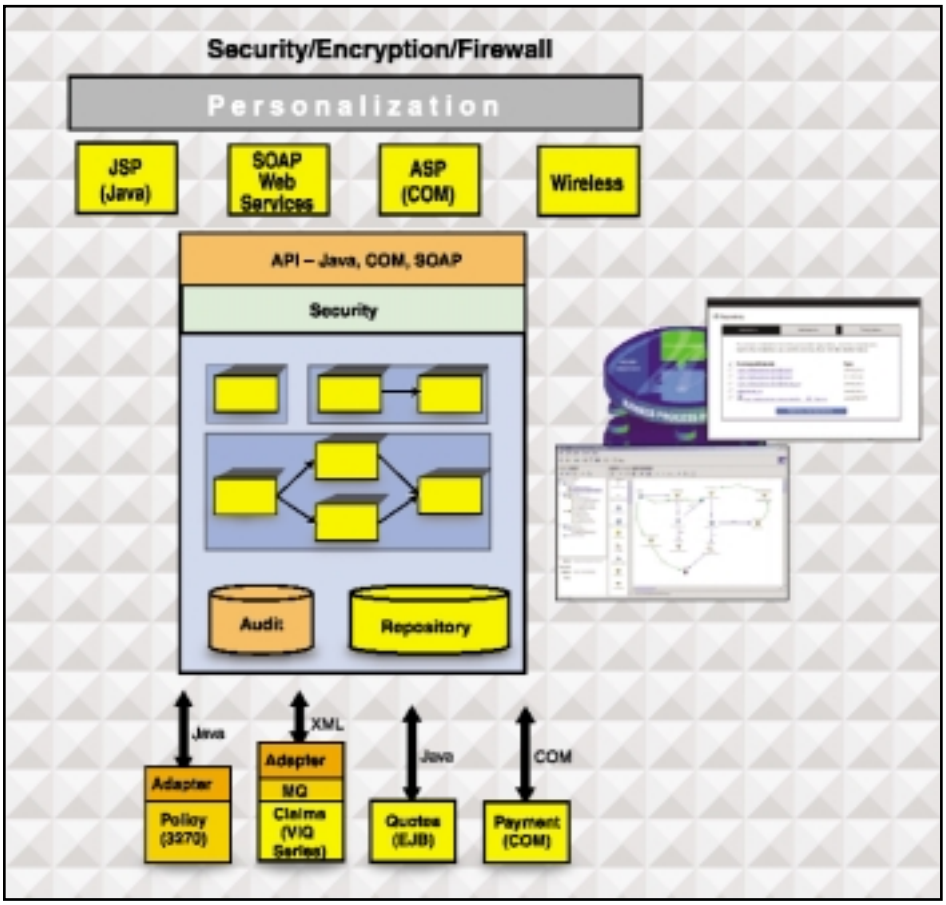


FIGURE 4 | **Complete WSOA built around BPI technology for the service manager**

- By a Web browser through a portal using JSP, ASP, servlets, etc.
- By fat clients such as Java or VB applications
- By mobile devices such as cellular telephones or wireless PDAs.

Putting The Pieces Together

Figure 4 shows a complete WSOA built around BPI technology for the service manager. The WSOA builds upon the existing technologies that are already deployed within numerous organizations. The business processes that define the operation of the business can be easily modeled – business users who understand the processes can do this work. Similarly, collaborative business processes between business partners can be modeled within this environment. These processes can be fully automated or they can retain the necessary elements of human interaction depending on business needs. The process metamodel is rich enough to support all the requirements of short-running, fully automated processes or long-running human-interactive processes.

The variety of applications and technologies typical of most organizations can be incrementally incorporated into the integration via wizards, tools, and adapters provided by the same modeling environment. The business process models are then deployed to an execution engine that automates all the issues normally associated with complex, distributed systems, including performance, scalability, load balancing, and fault tolerance. The engine provides access to the business process models through a wide variety of client protocols.

Typically, the system can be easily administered from a single, centralized location using nothing more than a Web browser. A powerful, sophisticated security infrastructure, based on existing corporate and Internet security standards guarantees secure access to the business functionality embodied in the exposed business processes and ensures that access is provided only to authorized entities.

Finally, the modeling environment provides a single, unified graphical user interface that business users and developers alike can use to do their work and share among those users. This reduces training issues and costs associated with deploying multiple tools.

Summary

Web services enable organizations to automate interactions between business units, thereby reducing the cost of doing business and making the overall process more effective, efficient, and flexible. Ultimately, Web services improve customer satisfaction and favorably impact the bottom line.

To effectively implement Web services, many organizations are already designing and building a WSOA. Fortunately, this new wave of application development is a natural extension of component-based standards and architectures like J2EE and COM. Web services are built on the distributed, component-oriented environment that is already in place at many organizations.

A critical enabling technology for implementing a services architecture is business process integration. BPI lets organizations create an internal WSOA that allows them to effectively leverage their existing investments and take advantage of Web services. A WSOA built on the foundations of BPI provides a practical approach to delivering business value while preserving existing investments in people and systems. This is because BPI enables the WSOA to evolve incrementally and provides a seamless alternative to endless rewrites and custom coding.

THE LARGEST INTERNATIONAL

XML CONFERENCE & EXPO IN THE WORLD!

XML EDGE conference & expo

JUNE 24-27
JACOB JAVITS CONVENTION CENTER
NEW YORK, NY

OCTOBER 1-3
SAN JOSE CONVENTION CENTER
SAN JOSE, CA

WIN A $35,000 LUXURY CAR!

ATTENDEES WILL BE INVITED TO TAKE A GOLF SWING TO WIN AND RIDE OFF IN A $35,000 LUXURY CAR!

XML-NEXT G OF ENTERPRISE DEPLOYMENT

REGISTER ONLINE TODAY

FOR LOWEST CONFERENCE RATES
EARLY SELL-OUT GUARANTEED!

VISIT WWW.SYS-CON.COM

Focus on XML

XML is today's essential technology to develop and deploy Web services. Here's your chance to learn from expert practitioners how XML is making the next phase of the Web a reality.

Focus on standards, interoperability, content management, and today's new quest for more efficient and cost-effective Internet and intranet-enabled business process integration.

Focus on XML during information-packed days while you enjoy an XML/Web Services Keynote panel, comprehensive conference sessions, and an unparalleled opportunity to exchange ideas with peers and industry leaders.

A Sampling of XML-Focused Sessions

- XML STANDARDS - AN OVERVIEW
- OASIS STANDARDS UPDATE
- UBL - A UNIVERSAL BUSINESS LANGUAGE
- BRINGING XML TO PKI
- LINK MANAGEMENT WITH XLINK
- PRACTICAL XSLT AND XPATH
- ENTERPRISE CONTENT MANAGEMENT WITH XML
- XML IN THE ENTERPRISE AND INTER-ENTERPRISE WORLD

ROBERT MIKULA
XML CHAIR
BOARD OF DIRECTORS, OASIS
INDUSTRY EDITOR
JAVA SERVICES JOURNAL

Featuring...

- UNMATCHED KEYNOTES AND FACULTY
- THE LARGEST INDEPENDENT JAVA, WEB SERVICES, AND XML EXPOS
- AN UNPARALLELED OPPORTUNITY TO NETWORK WITH OVER 5,000 I-TECHNOLOGY PROFESSIONALS

Who Should Attend...

- DEVELOPERS, PROGRAMMERS, ENGINEERS
- I-TECHNOLOGY PROFESSIONALS
- SENIOR BUSINESS MANAGEMENT
- SENIOR IT/IS MANAGEMENT/C LEVEL EXECUTIVES
- ANALYSTS, CONSULTANTS

Hear these thought leaders in interactive, cutting-edge keynote addresses and panels...

JEAN-FRANCOIS ARNAUDIC
SENIOR VP R&D, ILOG
FORMER CHAIRMAN, W3C

TYLER JEWELL
PRINCIPAL TECH. EVANGELIST • BEA

DAVID LITWACK
CEO
SILVERSTREAM

ANNE THOMAS MANES
CTO
SYSTEMS

BARRY MORRIS
CEO
IONA

RICK ROSS
FOUNDER
JAVALOBBY

PATRICIA SEYBOLD
FOUNDER & CEO
SEYBOLD

For Exhibit Information

CONTACT: RICHARD ANDERSON
135 CHESTNUT RIDGE RD.
MONTVALE, NJ 07645
201 802-3058
RICHARD@SYS-CON.COM

SPONSORED BY:

IONA | END 2 ANYWHERE™
ADDS
bea
TogetherSoft
SilverStream
Actional
MERANT
Altaworks
PolarLake
ALTOVA

MEDIA SPONSORS

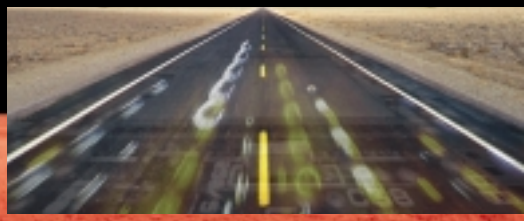
Federal Computer Week
Webservices.org
XML TIMES.com
Java Skyline
CF Advisor
WebServicesMall
XML.ORG
WIRE
jCert
wrox
OASIS
XML.ORG
SD Times
JAVA
wireless
XML
WebLogic
WebServices
WebSphere
ColdFusion

OWNED AND PRODUCED BY

SYS-CON MEDIA
SYS-CON EVENTS

Web Services Journal
FOCUS ON BPM

written by Steve Benfield



Real World WSFL

A useful and well thought-out specification that's easily extended

Web services has promised many things. One primary promise has been the ability to piece applications together by snapping Web services together like so many Lego blocks. The output of one service becomes the input to the next and so on.

Steve Benfield is CTO of SilverStream Software. SilverStream eXtend is the first complete environment for delivering service-oriented applications. SilverStream eXtend Composer: Business Process Manager, represents the first commercial implementation of WSFL on the market.
SBENFIELD@SILVERSTREAM.COM

In 2001, IBM published a specification called WSFL 1.0: Web Services Flow Language. WSFL is a language used to define business processes using Web services. By implementing WSFL, you can create process definitions that can be used by any WSFL-based business process engine. In addition, any process defined in WSFL can, itself, become a Web service, allowing composition of more and more complex and coarse-grained processes.

My company worked directly with IBM to interpret and implement their specification; this article describes WSFL at a high level and concludes with some thoughts on WSFL and its future.

Why Business Process Modeling?

Though developers have been building business processes for years, they've generally been hard-coded, which is precisely what we're trying to avoid when defining business processes using Business Process Modeling (BPM). It's easier to understand and comprehend a visual business process model than it is to read a collection of objects and Java code. WSFL provides a way to separate the actual process model from the underlying implementation. This allows nondevelopers and business analysts to build the business process flow and map the individual steps of the process to Web services. Developers spend their time implementing business functionality rather than writing the complex code required to link them together.

Basics of WSFL

At the core of WSFL are two things: a flow model and a global model.

- ***Flow model:*** Abstractly defines a business process and consists of activities, messages, control links, and data links.
- ***Global model:*** Connects the activities and messages of the flow model with the Web services used to implement them. The global model may also define the public interface of the flow and specify how the process flow's WSDL should be generated.

Flow Model

The actual process flow in WSFL is described as an acyclic-directed graph. Before you break out the mathematics textbook, this basically means that activities have directional links between them and you can't loop back to a previous activity.

Activities

Activities, which represent work to be

THE LARGEST INTERNATIONAL

WEB SERVICES CONFERENCE & EXPO IN THE WORLD!

web services EDGE conference&expo

JUNE 24-27
JACOB JAVITS CONVENTION CENTER
NEW YORK, NY

OCTOBER 1-3
SAN JOSE CONVENTION CENTER
SAN JOSE, CA

WIN A $35,000 LUXURY CAR!

ATTENDEES WILL BE INVITED TO TAKE A GOLF SWING TO WIN AND RIDE OFF IN A $35,000 LUXURY CAR!

WEB SERVICES
SKILLS, STRATEGY, AND VISION

REGISTER ONLINE TODAY

FOR LOWEST CONFERENCE RATES
EARLY SELL-OUT GUARANTEED!

VISIT WWW.SYS-CON.COM

Focus on Web Services

Web Services, the next generation technology that will enable the Internet to work for you and your business, and finally provide that ROI you have been after, will be put under a microscope at Web Services Edge East 2002.

Information-packed sessions, exhibits, and tutorials will examine Web Services from every angle and will provide cutting-edge solutions and a glimpse at current and future implementations. Hear from the innovators and thought leaders in Web Services. Enjoy a highly interactive CTO Keynote panel that will debate and discuss the realities and promise of Web Services.

A Sampling of Web Services-Focused Sessions

- PRACTICAL EXPERIENCES WITH WEB SERVICES AND J2EE
- STATE OF THE WEB SERVICES INDUSTRY
- THE ODD COUPLE: MAKING .NET AND J2EE WORK TOGETHER
- EXPLORING THE .NET MY SERVICES INITIATIVE
- STANDARDS WATCH
- GUARDING THE CASTLE: SECURITY & WEB SERVICES

SEAN RHODY
CONFERENCE TECH CHAIR
WEB SERVICES TRACK CHAIR
EDITOR-IN-CHIEF
WEB SERVICES JOURNAL

Featuring...

- UNMATCHED KEYNOTES AND FACULTY
- THE LARGEST INDEPENDENT JAVA, WEB SERVICES, AND XML EXPOS
- AN UNPARALLELED OPPORTUNITY TO NETWORK WITH OVER 5,000 I-TECHNOLOGY PROFESSIONALS

Who Should Attend...

- DEVELOPERS, PROGRAMMERS, ENGINEERS
- I-TECHNOLOGY PROFESSIONALS
- SENIOR BUSINESS MANAGEMENT
- SENIOR IT/IS MANAGEMENT/C LEVEL EXECUTIVES
- ANALYSTS, CONSULTANTS

Hear these thought leaders in interactive, cutting-edge keynote addresses and panels...

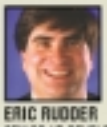
JEAN-FRANCOIS ABRAMATIC
SENIOR VP R&D, ILOG
FORMER CHAIRMAN, W3C

DAVE CHAPPELL
VP CHIEF TECHNOLOGY EVANGELIST
SONIC SOFTWARE

GREGG KIESSLING
CEO
SITRAKA

ANNE THOMAS MANES
CTO
SYSTINET

BARRY MORRIS
CEO
IONA



ERIC RUDDER
SENIOR VP DEVELOPER AND PLATFORM EVANGELISM
MICROSOFT

PATRICIA SEYBOLD
FOUNDER & CEO
SEYBOLD

For Exhibit Information

CONTACT: MICHAEL PESICK
135 CHESTNUT RIDGE RD.
MONTVALE, NJ 07645
201 802-3057
MICHAEL@SYS-CON.COM

SPONSORED BY:

IONA | END 2 ANYWHERE™
ADDS
bea
TogetherSoft
SilverStream
Actional
MERANT
Altaworks
PolarLake
ALTOVA

MEDIA SPONSORS

Federal Computer Week
Webservices.org
XML TIMES.com
Java Skyline
CF Advisor
WebServicesMall
XML.ORG
WIRE
jCert
wrox
BASIS
SD Times
JAVA
wireless
XML
WebLogic
WebServices
WebSphere
ColdFusion

OWNED AND PRODUCED BY

SYS-CON MEDIA
SYS-CON EVENTS

performed, are eventually mapped to Web services. In the flow model, you can define the input message, output message, and faults of an activity. The activity's messages will eventually be mapped to the messages of the underlying Web service implementation.

Control Links

Activities are wired together through control links. Any number of control links can exit or enter an activity, but there can be only one control link between any two activities. Once an activity is completed, the WSFL engine will traverse to the activities specified by one or more control links. Parallelism is automatic since control links are processed simultaneously.

Data Link

A data link specifies the flow of data from one activity to another. Essentially, the output message returned by an activity becomes the input message for a downstream activity. If the messages match (same elements, etc.) then the mapping is done automatically. WSFL allows you to map parts of multiple messages into the input of an activity. For example, one activity calculates the price of an item and another determines the warehouse from which to ship the product. Both pieces of data could be combined and sent to a downstream activity. XPATH expressions are used to specify how to map data. If the same data arrives at an activity (two different upstream processes provide a price, for example), WSFL defines a set of mapping rules to determine which piece of duplicate data "wins" and is sent to the activity.

Conditional Links

Every control link can have a transition condition. The transition condition is an XPATH expression that evaluates to true or false. During runtime, all transition conditions are evaluated to determine whether a branch from an activity is actually traversed. An example transition condition might be "not(Credit_Check_Approved = 'Y')" If the credit check isn't approved, the link will be valid and the engine runs the downstream activity.

Join Activities and Conditions

An activity that has more than one control link entering it is called a *join activity*. Join activities have *join conditions*. If the join condition of an activity evaluates to true, the activity will be executed. If not, it won't. Join conditions are needed because each link is unaware of other links. If you want an activity to fire only if two or more previous activities returned a certain result, you would put that condition in the join condition. This would be done, for example, if an activity should only be executed if the customer's credit is OK and there is inventory in stock.

You can't use a transition condition because each link would only be aware of its previous activity's data. In this case you need information from both activities to make a decision. Figure 1 shows two activities that merge into one. Once the individual link statuses are (the credit is ok, inventory is in stock), the "create invoice" join condition will be evaluated. The join condition enforces that both links must be true for the process to proceed. (If either were false, then you would never get to the activity anyway.)

Join conditions can have a *join evaluation* property that is "deferred" or "immediate." The default is deferred and means that all links coming into an activity must be valid and completed before the join condition is evaluated, ensuring that all activities have finished. The result is synchronized execution. If the evaluation is immediate, the join condition is evaluated each time a link is completed; this way, an activity can fire before all the previous activities are completed. An example of this would be a supplier inventory check, where at least one supplier must confirm an item is in stock. The flow can continue once one supplier confirms the item is in stock without waiting for the other suppliers' responses.

Exit Conditions

Activities have exit conditions that are evaluated when the activity has completed. If it evaluates to false, the activity will execute again. This construct allows you to cre-

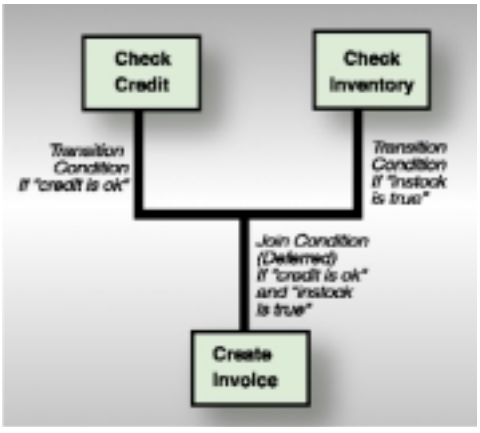


FIGURE 1 | **Join conditions**

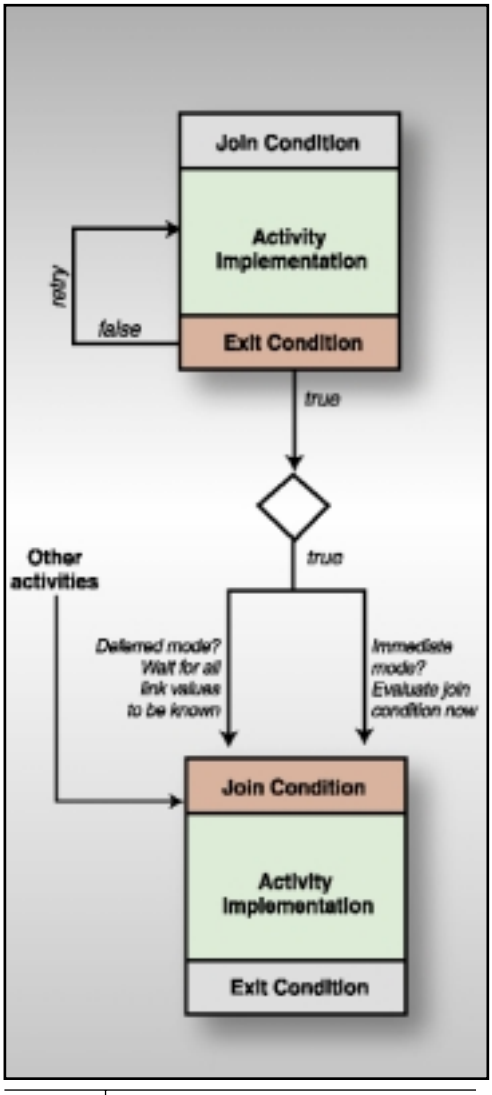


FIGURE 2 | **Process execution model**

ate a do-until loop within an activity (yes, you can create an endless loop). If you want to loop through complex functionality, then you can use the recursive composition feature of WSFL, which allows any complex process to also be a Web service used by another WSFL process.

FlowSource and FlowSink

When you define a process, you also define the incoming and outgoing message for that process. These are referred to as the *FlowSource* and the *FlowSink*. The FlowSource is always available to any process and any activity's output can be written to the FlowSink (see Figure 2).

Start and End Activities

Any activity without an incoming link is a *start activity*. Any activity with no outgoing links is an *end activity*. When a process is initiated, all start activities are executed and passed data from the FlowSource. When all end activities have been executed, the process is complete.

Logic Construction

It's possible to model almost anything you need in WSFL using conditional links, join conditions, and exit conditions. Traditional workflow systems use things like XOR-links, AND-links, and OR-links, which tend to be implemented in code; one of the problems with these is that if the business model changes, you might need to change the link type, which would require recoding of a link. With WSFL, you merely add a link and change any needed conditions; rapid process maintenance is a major benefit of WSFL.

The Global Model

Global models define how individual flow models are actually implemented. The flow model doesn't have mappings to particular Web services. Think of activities as placeholders where Web services will be plugged in. The global model lets you link individual activities with actual Web services.

In the global model, you can define service provider types, port types, service providers, service locators, and plug links. Going into detail on these would require another article twice the size of this one because of the abstraction they provide. Plug links are used to map specific activities in the flow model to the actual Web services that will be used. Services can be hard-coded, looked up from UDDI services, or even specified by the data that is used within the process itself.

Given the state of Web services today, I expect that most implementations of business processes will use hard-coded services references and that some of the more flexible portions of the global model will not be used for some time. However, WSFL does provide extreme flexibility in how actual services can be specified and called during execution time and it's good that WSFL anticipates the dynamic nature of Web services.

Another thing the global model allows is the definition of the public interface of the flow itself. It contains mappings to define what the resulting WSDL of the model will look like and allows outside services to call or return data to specific services within the flow as required. For example, you might have an activity in a flow that merely waits for notification from some outside process. That outside process will need a way to get to the specific activity managed by the WSFL engine.

Real World WSFL

As the first real-world commercial implementer of WSFL, my company faced a few hurdles and raised quite a few issues with IBM on where the spec itself should go. The bottom line is that we believe WSFL 1.0 is an excellent start toward a comprehensive specification for the orchestration and assembly of Web services into processes. Our overall experience was quite positive and we believe that early adoption of WSFL has given us a clear competitive advantage in the Web services orchestration marketplace.

There are a few areas in which WSFL is currently lacking, however. Because of these limitations, any real-world implementation must extend the current version of the specification. WSFL lacks some features needed for effective B2B implementations; many of these features are scheduled to be expressed in something called WSEL – Web Services Endpoint Language. WSEL describes things like timeout values, retry values, and other quality-of-service information needed to make B2B interactions work. WSEL has not been released, but the WSFL specification provides some high-level guidance as to where WSEL is heading.

WSFL doesn't handle asynchronous recursive processes very well. In BPM circles these are known as fan-in and fan-out scenarios; while they can be implemented in WSFL, they are complex and error-prone and require special services to make the processing work. I believe WSFL should have provisions for such requirements.

Finally, WSFL doesn't address some B2B and human workflow semantics, such as addressees, correlation IDs, and priority levels. These are required so that waiting processes can be delegated to various individuals and groups within an organization. WSFL's provision for a process ID is not sufficient.

Any major commercial implementation based on WSFL will have to have extensions to the base WSFL in order to be viable. Fortunately, the standards process works and all of the current weaknesses in WSFL are being addressed.

Summary

My company found WSFL to be a very usable and well thought–out specification and while we did have to extend it for commercial implementation, we were able to do so in a way that does not violate any of WSFL's basic premises and philosophies. When the WSEL specification emerges, vendors should not have problems updating their engines to handle it.

If you're serious about Web services orchestration and moving away from hard-coded business processes, I highly recommend choosing a WSFL-based process management system.

Web Services Journal
FOCUS ON BPM

written by Rob Cutlip & Ric Telford

The Orchestration of Business Processes

An example of Web services and the business process view

As the usage of Web service interfaces grows, we're starting to see the emergence of some key sectors where Web services will make a huge impact. Much as Java found its sweet spot on the Web application server, Web services will find its own sweet spots. Grid computing and e-business "building blocks" are two examples of Web services sweet spots. Another area where we see Web services as critical is Business Process Management (BPM).

Rob Cutlip is a software and solutions architect with the IBM Software Group. For the past six years, he has focused on building best-of-breed e-business solutions. An author, inventor, and sometimes nice guy, Rob has 15 years of experience in both technical and managerial positions with Fortune 500 companies. CUTLIP@US.IBM.COM

Ric Telford is a director of technology in the IBM Software Group. His primary focus is on service provider solutions and B2B solutions that leverage Web services standards. In his 19-year career, Ric has driven a number of software initiatives at IBM, including imaging, mobility, networking, security, and directory. Ric is currently coauthoring a book on Web services. RTELFORD@US.IBM.COM

BPM is the ability to orchestrate and control the execution of a business process across heterogeneous systems. BPM allows a user to see the components of the infrastructure from a business process view, rather than as a set of applications and databases. Part of orchestrating a business process is being able to manage the process even when it goes outside the enterprise, a common occurrence in today's e-business environment. When a process flows outside the enterprise, it needs a bridge from the private process to the public process. It is at this bridge point that Web services become important.

This article describes the role of Web services in the orchestration of business processes. We will look at two examples: (1) incorporating information from a public process into a private process, and (2) exposing a private process outside the enterprise. This process flow is described using a combination of technologies, most notably the BPM product from CrossWorlds, the company recently acquired by IBM. Using CrossWorlds tools and Web service technologies, it's possible to define a business process, such as a purchase order, that has both a private process and a public process.

The Basics

Before we get into the scenarios, let's review the basics of BPM first. In order to orchestrate a business process, you first define a common data format to be used between applications, and then define a workflow to move that data from one step of the process to the other. In CrossWorlds terms, this is done using a combination of *connectors*, *business objects*, *maps*, and *collaborations*. Connectors are the interfaces into the specific applications – wrappers around whatever the key APIs are that act as interfaces into the product. *Business objects* are the data – the information you want to extract from one application and move to another. *Maps* define the relationship between the application data ("Application Specific Business Object") and the common format of that data ("Generic Business Object"). Finally, a *collaboration* puts it all together – it is the collection of connectors, business objects, and maps put together in order to effect a business process.

Figure 1 illustrates typical business process flows using IBM CrossWorlds. Within the enterprise, applications communicate with the CrossWorlds Interchange Server or ICS via connectors (1). For outbound flows,

the collaboration (business process) can leverage a Web services gateway by converting the business object into a SOAP message (2). The reverse can be handled as well for inbound SOAP messages (3).

In the following pages, we'll discuss some practical examples of leveraging Web services within two common BPM scenarios.

Scenario 1: Using Web Services From Within a Business Process

As mentioned above, business process definitions take the form of a flow through a set of application and data transformation steps. For this first scenario, let's use the example of a simple business process called *OrderProcessing*. An appliance company, StoneCold Refrigerators, has a Web site that allows customers to order replacement parts directly from the company via its Web site. A parts catalog is published, customers can purchase parts, and the orders are stored in a staging database by the Web front end. Once an entry is created in the database, a collaboration is triggered and the *OrderProcessing* process begins.

StoneCold has a collaboration that extracts the record from the staging database, converts it to the generic business object format, and then performs a number of steps:

- It updates the company's CRM system with the information, allowing customer support to see the order.
- It identifies the vendor that supplies the part and, if the vendor is set up to fulfill orders directly, StoneCold passes the order along for fulfillment (for this scenario we will assume this is the case). The vendor, EverPure, provides replaceable filter cartridges for StoneCold's water-in-the-door system.
- It updates billing, order status tables, etc.

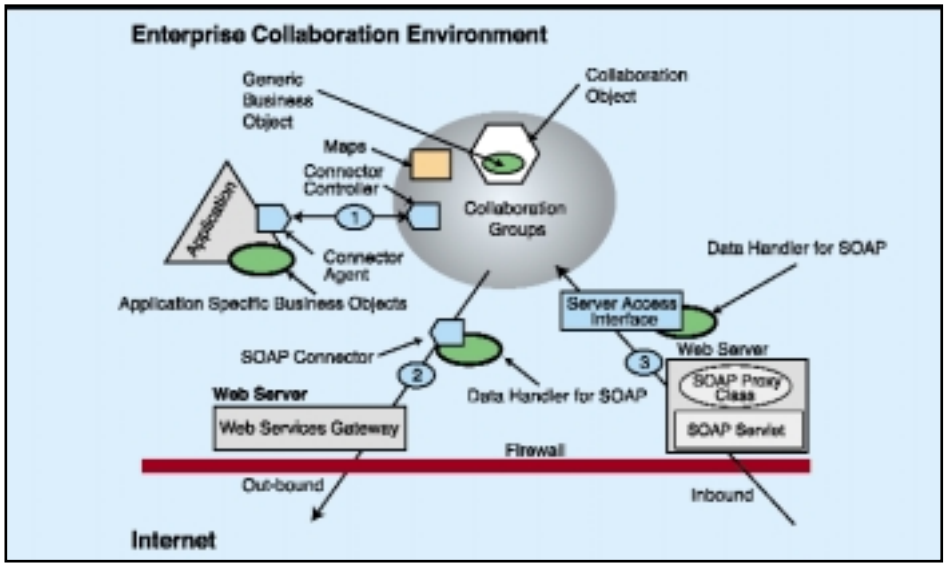


FIGURE 1 | **Business process flows: enterprise collaborations and Web services**

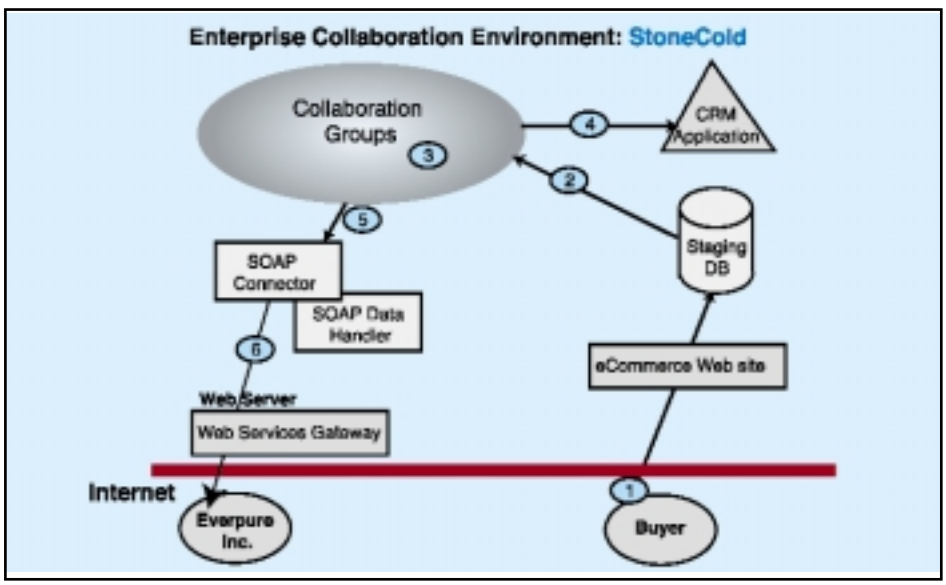


FIGURE 2 | **StoneCold's enterprise collaboration accessing an EverPure, Inc., Web service**

StoneCold's business process now extends beyond the scope of its own systems. The business process, *OrderProcessing*, now includes EverPure's IT system. Web services provide the interface definition between the two companies' processes. It's immaterial to StoneCold how EverPure's process is built, due to the nature of Web services. EverPure exposes this process using a WSDL definition and a SOAP interface, allowing StoneCold to transact directly with EverPure's process (we will examine EverPure's *OrderStatus* business process in Scenario 2).

In CrossWorlds, the connector for Web services is metadata-driven. The connector uses the metadata encoded in the business objects it processes, rather than relying on instructions coded in the connector. The connector calls the SOAP data handler that converts the CrossWorlds generic business object into an appropriate SOAP request message. The SOAP Connector then calls the EverPure, Inc., Web service using the

corresponding URI. Once the destination Web service has processed the request, it sends back a SOAP response message. The SOAP connector using the SOAP Data Handler converts the message back into a generic business object and passes it to the appropriate collaboration.

The end-to-end transaction is summarized in Figure 2. Purchases are staged in a database (1) from which a collaboration periodically polls for new entries (2). Part of the collaboration process is to identify appropriate suppliers for various items within an order (3) and update StoneCold's CRM system with pertinent customer information (4). The CrossWorlds collaboration calls the SOAP connector associated with the EverPure, Inc. Web service (5). The SOAP Connector then calls the EverPure, Inc. Web service using the corresponding URI (6).]

Scenario 2: Exposing a Business Process as a Web Service

In the first scenario we saw how to invoke a Web service from an internal business process. In this scenario, we'll examine how to expose a business process as a Web service, thus allowing the business process to be kicked off by invocation from a public process view.

As mentioned above, EverPure has their own process, *OrderStatus*, for allowing customer applications to perform order status checks. Like StoneCold, EverPure, Inc. uses CrossWorlds for business process management. Their OrderStatus process is defined as a CrossWorlds collaboration.

The OrderStatus collaboration is exposed to the Internet via a WSDL definition using CrossWorld's tooling (see below). The client application at StoneCold sends a SOAP request to the URI specified in this WSDL file. A UDDI registry, whether public or private, may be used to discover the service.

The interface between a business process and Web services in CrossWorlds is provided by a SOAP proxy class and servlet. The Web server routes the message to the SOAP servlet. The servlet in turn calls the appropriate SOAP proxy class. The proxy class then calls the CrossWorlds Server Access Interface (SAI), which converts the SOAP message into a generic business object (GBO). The SAI then calls the appropriate collaboration with the GBO. Once the collaboration has completed processing, it returns the response message to the Server Access Interface. The SAI then calls the data handler to open the SOAP message format, returning the message to the SOAP proxy class and SOAP servlet.

As Figure 3 shows, the client application at StoneCold sends a SOAP request message (1) to the URI specified in the WSDL file that describes the collaboration exposed as a Web service. The proxy class then calls the Server Access Interface (2) The SAI calls the SOAP data handler that converts the SOAP message into a generic business object (3). The SAI then calls the appropriate collaboration with the generic business object (4).

Being able to expose a business process as a Web service is becoming an increasingly important function of a BPM system. Over time, the mechanics of doing this will become easier and the implementation options more diverse. Today, with CrossWorlds, as in Scenario 2, the primary way is to use the Web Services Generation Utility to produce the code and definitions needed to expose a specified collaboration and port as a Web service. In the remainder of this article, we'll examine the basic steps required to expose a CrossWorlds business process to the outside world as a Web Service.

- ***Expose the collaboration:*** Once you've identified the collaboration you'd like to externalize as a Web service, you need to perform key foundation tasks: (1) change the collaboration's trigger port to allow external output, (2) develop maps to support the conversion of existing business objects to SOAP messages, and (3) make sure the business objects are properly populated.
- ***Develop business objects:*** Using the Business Object Designer tool, you'll need to create new input and output business objects to externalize the collaboration as a Web service. Business objects can be considered as messages.
- ***Create Data Handler metaobjects:*** As we saw in Scenario 2, the SAI calls the SOAP Data Handler to transform SOAP messages into business objects and in turn transform business objects into SOAP messages. To perform these tasks, the data handler metaobjects are required.

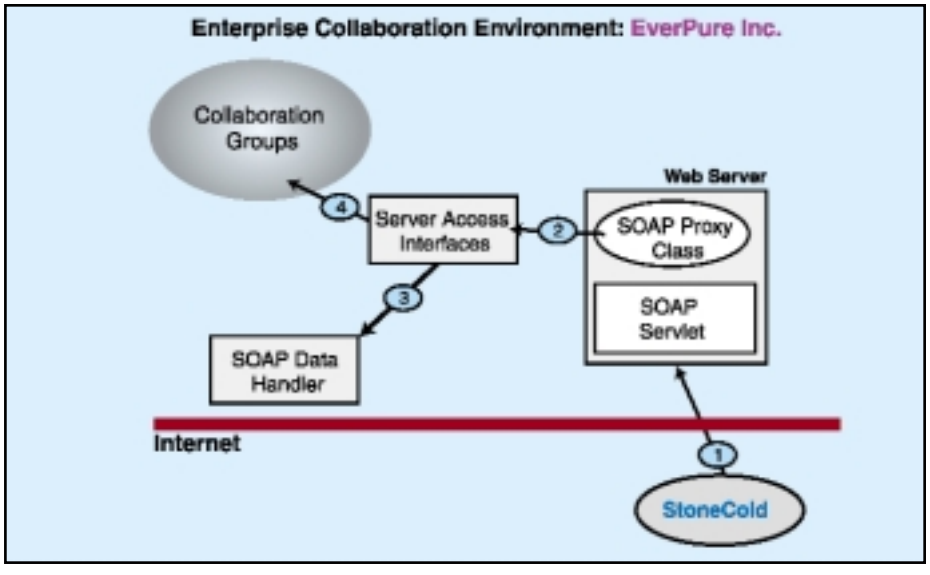


FIGURE 3 | **StoneCold accessing EverPure collaboration environment via Web services**

FIGURE 4 | **A Child Metaobject**

Metaobjects contain transformation information, as well as an associated child metaobject as shown in Figure 4. The child metaobject defines the business object, the verb (create, retrieve, update, or delete), and the body name, e.g., getOrderStatus. The metaobjects are supplied to the Web Services Generation Utility in our example and can be seen in Figure 5, under Operation Details.

- ***Run the Web Services Generation Utility:*** In order to expose this collaboration as a Web service, the Web Services Generation Utility is invoked with a set of parameters to create the requisite WSDL file and SOAP interfaces. In Figure 5, once we've connected to the CrossWorlds server, (prometheus888), we specify the MIME type, collaboration name, collaboration port, and proxy class name.

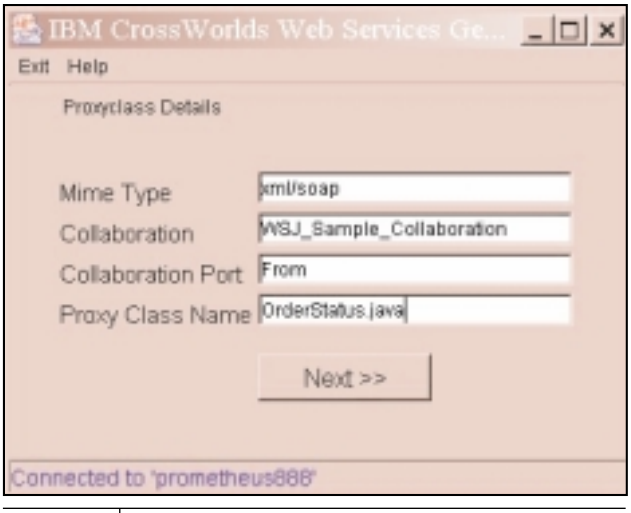


FIGURE 5 | **Specifying Proxyclass details**

The default MIME type is XML/SOAP. The MIME type tells the proxy class which Data Handler to use for converting the SOAP message into a business object.

The collaboration name is the executable collaboration object that the proxy class will call. The collaboration port, configured earlier to receive input from an external process, tells the proxy class where to send messages.

After the generation process is completed, the Web Services Generation Utility displays the six files shown in Figure 6.

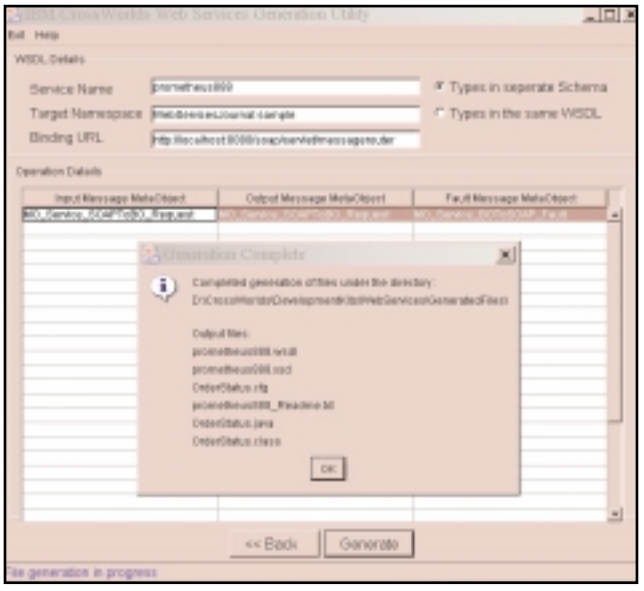


FIGURE 6 | **Web Services Generation Utility**

The WSDL service file shown in Listing 1 describes the Web services. The WSDL file documents five namespaces as attributes and uses seven elements to define network services.

prometheus888.xsd is the XML Schema file for the business objects used by the Web service. This file contains the type definitions for each business object referenced in the metaobjects specified earlier. The OrderStatus.cfg file contains the name-value pairs used by the proxy class and the prometheus888_Readme.txt file contains deployment information for the proxy and its methods on SOAP-enabled Web servers.

The last two files generated are the OrderStatus.java and OrderStatus.class files. The Java source code is provided for the proxy class to facilitate customization, if needed. The generated OrderStatus.class is ready to deploy.

- ***Deploy the Proxy Class:*** Individual deployment requirements vary depending on which SOAP-enabled Web server is used. Any Web server that listens for SOAP requests (SOAP 2.2) should be able to make the appropriate calls to the Java proxy class.

Conclusion

Business process management tools such as CrossWorlds have given companies the ability to start thinking in terms of the process they are trying to perform, rather than in terms of the applications they are using. As more and more business processes start extending beyond the walls of the company, there will be an increasing need for seamless transaction

flows between enterprises. Web service interfaces provide the perfect set of formats and protocols for making these transactions both standard and efficient. By tightly coupling the power and capabilities of BPM tools and Web services interfaces, a company can reap the real benefits of e-business.

Listing 1: prometheus888.wsdl

```
<?xml version="1.0"?>
<definitions name="prometheus888"
  targetNamespace="WebServicesJournal:sample"
  xmlns:tns="WebServicesJournal:sample"
  xmlns:typens="WebServicesJournal:sample"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/1999/XMLSchema"
  xmlns="http://schemas.xmlsoap.org/wsdl/">

 <wsdl:import namespace="WebServicesJournal:sample"
   location="prometheus888.xsd"/>

 <wsdl:message name="getOrderStatusMessage">
  <wsdl:part name="OrderId" type="xsd:string"/>
  <wsdl:part name="OrderNumber" type="xsd:string"/>
  <wsdl:part name="CustomerId" type="xsd:string"/>
  <wsdl:part name="OrderStatus"
type="typens:ArrayOfSOAP_OrderStatusLine"/>
 </wsdl:message>

 <wsdl:message name="getOrderStatusResponseMessage">
  <wsdl:part name="OrderId" type="xsd:string"/>
  <wsdl:part name="OrderNumber" type="xsd:string"/>
  <wsdl:part name="CustomerId" type="xsd:string"/>
  <wsdl:part name="OrderStatus"
type="typens:ArrayOfSOAP_OrderStatusLine"/>
 </wsdl:message>

 <wsdl:message name="soap:faultMessage">
  <wsdl:part name="faultcode" type="xsd:string"/>
  <wsdl:part name="faultstring" type="xsd:string"/>
  <wsdl:part name="faultactor" type="xsd:string"/>
  <wsdl:part name="detail"
type="typens:SOAP_OrderStatus_Request"/>
 </wsdl:message>


 <wsdl:portType name="prometheus888PortType">
  <wsdl:operation name="getOrderStatus">
   <wsdl:input message="tns:getOrderStatus"/>
   <wsdl:output message="tns:getOrderStatusResponse"/>
   <wsdl:fault message="tns:soap:fault"/>
  </wsdl:operation>
 </wsdl:portType>

 <wsdl:binding name="prometheus888Binding"
type="tns:prometheus888PortType">
  <soap:binding style="document" transport="http://schemas
    .xmlsoap.org/soap/http"/>
  <wsdl:operation name="getOrderStatus">
   <soap:operation soapAction=""/>
   <wsdl:input>
    <soap:body use="encoded"
      namespace="urn:OrderStatus"
      encodingStyle="http://schemas.xmlsoap.org/soap/
        encoding/"/>
   </wsdl:input>
   <wsdl:output>
    <soap:body use="encoded"
      namespace="urn:OrderStatus"
      encodingStyle="http://schemas.xmlsoap.org/soap/
        encoding/"/>
   </wsdl:output>
   <wsdl:fault>
    <soap:body use="encoded"
      namespace="urn:OrderStatus"
      encodingStyle="http://schemas.xmlsoap.org/soap/
        encoding/"/>
   </wsdl:fault>
  </wsdl:operation>
 </wsdl:binding>

 <service name="prometheus888">
  <port name="prometheus888Port"
binding="tns:prometheus888Binding">
   <soap:address
location="http://localhost:8080/soap/servlet/messagerouter"/>
  </port>
 </service>
```

Download the code at

sys-con.com/webservices

Reviewed by Paul Kaiser

About the Author:
Paul Kaiser is a consultant for Info Technologies, Inc., in New Jersey, where he focuses on application integration. PAULKAISER@YAHOO.COM.

Sonic Software's SonicXQ Release 1.0

A flexible integration tool

Enterprise application integration (EAI) has been getting a lot of attention lately. Regardless of size, companies need to get their internal applications to work together and expose them to their trading partners in order to gain a competitive advantage. Integration comes in different forms, from simple data exchange to fully integrated business processes. In between is a myriad of solutions addressing the whats and hows that come up during any integration-planning phase. Sonic Software recently released SonicXQ 1.0, a product aimed at IT groups trying to solve the integration problem.

Service-Based Architecture

SonicXQ provides a framework for integrating a variety of application types, but the key to putting SonicXQ to work for you is to have the required functional components packaged as reasonably coarse-grained, self-contained services. This is a good practice in general and more so for integration efforts.

Service-based architectures (SBA), pre–Web services, divided the world into two groups, service providers and service consumers. With the advent of UDDI, service registries have been added to the picture. But simply put, an SBA provides a model through which providers can register and be discovered and invoked by consumers.

Product Architecture

SonicXQ is essentially an adapter framework bound to a messaging backbone. The framework uses a service container to implement the bindings to the message backbone. On an initial examination, it looks like an integration broker. But unlike other integration broker implementations, it was designed to be lightweight and deployed as a set of distributed containers that support autonomous services and permit them to be aggregated into business processes.

Service Container

The service container is similar in concept to the containers in J2EE. It provides an execution environment for one or more business services and/or adapters. The service container provides the bridge between the message backbone and the services running in the container.

Sonic provides an extensible, explorer-style tool called Sonic Explorer (see Figure 1). You can use it to manage the directory and containers via Java Management Extensions

sonic SOFTWARE

CONTACT:
Sonic Software
14 Oak Park
Bedford, MA 01730
Phone: 781 999-7100 (sales)
Web:www.sonicsoftware.com/products/sonicxq.htm

PRICING:
Developer version: $3,750 / CPU
Deployment version: $5,000–10,000 per CPU
A free 30-day evaluation edition of the software is available.

TEST ENVIRONMENT:
Hardware: Dell XPS 600 MHz Pentium III, 256MB RAM
OS: Windows NT 4.0 – SP6a

SUPPORTED PLATFORMS:
Operating Systems
Solaris 2.8
Windows NT 4.0 – SP6a
Windows 2000 – SP2

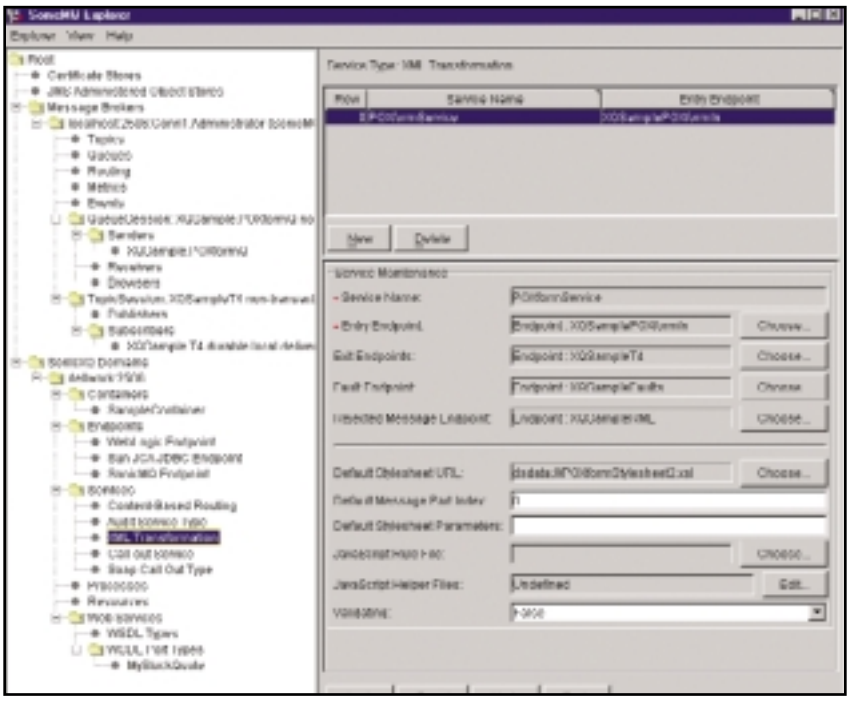


FIGURE 1 **Sonic Explorer showing an XML transformation service**

FIGURE 2 | **A mock inbound purchase order process**

(JMX). Management messages are sent over the same messaging backbone as the application messages.

Messaging

A SonicMQ message broker must be available to connect to when installed. No other messaging products are supported. SonicXQ ships with MQ so customers do not have to purchase SonicMQ separately or have it installed currently. SonicMQ is a robust product and provides bridges to other messaging platforms; however, companies that already have a messaging infrastructure may not want to add another product to the mix.

Services and Adapters

SonicXQ comes with three service types defined: content-based routing, transformation, and call-out. The content-based routing service utilizes JavaScript and XPath expressions to examine the content of the message and determine to where the message should be routed. SonicXQ provides "helper" JavaScript functions to simplify access to message headers and content. In addition, routing rules can call out to simple Java objects, EJBs, and resource adapters.

The transformation service provides an execution environment for XSLT stylesheets. It can produce one or more messages from a single incoming message. The service provides XSLT extensions that permit the use of external Java classes to enhance the transformation process.

If these don't fit the bill, you can write your own services. And you can also add new service types. Sonic Explorer can be configured to understand the elements that compose your types.

External Resource Integration

Large-scale enterprise information systems (EIS) like SAP and PeopleSoft have proprietary APIs and are moving toward providing open, standards-based, access mechanisms. Two such mechanisms supported by SonicXQ are the J2EE Connector Architecture (JCA) and Web services.

SonicXQ provides support for the required elements of the JCA Service Provider Interface as part of the service container implementation. It does not provide implementations of the optional security and transaction interfaces.

SonicXQ provides a JCA adapter to connect the framework to stateless session EJBs running in a WebLogic application server. The request to invoke the EJB method must be encoded in a SOAP message and sent to the endpoint associated with the bean.

Prerelease support for receiving asynchronous messages from a JCA adapter is included. This requires that the developer implement an interim interface that supports the JSR 112 standard. Once JCA 1.5 is released, the interface will be deprecated and developers should use the appropriate javax.resource interface.

SonicXQ provides support to expose services as Web services. It simplifies the creation of WSDL files by allowing the developer to import WSDL types from XML schema definition files and assemble the WSDL ports through the Sonic Explorer interface. It can then create the WSDL interface, implementation, or a combined file.

Process Management

Figure 2 shows a simple example of services aggregated into a process. A process is a predefined flow of services or other processes. The flow is defined by a message itinerary. SonicXQ creates and maintains the association between an itinerary and the messages that flow through the process.

This was a nice feature to see. You can create processes in SonicXQ without any prior knowledge when developing the services. The more you design your services as independent functions, the more flexibility you have in configuring processes.

Security Features

SonicXQ supports the following security features:

- HTTP/HTTPS support
- Message encryption at the JMS level (40- & 56-bit MD5/DES)
- 128- & 168-bit SSL support for message transport
- User authentication via username/password and certificates
- Message integrity and privacy at the queue & topic level
- Service container configuration file encryption
- Support for J2EE security protocols

Availability, Scalability, Reliability, Recoverability

SonicXQ allows the same service to be hosted in multiple containers while maintaining connections to the same endpoints. Thus, the degree to which the service must scale will dictate in how many containers and on what size hardware the service will be hosted. This also affords good availability. The degree to which a service must survive any single point of failure will dictate the number and class of machines on which the container runs, and the network topology. SonicMQ also supports a clustered environment to address these issues.

Also, the weight of reliability and recoverability falls on the shoulders of SonicMQ. A SonicXQ service is configured with a quality-of-service level that is directed at the messaging layer. This quality level may be *best effort, at least once,* or *exactly once.* Using *exactly once* creates a fully transacted session.

Other Topics

Both the SonicMQ and SonicXQ products installed smoothly on my Windows NT computer and, for the most part, all of the samples ran cleanly. I found the Developer's Guide useful to get to know the tool; it includes quite a bit of detail. However, a more complete explanation of concepts and more extensive samples would be helpful, especially for processes.

The service container does not support JTS. If it's necessary, a possible workaround may be to develop the functionality that requires JTS as a session EJB and integrating it via the adapter support. It would be nice to be able to develop a more direct integration, though.

Conclusion

SonicXQ is a flexible integration tool that has a good balance between ease of use and extensibility. Coupled with SonicMQ and its HTTP interfacing capabilities, SonicXQ can connect many of the applications companies have today and those they will have tomorrow. If you need to integrate applications, new or existing, consider adding SonicXQ to your list of candidate tools.

Web Services Journal
FOCUS ON BPM

written by Gunjan Samtani & Dimple Sadhwani

Business Process Management & Web Services

Design, deploy, execute, analyze, and optimize business processes as Web services

The journey of business process management (BPM) from technologies such as document flow routing to service-oriented architecture (SOA)–based Web services has been a process of evolution rather than revolution.

Gunjan Samtani, divisional vice president of information technology at UBS PaineWebber, has several years of experience in the management, design, architecture, and implementation of large-scale EAI and B2B integration projects. He's the primary author of B2B Integration: A Practical Guide to Collaborative E-Commerce (www.worldscientific.com/books/economics/p263.html). GSAMTANI@UBSPW.COM

Dimple Sadhwani, senior application architect at Island ECN, has many years of experience working for financial and telecommunication companies on large scale trading systems, CRM applications, Internet/intranet portals, and client/server applications. As well as coauthoring *B2B Integration: A Practical Guide to Collaborative E-Commerce*, she has also authored several articles in the field of Web services.
DSADHWANI@ISLAND.COM

OUR PREDICTIONS

- Web services will play a major role in business processes design, deployment, execution, maintenance, and optimization.
- The use of Web services within companies for enterprise and business-to-business application integration will be as much about business process management and reengineering as it will be about technology.
- The Web services standards for BPM, such as WSFL, will have to evolve and mature before any significant use of this technology occurs within enterprises.

In this article, we'll discuss the convergence of Web services and business process management, how Web services alleviate some of the core problems with BPM, and how Web services can help companies evolve into process-based organizations leading to increased efficiency and lower operational costs.

It's all about processes, after all.

Business process management (BPM) enables enterprises to automate and integrate the disparate internal and external corporate business processes. It does so by supporting dynamic process topologies that allow the boundary between processes and participants to be determined either statically or dynamically on a real-time basis. Further, its implementation provides every corporation the opportunity to redefine and automate core business processes, which results in streamlined business operations and reduced cost.

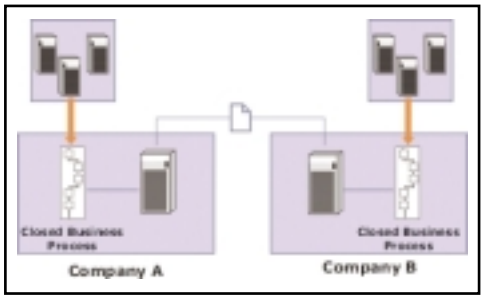


FIGURE 1 | **Closed business process**

BPM for enterprise application integration (EAI) enables companies to achieve internal systems that are truly integrated using automated workflows. The business processes that control information flow by coordinating interactions with business applications and systems within an organization are called private or closed (see Figure 1).

BPM for business-to-business integration (B2Bi) focuses on how business partners can refine their business processes so that the applications supporting them can be seamlessly integrated. With effective BPM, companies can become part of a unified business process flow and unified supply chain. Unified workflows allow the dynamic sharing of state information among trading partners through which all communication can be tracked and recorded. Since B2B transactions can span multiple days, unified workflows become critical in ensuring the completion of automated business transactions. The external business processes that control interactions among independent trading partners are called public or open (see Figure 2).

Web Services: Business Operations by Definition

By definition, Web services describe a collection of business operations that are network-accessible through standardized XML messaging based on Simple Object Access Protocol (SOAP), a standard messaging layer used to exchange XML documents in Web services. In other words, Web services expose applications supporting business operations, encapsulating business logic, and accessing business data over the network or Internet using interfaces that can be invoked. The main advantage of Web services is that companies can use these interfaces for process management, logic transformation, and integration for legacy and packaged applications, instead of writing nonstandards-based custom code for each application.

A Web service can be implemented as its own business process, or it may be composed of many business processes (both public and private) with each business process being implemented as a Web service in itself. Each activity that is part of the workflow of a business process is logically linked to a Web service.

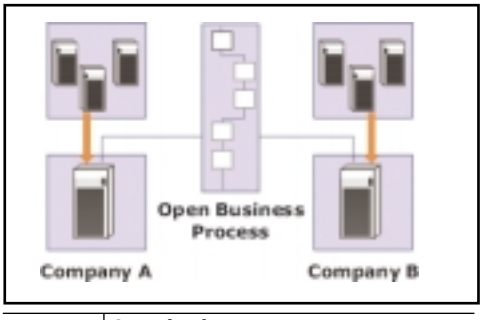


FIGURE 2 | **Open business process.**

Web Services Alleviate Complexities of BPM

Although BPM poses difficulties with both soft (such as human resistance and behavioral issues) and hard elements (such as technology, tools, and techniques), we will limit our discussion to technological aspects and complexities and how Web services alleviate some of those.

- ***Easier BPM:*** Web services help to clearly separate business process logic and the participating business services, thereby making the development, execution, and management of these services much easier. Apart from easier application integration, user-centric Web services make the human intervention (which may be an activity in a business process workflow) easier by providing personalization, interface customization, and support for multiple languages, greatly enhancing the user experience.
- ***Easy integration with applications:*** A typical business process may be supported by multiple diverse applications such as ERP, CRM, SCM, and legacy systems. It's virtually impossible to manage a workflow and execute the different tasks associated with it, which may require using other systems' APIs or exchanging messages with them, unless the underlying technology provides easy integration facilities. XML-based Web services are the ideal technology for BPM, as they allow applications to communicate across the Internet in a platform- and language-independent fashion.
- ***Based on open standards:*** Since B2Bi requires integration of business processes across corporate boundaries using exchange of XML-based documents or messages, the communication among different systems should be based on open standards. Web services fully leverage open standards, including Hyper Text Transfer Protocol (HTTP); eXtensible Markup Language (XML); Simple Object Access Protocol (SOAP); Web Services Description Language (WSDL); and Universal Discovery, Description, and Integration (UDDI). Application-centric Web services enable companies to integrate business processes without the constraints of proprietary infrastructures, platforms, and operating systems.

Enabling Technologies

The key technologies and specifications that enable the orchestration of business processes as Web services include WSFL, BPML, XLANG, and ebXML. These standards together have the same significance to BPM as XML has to e-commerce.

- ***Web Services Flow Language (WSFL):*** An XML language for the description of Web services compositions as part of a business-process definition. WSFL considers two types of Web services compositions: the first type specifies an executable business process resulting in a composition that describes how to achieve a particular business goal; and the second specifies business collaboration, i.e., the interaction pattern of a collection of Web services.
- ***Business Process Modeling Language (BPML):*** An XML-based, open-standard metalanguage for the design, definition, deployment, and management of business processes that span multiple applications, corporate departments, and business partners. BPML is actually an XML schema that provides a standard way to model mission-critical business processes.
- ***XLANG:*** A Microsoft initiative, it's an XML-based language that describes the logical sequencing of business processes and their implementation by using various application services.
- ***Electronic Business XML (ebXML):*** Jointly sponsored by the United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT) and the Organization for the Advancement of Structured Information Standards (OASIS), ebXML provides a standard method to exchange business messages, conduct trading relationships, communicate data

Subscribe *Now*

for Instant Access to CF Advisor

CFAdvisor.com!

12 months $89
24 months $169

Now on CD!

THE MOST COMPLETE LIBRARY OF EXCLUSIVE *CF ADVISOR* ARTICLES!

3 YEARS
40 ISSUES
200 ARTICLES
ONE CD

ONLY $71.99

Since May 1998, CF Advisor™ has provided the CF community with in-depth information that quickly pays dividends on the small subscription investment. For just $89.00, you get 12 full months of uninterrupted access to CF Advisor™ – including:

- usable code
- skill-building articles
- tips
- news updates
- conference reports
- interviews with industry movers-and-shakers.

You get full access to our sizable archives covering such subjects as:

- custom tags
- functions
- variables
- framesets
- style sheets
- structures
- javascript usage
- debugging techniques
- programming practices and much, much more...

Subscribing now guarantees that you'll immediately receive access to everything you need for keeping up-to-date with ColdFusion, no matter where you are. All the content is available exclusively over the Net and is regularly updated throughout the month.

THE MOST COMPLETE LIBRARY OF EXCLUSIVE *CF ADVISOR* ARTICLES ON ONE CD!

"The Complete Works"

CD is edited by *ColdFusion Developer's Journal* Editor-in-Chief Robert Diamond and organized into 21 chapters containing more than 200 exclusive *CF Advisor* articles.

Easy-to-navigate HTML format!

E-Commerce
Interviews
Custom Tags
Fusebox
Editorials
Databases
News
CF & Java
CFBasics
Reviews
Scalability
Enterprise CF
Wireless
Verity
Source Code CF
Applications
Object-Oriented CF
WDDX
Upgrading CF
Programming Tips
CF Tips & Techniques
ProgrammingTechniques
Forms

Order Online and Save 10% or More!
www.JDJSTORE.com
OFFER SUBJECT TO CHANGE WITHOUT NOTICE

SYS-CON MEDIA

135 CHESTNUT RIDGE ROAD
MONTVALE, NJ 07645

WWW.SYS-CON.COM

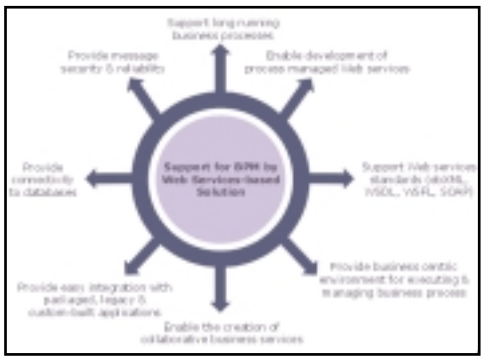


FIGURE 3 | **Features of a BPMS**

in common terms, and define and register business processes.

Business Process Management Systems (BPMS)

All the Web services servers, such as Microsoft .NET-based servers, J2EE-based application servers, integration brokers, and pure Web services servers, have to include or provide connectivity to a business process management component. This component, also known as BPMS, supports the full life cycle of process design, deployment, execution, analysis, and optimization. Any BPMS supporting Web services must provide the following (see Figure 3):

- Support long-running business processes
- Enable development of process-managed Web services
- Support Web services standards (ebXML, WSDL, WSFL, SOAP)
- Provide business-centric environment for executing and managing business processes
- Enable the creation of collaborative business services
- Provide easy integration with packaged, legacy and custom-built applications
- Provide connectivity to databases
- Provide message security and reliability

BPMS is increasingly becoming a core segment of the major integration platforms available commercially. It's worth mentioning that a BPMS should support all of the leading XML standards that define the vocabulary and/or the business processes, for instance, RosettaNet for electronic component industry and OBI for purchase management.

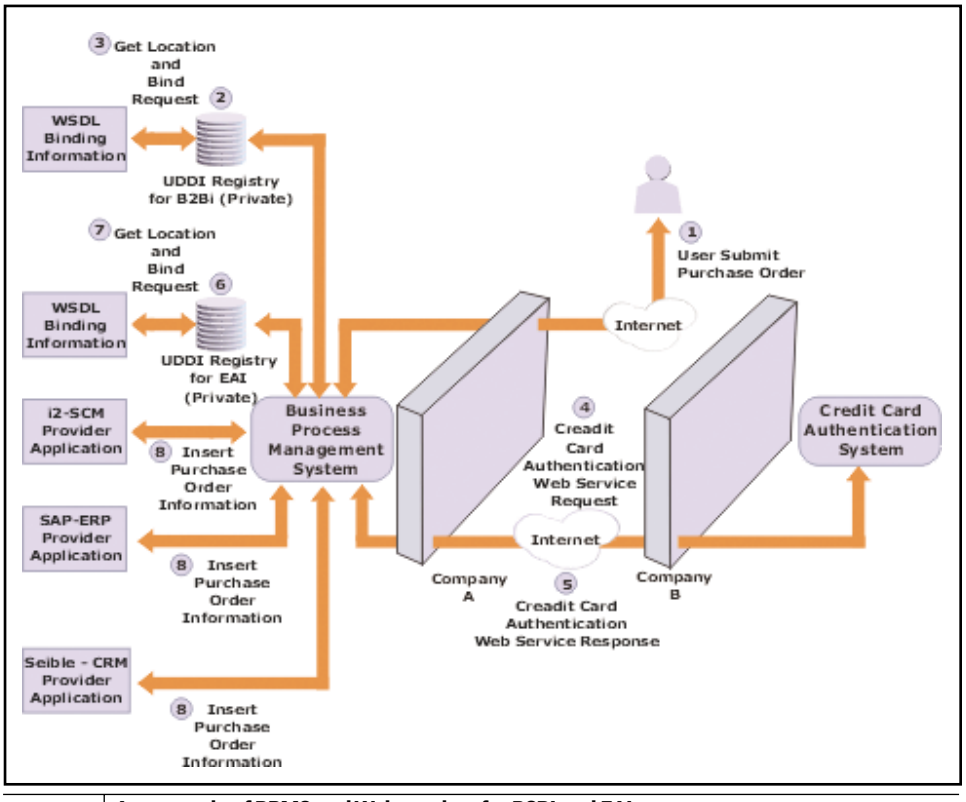


FIGURE 4 | **An example of BPMS and Web services for B2BI and EAI**

An Example of BPMS and Web Services

The retail and wholesale buyers of a supplier use an Internet portal to place orders of different goods. The portal is supported by a BPMS (either an integration broker or an application server, which provides support for business process management) that provides a business-centric environment for executing and managing business processes, such as order entry, as Web services.

The business process of order entry comprises using a third-party Web service for credit card authentication and using internal Web services to update the enterprise resource application (ERP), customer relationship management (CRM), and supply chain management (SCM) systems.

In this example, a BPMS manages a business process of order entry using Web services to loosely integrate third-party credit card authentication functionality, along with multiple internal applications.

Figure 4 shows the sequence of steps.

BPMS and Web Services for B2Bi

1. Through the procurement portal, the buyer submits a purchase order containing item number, quantity, shipping address, and credit card information.
2. Company A's BPMS gets information about Company B's Web service (credit card authentication) by doing a lookup in the private UDDI registry. This private registry is used to locate all external services, in this case, a Web service used for B2Bi.
3. The location of and WSDL binding information for the Web service is sent to Company A's BPMS.
4. Company A's BPMS invokes the Web service published by Company B to authenticate the buyer's credit card. The communication is based on SOAP over the Internet.
5. Company B's system receives the Web service request and sends the credit card validation response back to Company A. The communication is based on SOAP over the Internet.

It's worth mentioning here that this communication can be based on XML standards defined for the credit card validation business

THE LARGEST INTERNATIONAL

WEB SERVICES CONFERENCE & EXPO IN THE WORLD!

web services EDGE conference&expo

JUNE 24-27
JACOB JAVITS CONVENTION CENTER
NEW YORK, NY

OCTOBER 1-3
SAN JOSE CONVENTION CENTER
SAN JOSE, CA

WIN A $35,000 LUXURY CAR!

ATTENDEES WILL BE INVITED TO TAKE A GOLF SWING TO WIN AND RIDE OFF IN A $35,000 LUXURY CAR!

WEB SERVICES
SKILLS, STRATEGY, AND VISION

REGISTER ONLINE TODAY

FOR LOWEST CONFERENCE RATES
EARLY SELL-OUT GUARANTEED!

VISIT WWW.SYS-CON.COM

Focus on Web Services

Web Services, the next generation technology that will enable the Internet to work for you and your business, and finally provide that ROI you have been after, will be put under a microscope at Web Services Edge East 2002.

Information-packed sessions, exhibits, and tutorials will examine Web Services from every angle and will provide cutting-edge solutions and a glimpse at current and future implementations. Hear from the innovators and thought leaders in Web Services. Enjoy a highly interactive CTO Keynote panel that will debate and discuss the realities and promise of Web Services.

A Sampling of Web Services-Focused Sessions

- PRACTICAL EXPERIENCES WITH WEB SERVICES AND J2EE
- STATE OF THE WEB SERVICES INDUSTRY
- THE ODD COUPLE: MAKING .NET AND J2EE WORK TOGETHER
- EXPLORING THE .NET MY SERVICES INITIATIVE
- STANDARDS WATCH
- GUARDING THE CASTLE: SECURITY & WEB SERVICES

SEAN RHODY
CONFERENCE TECH CHAIR
WEB SERVICES TRACK CHAIR
EDITOR-IN-CHIEF
WEB SERVICES JOURNAL

Featuring...

- UNMATCHED KEYNOTES AND FACULTY
- THE LARGEST INDEPENDENT JAVA, WEB SERVICES, AND XML EXPOS
- AN UNPARALLELED OPPORTUNITY TO NETWORK WITH OVER 5,000 I-TECHNOLOGY PROFESSIONALS

Who Should Attend...

- DEVELOPERS, PROGRAMMERS, ENGINEERS
- I-TECHNOLOGY PROFESSIONALS
- SENIOR BUSINESS MANAGEMENT
- SENIOR IT/IS MANAGEMENT/C LEVEL EXECUTIVES
- ANALYSTS, CONSULTANTS

Hear these thought leaders in interactive, cutting-edge keynote addresses and panels...

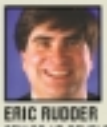
JEAN-FRANCOIS ABRAMATIC
SENIOR VP R&D, ILOG
FORMER CHAIRMAN, W3C

DAVE CHAPPELL
VP CHIEF TECHNOLOGY EVANGELIST
SONIC SOFTWARE

GREGG KIESSLING
CEO
SITRAKA

ANNE THOMAS MANES
CTO
SYSTINET

BARRY MORRIS
CEO
IONA



ERIC RUDDER
SENIOR VP DEVELOPER AND PLATFORM EVANGELISM
MICROSOFT

PATRICIA SEYBOLD
FOUNDER & CEO
SEYBOLD

For Exhibit Information

CONTACT: MICHAEL PESICK
135 CHESTNUT RIDGE RD.
MONTVALE, NJ 07645
201 802-3057
MICHAEL@SYS-CON.COM

SPONSORED BY:

IONA | END 2 ANYWHERE™
ADDS
bea
TogetherSoft
SilverStream
Actional
MERANT
Altaworks
PolarLake
ALTOVA

MEDIA SPONSORS

Federal Computer Week
Webservices.org
XML TIMES.com
Java Skyline
CF Advisor
WebServicesMall
XML.ORG
WIRE
jCert
wrox
BASIS
SD Times
JAVA
wireless
XML
WebLogic
WebServices
WebSphere
ColdFusion

OWNED AND PRODUCED BY

SYS-CON MEDIA
SYS-CON EVENTS

process for the vertical industry to which Company A belongs. For example, if Company A belongs to the electronic components industry, the request from Company A and the response from Company B can be based on RosettaNet's PIP. Other examples of B2B XML standards include ebXML and cXML.

BPMS and Web Services for EAI

6. If the response from Company B approves the credit card, as a task of the purchase order business process Company A's BPMS gets information about the ERP, CRM, and SCM packages Web services by doing a lookup in the private UDDI registry. This registry is used for internal services, in this case Web services used for EAI. However, if the response is negative, as defined in the workflow of the business process, the BPMS of Company A may throw an exception and report it to the buyer through the portal application.
7. The location of and the WSDL binding information for the Web services published by the ERP, CRM, and SCM systems is sent to the BPMS of Company A.
8. The integration broker invokes the Web service and enters the purchase order information in the ERP, CRM, and SCM packages.

Evolving into a Process-Based Organization

To evolve into a process-based organization, a company has to analyze and design its business process, followed by implementing, monitoring, and finally optimizing processes. Web services can be the enabling technology for all of these steps. A Web services-based organization would transition from a static, structured, and predictable model to a dynamic, creative, and knowledge-based model.

A small word of advice: companies should use Web services for private processes before they start venturing to use them for public processes. Public processes pose greater risks as far as security, transaction management, auditing, and persistence are concerned.

Directly Affecting the Bottom Line

Web services enable organizations to automate business processes, helping companies realize increased efficiency and cost savings through reduced manual processes. They are the building blocks of an enterprise's business operations and can be reorganized in any number of patterns to adjust to changing business requirements and to customize and optimize business objectives, eliminating the expensive process of rebuilding the systems from scratch.

Finally, since Web services run through industry-standard protocols, they enable organizations to eliminate the need for proprietary hardware, software, and network protocols.

Missed an issue?

We've got 'em all for you on CD!

JAVA DEVELOPER'S JOURNAL

The most complete library of exclusive *JDJ* articles on one CD!

Check out over 500 articles covering topics such as...

Java Fundamentals, Advanced Java, Object Orientation, Java Applets, AWT, Swing, Threads, JavaBeans, Java & Databases, Security, Client/Server, Java Servlets, Server Side, Enterprise Java, Java Native Interface, CORBA, Libraries, Embedded Java, XML, Wireless, IDEs, and much more!

JDJ The Complete Works
~~Reg $119.99~~
Buy Online
Only $71.99

WebServices JOURNAL / XML JOURNAL

The most complete library of exclusive *WSJ* & *XML-J* articles on one CD!

Check out over 400 articles covering topics such as...

XML in Transit, XML Industry Insider, UML, Integration, WSDL, VoiceXML, XML & E-Commerce, Beginning Web Services, Web Services Tips & Techniques, Ubiquitous Computing, Information Management, XML Script, UDDI, .NET, Objects & XML, XML Middleware, and much more!

Web Services Resource CD
~~Reg $119~~
Buy Online
Only $79

COLDFUSION Developer's Journal

The most complete library of exclusive *CFDJ* articles!

Check out over 250 articles covering topics such as...

Custom Tags, ColdFusion and Java, Finding a Web Host, Conference Reports, Server Stability, Site Performance, SYS-CON Radio, ColdFusion Tips and Techniques, Using XML and XSLT with ColdFusion, Fusebox, Building E-Business Apps, Application Frameworks, Error Handling, and more!

SPECIAL OFFER:
Buy CFDJ & CFA
The Complete Works
For Only $129.99

CFDJ The Complete Works
~~Reg $79.99~~
Buy Online
Only $71.99

CF Advisor

The most complete library of exclusive *CFA* articles!

Check out over 200 articles covering topics such as...

E-Commerce, Interviews, Custom Tags, Fusebox, Editorials, Databases, News, CF & Java, CFBasics, Reviews, Scalability, Enterprise CF, CF Applications, CF Tips & Techniques, Programming Techniques, Forms, Object-Oriented CF, WDDX, Upgrading CF, Programming Tips, Wireless, Verity, Source Code, and more!

CFA The Complete Works
~~Reg $79.99~~
Buy Online
Only $71.99

JDJStore.com

Order Online and Save 10% or More!

www.JDJSTORE.com

OFFER SUBJECT TO CHANGE WITHOUT NOTICE

COLLECT ALL 4
FOR ONLY $229.99
JDJ, WSJ & XML-J, CFDJ, & CFA

Web Services Journal
FOCUS ON BPM

written by Mark K. Ruport

Business Process Management: Tools for Eliminating Waste and Adding Value

Doing more with less

In this uncertain global economy, many companies have pared resources to the point that they have driven employees to a level of frustration that few of us deem tolerable. Corporate executives and stockholders consistently want more: more revenues, more profitability, and "more with less" from all departments. Yet, in trimming human resources to improve the bottom line, few companies examine their workflow to identify and eliminate extraneous processes or unnecessary roadblocks that would help maintain realistic objectives for employees, let alone enable an eight-hour workday.

Mark K. Ruport is the president and CEO of Optika. Mark has 20 years of experience in the computer software and services industry. He holds a BS in business administration and an MBA from Bowling Green State University.

In a world of e-business, many companies are plagued with outdated, manual business processes that could be streamlined by modeling and automation. Business Process Management (BPM) helps companies measure their core processes – analyzing which processes work and improving those that don't – to eliminate waste and add value. To be effective, this approach requires cross-functional, process-oriented management and a migration to electronic processes.

In order to "do more with less," organizations need to be able to process more transactions with fewer resources. Many factors – the implementation of a new Enterprise Resource Planning (ERP) or line-of-business system, increased sales, additional vendors and suppliers, more paperwork – can drive the need to either add resources or identify workflow inefficiencies. The following organizational "pains" force the need for BPM:

- An increase in the number of vendor or customer invoices
- Inability to integrate processes with the organization's ERP system
- Customer and trading partner dissatisfaction
- Errors and inaccurate postings
- Lack of filing and document storage space
- Lengthy cycle times, which can result in missed early-pay vendor discounts, increased days sales outstanding, and excessive exception handling
- High staff turnover due to frustration with manual systems
- Inability to deal with "out of tolerance" transactions

Enterprises that continue to hard-code all flow control, or insist on manual process steps and don't incorporate BPM's benefits, will lose out to competitors. To address a number of these pains, and to improve operations, organizations are implementing workflow or process automation software. During the implementation of workflow software, companies are encouraged to carefully describe and model their business processes, redefining steps and cutting the fat along the way.

Workflow software is more than just the electronic queuing and automation of the steps in a process. It often involves changing the process – how people do their jobs – from start to finish. Workflow software is an excellent enabler for changing business processes, primarily because the software is so flexible and easy to tweak, adapt, and overhaul. Workflow software is also great for identifying problems in a process. It can be a diagnostic tool that helps determine if your business processes are effective or not.

Traditional manual methods can be redefined to gain improved efficiencies by modeling the steps in the process and using innovative workflow products. Business modeling has led to new procedures in departments such as accounts payable, accounts receivable, customer service, human resources, loan applications, and claims/benefits processing. By implementing comprehensive workflow software, each of these areas has been able to leverage the value of their ERP systems, reduce labor resource requirements, decrease collection time and fees, and improve customer and partner satisfaction.

Additionally, imaging technology allows users – internal or remote – to capture, store and manage all of the documents and data associated with business transactions. Users need not be concerned about the format or source of documents and can access all of the information directly from their desktops.

One Working Example

In a strategic initiative to streamline its record keeping and to improve its customer service, SARGENT Manufacturing implemented imaging and workflow software to manage their extensive number of company records and documents. As a manufacturer of architectural hardware, SARGENT is required to maintain records of all key and security products and systems for the lifetime of the building for which they were produced.

Using imaging software, SARGENT now securely stores and can easily access documents associated with a customer's account. Customer service representatives can respond to customer inquiries promptly by accessing information electronically – information previously stored in file cabinets or microfilm. Using workflow software, SARGENT has completely automated the accounts payable approval process. SARGENT's processes overall are more efficient and both their vendors and customers have seen significant improvements in their service organizations.

Conclusion

Organizations will reap significant savings by defining and flowcharting current document management processes and addressing the challenges associated with the volume, storage, and control of filed information. After identifying what needs to be captured and who needs access to it, organizations will be in an excellent position to recommend an automated solution that offers a compelling return on investment in a short amount of time. Implementing the right processes and making a technology investment in imaging and workflow software can provide organizations with the competitive advantage necessary in today's economic environment.

Web Services Journal
FOCUS ON BPM

written by Mark Palmer

The Seven Principles of Web Services Business Process Management

Save time and money, but increase value

As Web services technology begins to impact business process management (BPM) and application-to-application integration, a question arises: "What business and technical challenges will we face applying Web services technology to BPM and application integration?" This article presents the seven dominant principles of good Web services design. The principles are based on 15 years of software technology evolution, combined with practical experience from today's deployed Web services applications.

Mark has over 14 years of experience in technology, most recently as CTO of YouthStream Media Networks where he led all technology initiatives, from internal operations to the creation of the Sodalis platform for integrating and supporting several hundred of YouthStream's partners, including leading colleges and universities in the United States.
MARK.PALMER@IONA.COM

The technical motivation for Web services is shown in Figure 1. In the '90s, the Internet enabled a network of browsers to access a flow of content. Today, the Internet enables a network of applications to access a flow of business transactions and processes. Web services standards promise to enable that shift.

The business imperative is also clear: yesterday's technological approaches to integration have failed – they are too expensive, too difficult to implement, and too vendor-specific.

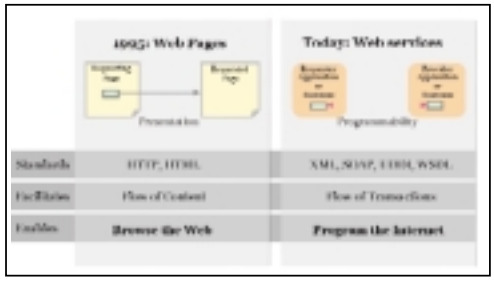


FIGURE 1 **Technical motivation for Web services BPM**

So let's address these technical and business challenges proactively by looking at a series of principles that can guide your enterprise toward a better approach to integration.

Principle 1: Plan for a Virtual Enterprise

Industry pundits bifurcate the integration space into internal (enterprise application integration, or EAI) and external (business-to-business, or B2B) application integration. In practice, this distinction is useless. The operations of corporations are becoming virtual – within an enterprise, functions like customer relationship management, sales force automation, and billing are outsourced to third parties. New insight into the operations of partners makes you more knowledgeable and efficient. The notion of a purely internal application – one that doesn't traverse firewalls or public networks – is disappearing.

The distinction between internal and external applications is hazy from a technology point of view as well. Both require process-flow management, packaged application adaptors, messaging, and security.

So, forget the pundits and think in terms of business problems, not market classifications.

Principle 2: Be Aware of the Hidden Integration Foil: Heterogeneity

The hidden foil to integration is heterogeneity, which rears its ugly head during

implementation. Why? In any integration, the applications involved typically:

- Have different representations of the same data
- Have different notions of similar business processes
- Are written in multiple programming languages
- Execute on more than one operating system
- Use more than one communication medium
- Were developed by teams with varying degrees of skill
- Use proprietary integration technologies

Individually, these problems present challenging technical hurdles; combine them, and you get prematurely aged project managers.

Some might say the answer lies in the emerging platforms for e-business: .NET and J2EE. Unfortunately, as we can see from the points of heterogeneity above, these environments don't come close to addressing our technology diversity. We don't really expect Microsoft software to run on IBM's OS/390 systems, Visual Basic developers to hack the Linux kernel, or COBOL applications to be rewritten in Java. We have no choice but to embrace our diversity.

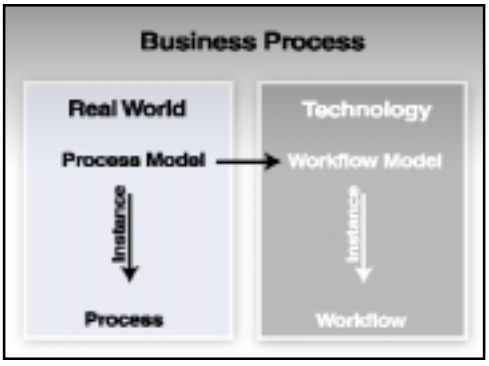


FIGURE 2 **Business process = process model + workflow model**

Simple awareness of the depth and breadth of heterogeneity issues is what Principle 2 is all about. The remaining principles work together to mitigate the heterogeneity problem.

Principle 3: Describe Business Processes Succinctly

When compared to most applications, which affect subsets of an enterprise or pieces of a supply chain, the impact of BPM is pervasive. It spans organizations within an enterprise, impacts scores of trading partners and consumers, and employs a vast array of technologies.

Figure 2 describes a way to think about business processes as a combination of real-world and technology models:

- The process model describes the steps that occur in the real world (e.g., the trucks that deliver goods from point A to point B, the schedules and locations of the drivers).
- Workflow models describe the technology interactions that support, interact with, or implement the real-world process model (e.g., system X sends request for purchase order to system Y).

More than any other application, the requirements for BPM should be expressed in plain English and reviewed by participants from each function involved. Process models should be described in the briefest of terms; they should be edited with the same scrutiny an editor applies to an important piece of writing. For example, do all participants in a business process agree that a feature request document should go to a customer support rep first, or should they go to a product manager? What are the exceptions? What are the expected time frames for each point in the process flow? Can steps be eliminated? Can steps be combined?

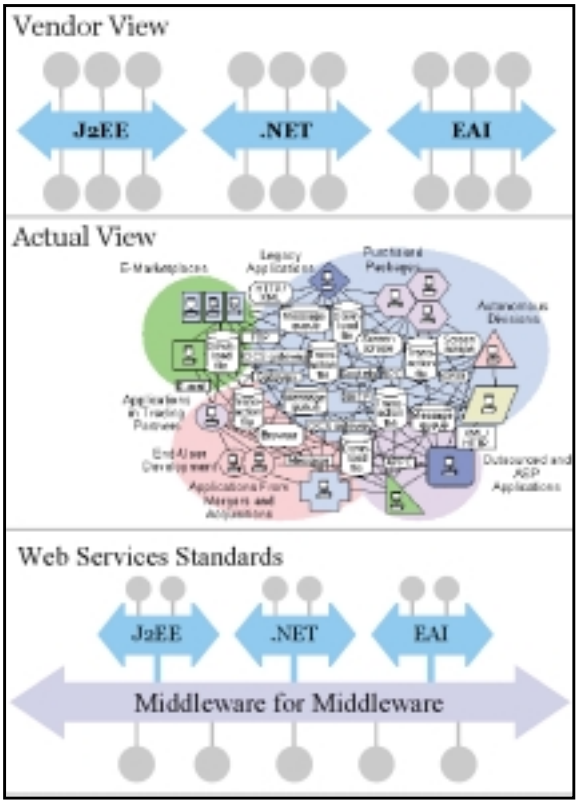


FIGURE 3 **Vendor view versus Actual view**

The workflow models themselves should also be expressed succinctly, although English may not be the best language for expression. Industry-standard languages and modeling tools are often better choices to express these models, but it's still critical for everyone to understand and agree that they're accurate.

Effective organizations follow Principle 3 and begin with clear, well-understood business process models.

Principle 4: Leverage Standards

The software industry's solution to the heterogeneity problem is middleware. Vendors sell middleware as a "software bus" that isolates applications from the complexity of a heterogeneous environment (see Figure 3, Vendor view). Simply develop a single interface to the middleware, and other applications can plug in to the bus, like an appliance plugs in to an electrical outlet for power.

Unfortunately, traditional middleware has failed to achieve that promise (see Figure 3,

Actual view). While EAI technology is useful for system-to-system integration, it requires homogeneous technology on each end of the connection and tends to create proprietary islands of integration. Proprietary tools and custom applications add more moving parts to the ugly reality of the enterprise technology infrastructure.

But the software industry has a history of neutralizing proprietary architectures once they become unwieldy with standards. From relational databases (SQL) to networks (TCP/IP) to operating systems (Linux), standardization – industry-driven or de facto – reduces our reliance on proprietary techniques and simplifies architecture.

Web services standards (see Figure 3, Web services standards) define "middleware for middleware." While proprietary products will not go away, proprietary interfaces between them will. Standardization cleans up the heterogeneity mess not by eliminating it, but by defining standard interfaces and services. Proprietary systems that support these standards can participate in the universal bus architecture that liberates software assets.

The case of Nordstrom.com demonstrates the business impact of Web services standards. Nordstrom's integration architecture is based on an XML/UDDI/SOAP Web services solution. The system integrates an IBM mainframe, HP minicomputers, Microsoft Windows, J2EE, .NET, OS/390, and Visual Basic. A simple request to check if Lancôme body cream is available requires a search that spans many systems. Each component in the system was modified to use Web services standards, which allows each application in the Nordstrom architecture to program to the standard interfaces supported by their tool of choice.

Despite the fact that Web services standards are still evolving, they provide real value today. Most popular tools support Web services already, so developers can use the tools they already know. Since the tools use the same standards, they require less proprietary training – think of the impact SQL had on the accessibility of databases. Standards are simpler to learn, which means developers of all levels can fully participate in the Web services party.

Principle 4 is about embracing standards early and incrementally – the sooner you incorporate Web services integration methodologies into your architecture, the sooner you'll liberate assets from proprietary tools. Furthermore, you'll begin to bridge the islands of integration that have formed and mitigate the heterogeneity challenge (Principle 2: Be aware of the hidden integration foil: heterogeneity).

Principle 5: Anticipate Change with Adaptive Infrastructure

When designing the interfaces for an application integration, think a year or two ahead. Envision a system that can withstand changes in the business, in the application, and in the underlying technology infrastructure. Systems that can adapt best to change are best suited for application integration.

Why is it important to develop adaptive architectures, from a business perspective?

1. The best way to optimize integration projects is to avoid them.
2. If your interfaces adapt, subsequent applications can be integrated more easily.
3. The longer your interfaces last, the better job you have done defining your requirements (Principle 3: Describe business processes succinctly).
4. Stable interfaces imply you have

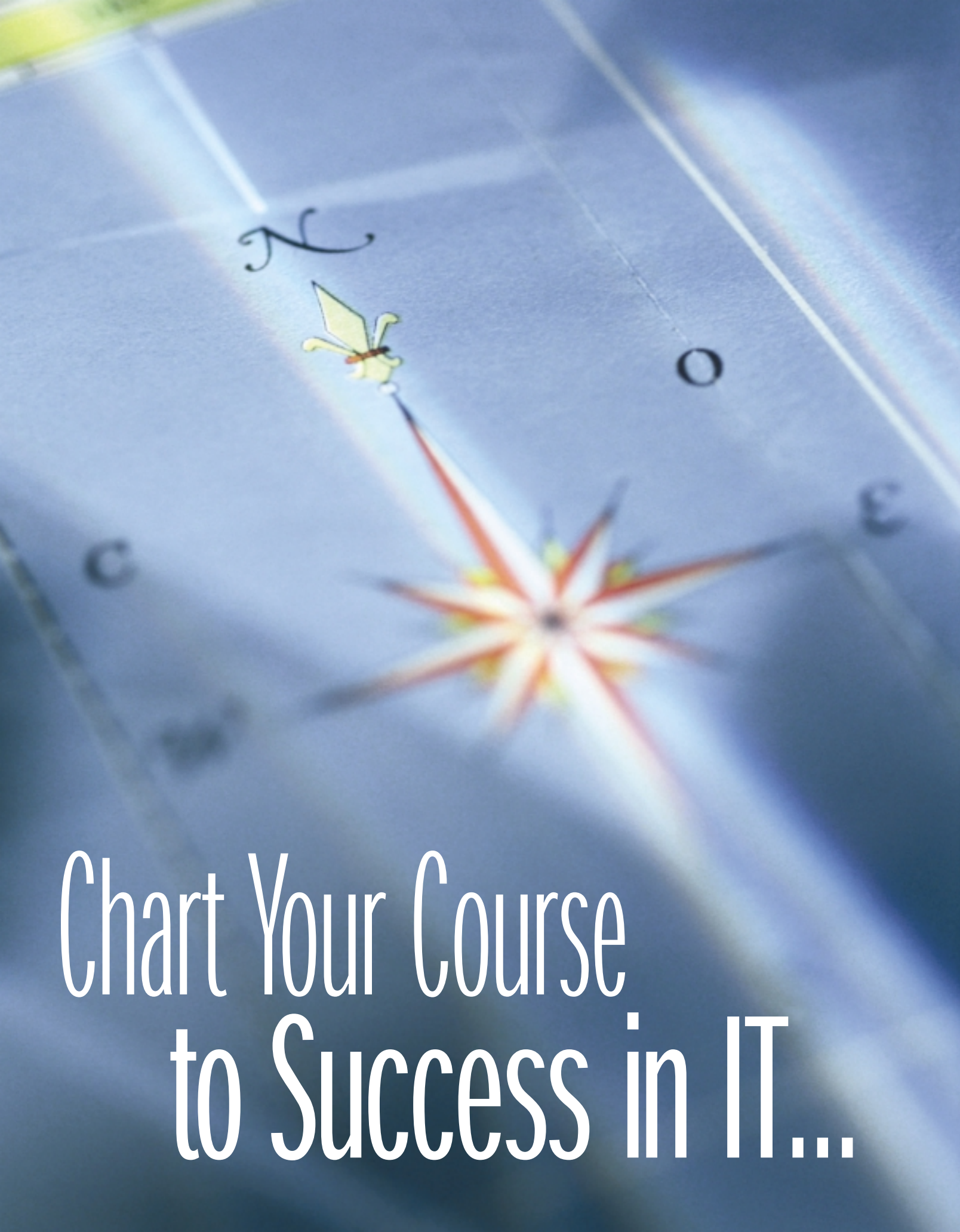


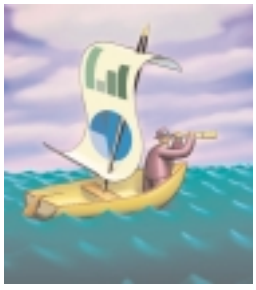
Chart Your Course to Success in IT...

...order your copy of

Java Trends: 2003

Available July 1*

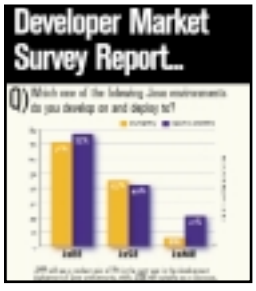
Don't go astray. In the vast sea of Internet technology, market conditions change constantly. Will Java remain the hot platform it is today? Will C# rapidly gain ground? What are the world's foremost Java developers aiming their sights toward? Which companies come to mind when planning their next project? How can their thinking direct your efforts?



EnginData Research has studied the IT industry extensively, spotting many key trends and alerting industry leaders along the way. In the first quarter of 2002, they embarked on their most challenging mission ever: the most in-depth study ever done of the Java development marketplace.

After months of analyzing and cross-referencing more than 10,500 comprehensive questionnaires, the results are now available.

EnginData's research is invaluable to leading IT vendors, integrators, Fortune 500 companies, trade-show organizers, publishers, and e-business organizations worldwide.



Here's just a sample of the critical data points the Java report delivers...

- ✔ Current and projected usage of languages and platforms
- ✔ Multiple rankings of hundreds of software vendors and tools
- ✔ Types of applications being developed
- ✔ Databases being used
- ✔ Purchasing patterns
- ✔ Company size
- ✔ Development and purchasing timelines
- ✔ Perceptions and usage of Web services
- ✔ Preferred Java IDE
- ✔ J2EE, J2ME, J2SE usage comparisons

As an IT specialist, **EnginData Research** focuses exclusively on three leading drivers in IT today – Java, Web services, and wireless development. Coming soon, our Web services survey will deliver such critical knowledge as:

- ✔ Time frame for developing Web services – enabled apps
- ✔ Percentage of apps with Web services today
- ✔ Sourcing and hosting Web services
- ✔ Perceived leaders in Web services tools and solutions

Navigate your company through the challenging IT marketplace with the trusted knowledge and intelligence of **EnginData Research**. Order your copy of the 2002–2003 Java market study by calling Margie Downs at 201-802-3082, or visiting our Web site.

engindata RESEARCH

www.engindata.com

*A limited number of preview copies will be available at our booth (#624) at JDJEdge International Java Developer's Conference & Expo, June 24–27, at the Jacob Javits Convention Center in New York.

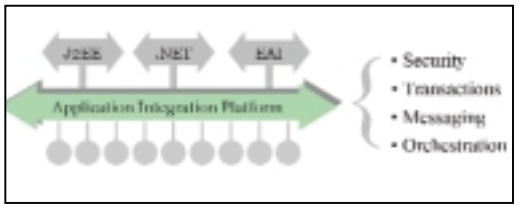


FIGURE 4 **An integration platform supports security, transactions, messaging, and orchestration**

accomplished a high level of granularity (Principle 7: Focus on Granularity, below), which is a natural use for BPM technology.

A recently deployed telecommunications application provides a good illustration of adaptive design. The application is an internal Operation Support System (OSS) that required integration with the Internet (via a J2EE application server) and a WAP interface. Phase one enabled Web-based access to the mainframe system using J2EE. Rather than designing an interface for the first application, the team considered the requirements of the next two integration targets as well. Phase two of the system would expose the same information to cellular customers via WAP. Phase three required the same integration, but with Microsoft's .NET. Since each system placed different demands on integration, the team decided to build an integration architecture that was self-describing (i.e., another application could query the system and learn about the interface programmatically) and extensible (i.e., via XML, new functionality could be added to the document-oriented architecture that would not break previous integrations).

While the requirements phase of the first project was increased by four weeks, the time to integrate the three systems was reduced because:

- Each development team utilized native standards–based interfaces in the tools they were already using – no training was required.
- Each project phase capitalized on Web services provided by systems on different software and hardware (heterogeneity was mitigated).
- Although new requirements were added for Phase 2, the Phase 1 system didn't break because the XML interface was extensible.
- Web services standards provided the architecture to support self-describing, extensible architectures.

By choosing techniques that allow interfaces to adapt over time, Principle 5 helps reduce costs, improves quality and ensures more efficient integration projects later.

Principle 6: Think Integration Platform, Not Integration Applications

An application is installed and used as a black box – without regard for its inner workings. A platform, by contrast, provides services to other applications so they can operate more effectively and can scale as more applications are added, load is placed on the system, and more or less resources are required.

Enterprises with hundreds of applications have unique platform needs. For example:

- The ability to distribute and balance the load across many instances of the same application
- The ability to manage many applications, each running in its own environment, as you would manage a single homogeneous system
- Security management, where each system has its own way of managing security (yet another heterogeneity dimension)
- The ability to allow small-scale integration projects to interoperate with these complex environments as well.

The general classes of platform services include security, transaction management, messaging, and the orchestration of business process flow, as depicted in Figure 4. Each class of service has emerging standards that apply to them, and a platform needs to take these issues into account.

Principle 6 orients you to think of the Web services "universal bus" as a platform that provides support for these classes of service.

Principle 7: Focus on Granularity: Think Faxes, Not Phone Calls

Good workflow design utilizes large-grain interfaces to exchange complex information. Let's say you want to buy running shoes. You could call a shoe salesperson and have a conversation, or send for a catalog and place an order. The differences in these two approaches illustrate the final principle of good business process management: proper selection of granularity.

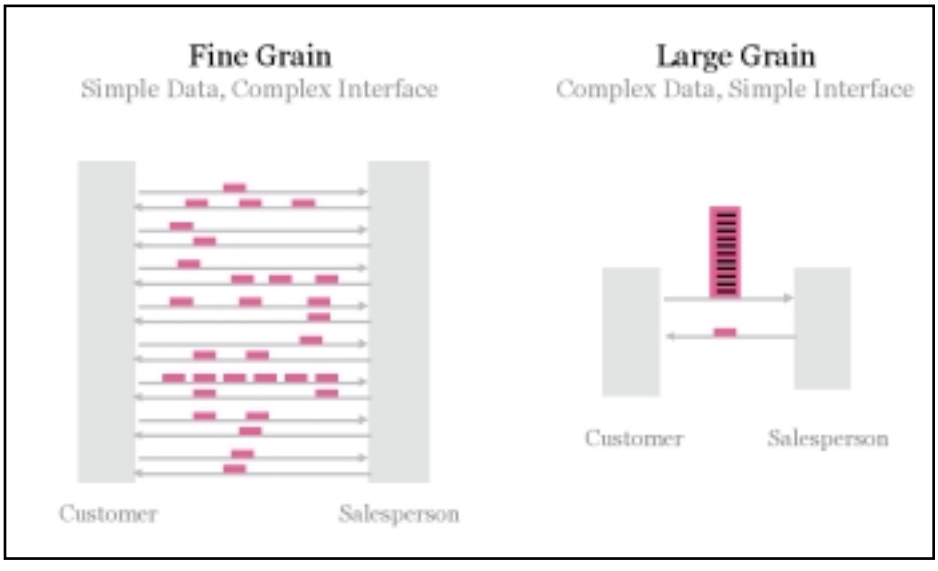


FIGURE 5 **A fine-grain and a large-grain conversation, or business process**

The fine-grain approach is like a phone call between two people. Consider the following conversation, or business process flow:

- "Do you have Saucony racing flats?"
- "No."
- "How about Nikes?"
- "Yes."
- "Do you have Nike Zoom Rival D?"
- "Yes, but only in sizes 10 and 9."
- "Great, I'm a 10; do you have three pairs in stock?"
- "No, just two."
- "OK, I'll take two pairs then."
- "That'll be $149.96."
- "What kind of credit cards do you accept?"
- ... and so on

This conversation can be seen conceptually in Figure 5. A few observations about this fine-grain conversation:

- ***Simple data is exchanged:*** "shoe type," "quantity," "size," and "price."
- ***Lots of data is exchanged, in small chunks.***
- ***The conversation is tightly coupled:*** Replies are meaningless without the context of the question. The phrase "sizes 10 and 9" tells you nothing useful on its own.
- ***You have to understand to whom you're talking:*** You have different conversations with a shoe salesperson than a waiter.
- ***Synchronous communication:*** Because the conversation is tightly coupled, you must be able to associate replies with requests.

Now for the large grain approach: I send a letter to a mail-order shoe company requesting their catalog. A few days later, you receive it by mail. Reading the catalogue tells what shoes are available, their prices, and sizes. You fill out an order form and fax it to the company with your credit card number. You assume if a shoe isn't in stock, they'll order more from a warehouse.

Things to note about the large-grain approach:

- ***Complex data is exchanged:*** structured, self-describing data is exchanged. You got an entire shoe catalog. You knew it was a catalog by reading it, not by knowing that it was a response to my request.
- ***Large chunks of data are exchanged infrequently.***
- ***Loose coupling:*** You didn't need to directly associate the catalog and your previous request because the catalog is sufficiently self-describing to anyone who knows how to read a catalog (understands the schema).
- ***Asynchronous communication:*** Enabled by the self-describing data and loose coupling. You could not know directly that the envelope containing the catalog was a response to your request.

This is the approach of large-grain systems, the ones best suited for effective business process management. It's also the approach that matches existing manual processes that are fax- or document-based. Somewhere in an enterprise, right now, a faxed order is being entered into a billing and shipping system. Proper choice of software interface granularity can leverage these existing document based systems.

It may seem obvious that large-grain system interfaces would tend to be more stable, easier to describe, and easier to integrate – so why would any developer develop fine-grained interfaces? First, specifying high-level, abstract interfaces is not easy. Developers tend to start expos-

ing systems by mapping existing, low-level interfaces directly to Web services, since it's the most straightforward approach. Another factor conspiring to lead us astray is today's tools, which tend to encourage this simplistic approach to creating Web services. For example, most Web services "wizards" read existing service descriptions, objects, or component models and simply map them one-to-one to fine-grain Web services.

With such a confluence of factors steering developers toward a less desirable approach, it's easy to see why integration projects tend to be expensive, late, and error-prone. However, by observing Principle 7, we can avoid these issues by proactively specifying course grain architectures that ensure more loosely coupled, highly cohesive systems.

Pulling It All Together

To pull the pieces together, let's quickly describe an effective Web services architecture for business process management:

- It addresses the needs of the virtual enterprise – internal applications as well as applications that run on external networks (Principle 1: Plan for a virtual enterprise).
- It's constructed with the full awareness of the heterogeneity challenge (Principle 2: Be aware of the hidden integration foil: heterogeneity).
- Business processes are well-documented and understood (Principle 3: Describe business processes succinctly).
- Standards adherence is central to the architecture; not an afterthought (Principle 4: Leverage standards).
- Change is anticipated; the solution is designed to be adaptive. Interfaces are designed to be stable and withstand inevitable change (Principle 5: Anticipate change with adaptive infrastructure).
- The solution employs a platform, not point-to-point applications or proprietary architectures (Principle 6: Think integration platform, not integration applications).
- Applications have large-grain interfaces. They implement loosely coupled, highly cohesive interfaces, rather than low-level, tightly coupled conversations (Principle 6: Focus on granularity: think faxes, not phone calls).

Keep these principles in mind as you implement integration architectures, and you'll save time and money and derive more value from the integration of your enterprise.

FREE TUTORIAL
JUNE 27TH WITH WEB SERVICES EDGE
2002 EAST REGISTRATION. LIMITED OFFER!

TO REGISTER: www.sys-con.com or Call 201 802-3069

web services world tour EDGE 2002

$195
REGISTRATION FOR SYS-CON SUBSCRIBERS
BEST EDUCATIONAL VALUE FOR THE HOTTEST TECHNOLOGY SKILLS!

Take Your Career to the Next Level!




SEATING IS LIMITED. REGISTER NOW FOR THE CITY NEAREST YOU! WWW.SYS-CON.COM

Learn How to Create, Test and Deploy Enterprise-Class Web Services Applications

TAUGHT BY THE INNOVATORS AND THOUGHT LEADERS IN WEB SERVICES

EXPERT PRACTITIONERS TAKING AN APPLIED APPROACH WILL PRESENT TOPICS INCLUDING BASIC TECHNOLOGIES SUCH AS SOAP, WSDL, UDDI AND XML, PLUS MORE ADVANCED ISSUES SUCH AS SECURITY, EXPOSING LEGACY SYSTEMS AND REMOTE REFERENCES.

RECEIVE 4 FREE 1-YEAR SUBSCRIPTIONS
$345 VALUE!

SYS-CON i-TECHNOLOGY EDUCATION

REGISTRATION FOR EACH CITY CLOSES THREE BUSINESS DAYS BEFORE EACH TUTORIAL DATE. DON'T DELAY. SEATING IS LIMITED.

NON-SUBSCRIBERS: REGISTER FOR $245 AND RECEIVE THREE FREE ONE-YEAR SUBSCRIPTIONS TO *WEB SERVICES JOURNAL*, *JAVA DEVELOPER'S JOURNAL*, AND *XML-JOURNAL*, PLUS YOUR CHOICE OF *BEA WEBLOGIC DEVELOPER'S JOURNAL* OR *WEBSPHERE DEVELOPER'S JOURNAL*, A $345 VALUE!

SHARPEN YOUR PROFESSIONAL SKILLS.

KEEP UP WITH THE TECHNOLOGY EVOLUTION!

"Presented an excellent overview of Web services. Great inside knowledge of both the new and old protocols. Great insight into the code piece."

– Rodrigo Frontecilla

"Very articulate on the Web services SOAP topic and well-prepared for many questions. I've learned a lot from this seminar and I appreciate this seminar for my job. Thank you!"

– Kenneth Unpingco, Southern Wine & Spirits of America

"I liked the overview of Web services and the use of specific tools to display ways to distribute Web services. Good for getting up to speed on the concepts."

– B. Ashton, Stopjetlag.com

Echoed over and over by Web Services Edge World Tour Attendees:

"Good balance of theory and demonstration."

"Excellent scope and depth for my background at this time. Use of examples was good."

"It was tailored toward my needs as a novice to SOAP Web services – and they explained everything."

WHO SHOULD ATTEND:

- **Architects**
- **Developers**
- **Programmers**
- **IS/IT Managers**
- **C-Level Executives**
- ***i*-Technology Professionals**

IF YOU MISSED THESE...

BOSTON, MA (Boston Marriott Newton) SOLD OUT!
WASHINGTON, DC (Tysons Corner Marriott) SOLD OUT!
NEW YORK, NY (Doubletree Guest Suites) SOLD OUT!
SAN FRANCISCO, CA (Marriott San Francisco) SOLD OUT! CLASSES ADDED

BE SURE NOT TO MISS THESE...

Each city will be sponsored by a leading Web services company

...COMING TO A CITY NEAR YOU

2002

NEW YORK AT WEBSERVICES EDGE 2002 EAST	JUNE 27
BOSTON	JULY 10
SAN FRANCISCO	AUGUST 6
SEATTLE	AUGUST 27
AUSTIN	SEPTEMBER 10
LOS ANGELES	SEPTEMBER 19
SAN JOSE AT WEBSERVICES EDGE 2002 WEST	OCTOBER 8
CHICAGO	OCTOBER 17
ATLANTA	OCTOBER 29
MINNEAPOLIS	NOVEMBER 7
NEW YORK	NOVEMBER 18
SAN FRANCISCO	DECEMBER 3

2003

CHARLOTTE	JANUARY 7
MIAMI	JANUARY 14
DALLAS	FEBRUARY 4
BALTIMORE	FEBRUARY 20
BOSTON	MARCH 11

REGISTER WITH A COLLEAGUE AND SAVE 15% OFF THE LOWEST REGISTRATION FEE.

TOPICS HAVE INCLUDED:
Developing SOAP Web Services
Architecting J2EE Web Services

On May 13th, the San Francisco tutorial drew a record 601 registrations.

TO REGISTER: **www.sys-con.com or Call 201 802-3069**

XML and Web Services Unleashed

An excellent introductory reference – but not for the beginner

Paperback: 1,200 pages SAMS; ISBN 0672323419; $49.99

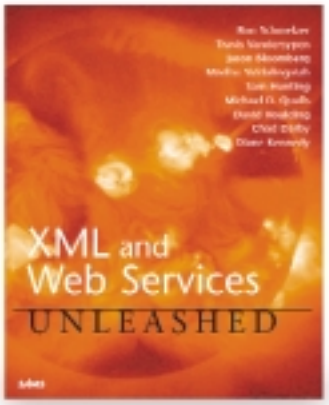
Sanjit Roy

Sanjit Roy is a senior consultant at Fusion Technologies, Inc., where he contributes to Web services and J2EE architecture and development. He has more than five years of experience in developing Web-based systems and has taught numerous courses on Web-based technologies.
ROYS@FUSIONTECH.COM

When I received my copy of this book, I was eager to read it after examining the table of contents. It covered almost every topic one could think of related to XML! I have quite a few books on XML and J2EE programming, but none covered as broad a spectrum of XML technologies as this one. However, because the authors tried to cover such a broad range, the book takes an introductory or "survey" approach to the subject matter. The book is well structured in spite of having nine authors. It's divided into four parts. The first part covers the fundamentals of XML and their validation using DTDs and schemas. These two topics are covered in about 100 pages. The authors provide many examples on schemas, which are assumed to be self-explanatory. A brief review of XPath, XPointer, and XLink is grouped into a separate chapter. Another chapter is dedicated to alternate schema representations using XDR, DSD, DCD, and Schematron. These are referred to as "dead" formats, but the chapter gives you an idea of the various other schema formats proposed, including the object-oriented SOX.

The second part (the best part!) focuses on building XML-based applications. XML parsing using DOM, JAXP, JDOM, and JAXB are covered in the same chapter. Some of the new features of DOM Level 2 specification are described briefly, including DOM traversal and range. XML parsing using SAX 2.0 is covered in the subsequent chapter. The sample code is written in Java using the Xerces parser. One interesting chapter focuses on XML transformation into XML/HTML using style sheets. Sample codes illustrate basic XSLT processing using MSXML Parser 3.0 and Apache Xalan API. The interesting part is the section on XSL Formatting Objects (XSL-FO), which lets you transform XML documents into PDF documents. The chapter on integrating XML with data is very generic, as different database vendors have proprietary XML implementations.

The book did not live up to its title for coverage of the various Web services topics, which are covered in three chapters. The first chapter introduces Web services and outlines its basic architecture. The description of the SOAP specification is very theoretical. The coverage of the basic SOAP syntax, envelopes, and header was adequate. However, the SOAP body element could have used more details. It would have been nice if the authors had written detailed examples of Web services and clients using both Microsoft and the Apache SOAP APIs. Also, the examples cover RPC-style Web services and miss out on the message-oriented Web services architecture. WSDL and UDDI are covered in a single chapter that mainly discusses the architectural issues, and there's brief coverage of the Microsoft SOAP Toolkit and the IBM Web Services Toolkit. The UDDI section covers the architecture and demonstrates the two competing vendor implementations – IBM's UDDI4J and Microsoft's UDDI SDK. There are two chapters dedicated to XML implementation models using the .NET framework using ADO.NET and Visual Studio .NET.



The third part of the book discusses various topics on implementing XML in e-business and B2B integration. These chapters expose the reader to many of the talked-about technologies, like the CommerceNet eCo Framework, XML/EDI, ebXML, and RosettaNet. Topics are covered briefly and their implementation models are discussed. There's limited coverage on delivering wireless and VoiceXML services in a separate chapter. The authors provide brief discussion of WML/VoiceXML concepts and their respective structures and elements. These topics are only introductory, and any reader developing these types of applications would probably need to refer to other materials.

The final part covers semantic Web topics. The main focus is on Resource Description Framework (RDF) and its specifications, working drafts and notes, data model, and schema.

Conclusion

Overall, all XML developers would like to have this book on their desks as an introductory reference to many XML-related technologies and topics. The book does not contain a CD-ROM; however, the source code for the examples can be downloaded from the publisher's Web site.

I must admit that this book isn't for the beginner. It assumes the reader has some fundamental knowledge of XML and Java programming and exposes the reader to the XML technologies that are currently available. The book does not delve very deep into any particular topic, but rather provides enough information to get the reader started.

This book is ideal for an experienced developer or architect who has started working with XML and wants to quickly cover the entire XML landscape.

Dot.com
- buyer's guide
- xml forums
- mailing list
- xml jobs
- xml store

Magazine
- advertise
- authors
- customer service
- editorial board
- subscribe

Content
- archives
- digital edition
- editorial
- features
- interviews
- product reviews
- source code

Conferences
- web services edge
  san jose, ca
  oct 22–25

search
Search

specials

best sellers
1. RoboHELP Office 2003 from eHelp
2. MultiActive: ec Builder Pro v6 for Win32 $204.99
3. Archos Portable 6GB MP3 Player / USB Hard Drive $248.99
4. Garmin: StreetPilot III GPS $699.99
5. V-Communications: Partition Commander V8.0 $39.99

What's Online

www.sys-con.com/webservices

Join the fourth wave of technology and become part of the newest paradigm in software development!

Web Services Edge East 2002

International Conference & Expo

Plan to join us at the Jacob K. Javits Convention Center in New York City, June 24-27, for the largest Web services conference and expo of the year.

Conference tracks will offer invaluable information on Web services, .NET, XML, and Java.

WSJ2.com

Check in daily for breaking news from the *WSJ* News Desk on Web services products, industry events, developments, and happenings. Be the first to know what's going on!

WSJ Industry Newsletter

Subscribe now for more in-depth Web services coverage. The most up-to-date coverage of companies producing products and technology that are at the forefront of this paradigm. Brought to you every month by industry leaders.

Join the Web Services Discussion Group

Web Services Journal proudly announces WSJList, the NEW Web services mailing list community at the center of discussions, technical questions, and more...

Join the WSJList now to monitor the pulse of the Web services industry!

Digital Edition

Don't have your print edition? Can't wait to read the next issue? Our digital edition is just what you need. As long as you have your computer with you, you can read *Web Services Journal* anytime, anywhere.

WebServices JOURNAL .NET J2EE XML

XML JOURNAL

THE FIRST & ONLY WEB SERVICES RESOURCE CD!

WEB SERVICES RESOURCE CD

THE SECRETS OF THE WEB SERVICES MASTERS

INCLUDES EXCLUSIVE .NET ARTICLES

MORE THAN 400 EXCLUSIVE WEB SERVICES & XML ARTICLES

EDITED BY SEAN RHODY

$119 CD VALUE FROM WEB SERVICES JOURNAL

web services EDGE conference & expo

$100 coupon Inside

EVERY ISSUE OF WSJ & XML-J EVER PUBLISHED

THE MOST COMPLETE LIBRARY OF EXCLUSIVE WSJ & XML-J ARTICLES ON ONE CD!

"The Secrets of the Web Services Masters"

CD is edited by well-known WSJ Editor-in-Chief Sean Rhody and organized into more than 40 chapters containing more than 400 exclusive WSJ & XML-J articles.

Easy-to-navigate HTML format!

Bonus:

Full .PDF versions of every WSJ & XML-J published since the first issue

- XML in Transit
- XML B2B
- Java & XML
- The XML Files
- XML & WML
- Voice XML
- SYS-CON Radio
- XML & XSLT
- XML & XSL
- XML & XHTML
- 2B or Not 2B
- XML Script
- XML Industry Insider
- <e-BizML>
- XML & Business
- XML Demystified
- XML & E-Commerce
- XML Middleware
- XML Modeling
- CORBA & XML
- Data Transition
- XML @ Work
- XML & Databases
- Electronic Data Interchange
- Ubiquitous Computing
- Information Management
- Objects & XML
- XML Pros & Cons
- Java Servlets
- XML Filter
- UML
- Integration
- WSDL
- Beginning Web Services
- Web Services Tips & Techniques
- Frameworks
- Management
- Security
- UDDI
- .NET

3 YEARS
25 ISSUES
400 ARTICLES
ONE CD

Special Limited Time Price

Now Shipping

$79 + S&H

ONLINE ORDER AT JDJSTORE.COM

SAVE $40

www.JDJSTORE.com

OFFER EXPIRES JUNE 30, 2002

WSJ News

Grand Central Communications and Lante Form Alliance

(San Francisco and Chicago) – Grand Central Communications, Inc., provider of a leading Web services network, and Lante, an information technology consulting company, have formed an alliance to jointly sell, market, and codevelop solutions that will enable enterprises to more easily and cost effectively integrate business processes with partners using Grand Central Communications' Web Services Network.

Grand Central and Lante will codevelop market offerings for the extended enterprise and will focus on solutions for industries such as financial services, energy, and insurance. The combined resources of the companies will facilitate the development of solutions that use new Web services standards to reduce the cost and complexity of integration, as well as provide incremental return on investment and ease of adoption among business partners.

www.grandcentral.com, www.lante.com

Macromedia Unveils New Integrated Product Family

(San Francisco) – Macromedia, Inc., has announced Macromedia MX, a new integrated family of client, tool, and server technologies for creating rich Internet applications that promise significantly more intuitive, responsive, and effective user experiences across platforms and devices.

Individually, the products in the Macromedia MX family are major releases that offer designers and developers powerful functionality for creating effective user experiences across the spectrum of Internet solutions from basic Web sites to complex Web applications. The Macromedia MX family includes the Macromedia ColdFusion MX server-scripting environment and a powerful suite of development tools: Macromedia Flash MX, Macromedia Dreamweaver MX, Macromedia Fireworks MX, and Macromedia FreeHand 10, which are available together in Macromedia Studio MX.

www.macromedia.com

NetBeans Implements Model-Driven Architecture

(Santa Clara, CA) – Sun Microsystems, Inc., has contributed a series of modules supporting Object Management Group's Model Driven Architecture (MDA) to the NetBeans open source project. This makes the NetBeans platform the first of its kind to support MDA, which protects organizations' software investments by capturing business logic – business processes and their appropriate interactions – in reusable models. It can significantly simplify the implementation of Web services, as the architecture insulates models from changes in the deployment infrastructure for the services.

Sun contributed Metadata Repository modules to NetBeans to make it easier for developers to support another programming language without extensive extra programming and to write NetBeans-based tools that interoperate with standards-based modeling tools. The Sun ONE Studio developer products (formerly Forte tools) are based on the NetBeans platform.

www.netbeans.org, www.sun.com

FULCRUM Now Makes the U.S. Scene

(Pune, India) – Software tools company AccelTree has debuted FULCRUM, an intelligent Java code builder, in the U.S. Neither an IDE nor a code generator, FULCRUM is a Java development tool that uses a proprietary concept of code templates

that can be used as building blocks to construct efficient Java objects and applications. The individualized templates, which serve as guides or examples of specific Java functions that can then be adapted to specific needs, can be overwritten and saved. FULCRUM also allows users to switch to manual coding at any time.

www.AccelTree.com

Anysoft Announces General Availability of Digital Cortex

(Newton, MA) – Anysoft, a software technology company that transforms applications into programmable objects, has announced the availability of Digital Cortex Sequencing System. Digital Cortex creates a windows environment in which applications become instantly accessible and externally programmable to transcend standards and programming languages. The result is a productive and universal methodology for integrating and enhancing applications, including the extension of applications to new delivery models, creation of software services from existing applications, and the provision of cross-application services.

www.anysoft.com

Software AG Launches XML Information Integration Tool

(Orlando, FL) – Software AG, Inc., the U.S. subsidiary of Software AG, Europe's largest systems software provider and a pioneer in XML technologies, has announced general availability of XML Mediator, a tool for building XML information exchange hubs. XML Mediator manages XML interactions that discover relevant information, a then uses that information to trigger behavior, such as routing documents or messages to an appropriate destination or transforming the content to other XML formats or presentation styles such as HTML, PDF, or WML.

XML Mediator helps companies integrate XML documents with reduced implementation time and reduced dependence on proprietary business integration tools.

www.softwareagusa.com

Rapid Adoption of Web Services a Certainty: Patricia Seybold Group

(Boston) – According to the Patricia Seybold Group, the Web services wave is an inevitable IT architectural evolution toward service-oriented architectures that will give business more flexibility in responding to and anticipating customers' constantly changing needs.

A new Patricia Seybold Group report, "The Web Services Freight Train," says Web services will have a more accelerated adoption rate than first predicted. Web services will be pervasive, infiltrating the majority of strategic applications. This is not to say that Web services will replace all other middleware, nor will existing applications be rewritten in Web services. But, as new, streamlined business processes emerge, they will require various functions in the application portfolio to be delivered on demand. These capabilities will be delivered by wrapping existing application functionality as Web services.

www.psgroup.com

SYS-CON'S WEB SERVICES EDGE

WORLD TOUR ATTENDANCE QUADRUPLES IN SAN FRANCISCO!

(San Francisco) – SYS-CON Events, Inc., presented the fourth stop of their Web Services Edge World Tour, the popular one-day tutorial series, to a sold-out audience of more than 650 attendees in San Francisco. The series was organized in response to developers' eagerness to plan ahead for SYS-CON's Web Services Edge 2002 International Conference & Expo East and West, which will take place June 24–27 in New York City and October 1–3 in San Jose, CA, (www.sys-con.com/WebServicesEdge2002East).

The first four stops of the Web Services Edge World Tour series focused on "Developing SOAP Web Services" and were sponsored exclusively by Systinet Corporation (www.sys-con.com/education).

The tutorial curriculum was designed to equip professional developers with all the tools and information they need to immediately begin creating, deploying, and using their own Web services.

Due to an overwhelming interest in the San Francisco tutorial, SYS-CON Events increased the originally planned single-session program to four sessions, two of which were presented by Oracle and focused on "Architecting J2EE Web Services."

SYS-CON Offers New Opportunities in San Francisco

SYS-CON Events also brought the attendees of the Web Services Edge World Tour together with leading

Web services vendors. The San Francisco attendees had a unique chance to ask questions and examine Web-services offerings from IONA, AltoWeb, Sonic Software, ObjectFocus, Systinet, and Oracle.

NEXT STOP:

Thursday June 21, 2002, New York City

The next stop on the education tour is June 27, 2002, in New York City. Stops in 12 additional cities will be announced before the end of this year, covering North America and several cities in Europe, Asia, and Australia.

www.sys-con.com

WSJ ADVERTISER INDEX

Web Services: Implications for Business Processes

A new trend – or a myth?

George Paolini

George Paolini is the chief marketing officer for Zaplet, Inc., (www.zaplet.com), a provider of collaborative business process management software. He previously led marketing and evangelism programs for Sun, helping to reposition the company as an Internet technology leader. He created the Java Community Process and the JavaOne Conference, and is credited with driving the widespread adoption of the Java platform, Jini, and XML.

GPAOLINI@ZAPLET.COM

I love Web services, for all the things they can't do.

It's not that I'm a pessimist, prone to look at the glass as half empty. Quite the contrary. Whenever I hear about Web services, I think about all the promises that won't be kept, and to me that looks like an opportunity to fill a void.

Web services will undoubtedly create an avalanche of new development opportunities. By standardizing interfaces and protocols, we can provide a common way for applications to communicate and data to be transmitted.

Solving application integration and data integrity eliminates two of the biggest headaches for CIOs today. (Security, policy administration, and others should naturally follow.)

But it's the next phase of Web services that I don't buy. And, unfortunately, this is the one where many people believe the hype. This is the notion that if we can break our applications into small enough chunks, and make them talk to one another, these components will magically come together to provide services that automate many tasks within our business processes.

You've probably seen the consumer-oriented demos that illustrate the point. You're on the road and your gas tank realizes it's running low, so it pings the GPS/cellphone combo to find the nearest gas station for your preferred brand. An enterprise version of this scenario might be around a business process. Perhaps it's a procurement system that constantly monitors a set of suppliers for the best price on a particular component and automatically places the order when its time to replenish the stock.

What's wrong with this picture? The problem is in the way we think about and model business processes. On paper or in theory, all business processes are perfect. Perfect, that is, until we get people involved.

In any business process, we have both transactions and interactions. We can automate transactions in black and white, yes/no binary format, but people interact in a world of fuzzy decisions, and work in that gray area of abstract concepts.

Worse yet, humans are unpredictable in behavior. We like to change things. We like to tinker with processes. We're not fond of talking to machines; we actually prefer to communicate in unstructured environments with other people, all of it disconnected from the world of transaction-based systems.

While our enterprise applications hum along automating transactions, we're holding discussions and debates, delegating tasks, and seeking approval in a parallel universe. Typically, this is done by e-mail, inarguably the de facto environment within the enterprise for communicating and collaborating.

So we design the perfect processes and build software applications to help automate the processes. Then we deploy the apps, and chaos ensues. This chaos manifests itself in what the industry euphemistically describes as an "exception."

There are two types of exceptions: those generated in a silicon-based world that doesn't understand the vagaries of human thought, and those generated in a human world that can't resist changing the rules. Let's look at each case.

Imagine a supply chain: an order is received on your docks from a supplier – only the count comes up short. Your receiving clerk will enter the count into an order management system and it will recognize the disparity. The best systems will even generate an e-mail alert.

What happens next? The recipients of that e-mail work to resolve the problem. Typically, they'll communicate and collaborate with their colleagues and superiors – by e-mail. When they reach a decision, they'll go back to the systems world and input the changes.

In the second scenario, imagine an application for automating pricing policies. Our star sales rep has a deal that's just too good to pass up. Problem is, it requires a 45% discount and the pricing policy is set at a 20% limit. No sense tinkering with the CRM system, it isn't programmed to accommodate a one-off change to the pricing policy. So our sales rep fires off an e-mail to the VP of sales, who makes an exception.

In both instances, we have parallel universes of systems and people. One works in a structured, binary mode; the other is more comfortable in the gray nuances of unstructured discussions, debates, and a little variety.

Which brings me back to Web services. Not only will Web services fall short of solving this problem, they'll actually exacerbate it. How?

All business processes require both transactions and interactions. The more transactions we "automate," the more opportunity we provide for exceptions that must be resolved in a world of human interaction.

So for those in the industry working to provide collaboration solutions for resolving those exceptions and bringing those business processes to a fruitful conclusion, this can be a very big opportunity indeed.

While the pundits say to developers: "Think Web services," the counterintuitive among us say, "Think about what Web services can't do."

Let's put it into perspective. Twenty years ago, computers were going to do away with the need for paper. Today, paper consumption is up by orders of magnitude, since computers not only failed to eliminate the need for paper, they actually increased the demand, as each of us mindlessly hits our "print" button. Hence, the big winners in the computer age so far are not the low-margin PC companies but the printer manufacturers.

Or how about this: 10 years ago, the Internet was going to "democratize information," threatening the very survival of traditional information-gathering operations, such as newsrooms and librarians. Ironically, the big winners turned out to be the professional news- and information-gathering organizations, who, with growing mounds of misinformation and (even worse) disinformation, have the critical skills to tell the true story.

And so, I'm predicting Web services will follow this trend, where the real winners will be those who clean up after the next big myth.