SYS-CON

TECHNOLOGY AND THE SARBANES-OXLEY RULES pg32

Volume 4 Issue 4

WebServices JOURNAL

.NET J2EE XML

MANAGING THE BOTTOM LINE

pg12

Service-level management is essential to deploying business-critical Web services systems

RETAILERS PLEASE DISPLAY
UNTIL MAY 31, 2004
$6.99US $7.99CAN

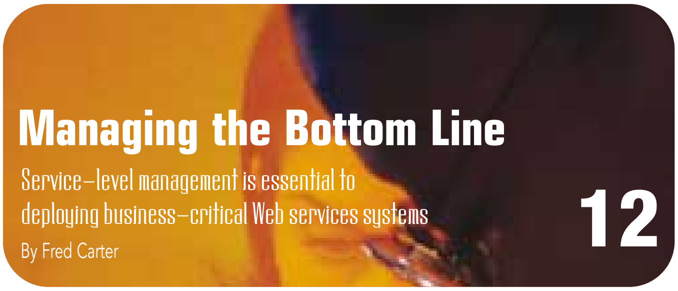
THE WORLD'S LEADING MAGAZINE DEDICATED TO WEB SERVICES TECHNOLOGIES • WSJ2.COM

April 2004 Volume 4 Issue 4

WebServices JOURNAL
.NET J2EE XML

Inside WSJ

WSJ: Features



Managing the Bottom Line
Service-level management is essential to deploying business-critical Web services systems
By Fred Carter 12

Why WSDL Is Not Yet Another Object IDL
maintaining a loosely coupled service architecture
By Savas Parastatidis & Jim Webber 20

Design Strategies for Web Services Versioning
Adapting to the needs of the business
By Chris Peltz & AnjaliAnagol-Subbarao 26

SOA Came to Boston at EDGE (East) 2004
Perspectives on the February 24–26 Conference & Expo 46

WSJ: Product Reviews

General Interface
An attractive alternative to the traditional Web tier
By Ahmed Sako 24

From the Editor

Comply or Die!
By Sean Rhody 7

Industry Commentary

Web Services as the Holy Grail?
By Jim Carty 11

WSJ: Management

Adopting Technology for Compliance
Risks and success strategies
By Chris Lesar 16

Sarbanes-Oxley & Web Services
Act now to get an early grasp of the inevitable
By Andy Astor 32

Web Services Management – Plotting a Course for Success
Slow adoption requires careful planning
By Robert Anderson 40

WSJ: Success Story

A Web Services Value Proposition
Lower cost, much higher benefits
By David Lady 36

WSJ: Integration

Web Services Converge with Data Integration
New ways to unlock the power of data
By James Markarian 42

Beyond Point to Point
WS-Eventing and Content-Based Routing for Integrating Multiple Applications via Web Services
By Adam Blum 50

APRIL
BUZZ

SOAPtest®

Verifying Algorithms Through Unit Testing

Complicated algorithms are typically contained within one class or function so that developers can easily manage them. How then, can you test the functionality of your algorithms and verify that they correctly solve problems before you have a full application written? The answer lies in unit testing.

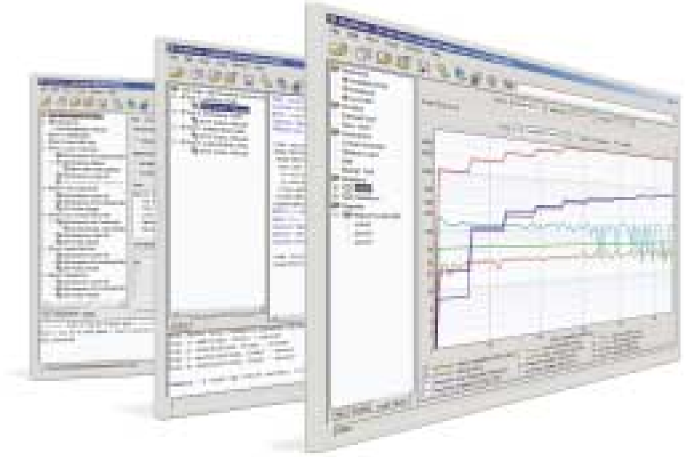
Unit testing is the perfect strategy for verifying and testing algorithms. Unit testing involves testing software code at its smallest functional point, which is typically a single class. Each individual class should be tested in isolation before it is tested with other units or as part of a module or application. By testing every unit individually, most of the errors that might be introduced into the code over the course of a project can be detected or prevented entirely.

I suggest that you investigate the benefits of performing unit testing to verify the functionality of your algorithms. Performing thorough unit testing reduces the amount of work you will need do at the application level, and drastically reduces the potential for errors.

By the way, unit testing is a fundamental part of Parasoft's Automated Error Prevention (AEP) Methodology. You can find more information about Automated Error Prevention at **www.parasoft.com**

*– **Adam Kolawa**, Ph.D.*
Chairman/CEO of Parasoft

It's automated. It's fast. And it's the most versatile Web Services testing tool ever.



Introducing Parasoft® SOAPtest®

The simple fact is, no other web service testing tool can do what SOAPtest can do. Or do it as fast and reliably. From ensuring functionality and interoperability to telling you how your web service is performing, SOAPtest automates all critical testing processes across the entire lifecycle of your web service.

But don't take our word for it.... Go to www.Parasoft.com/SOAPtest and try it for free – no commitment, no obligation. If you like it (and we suspect you will) just let us know. We'll be more than happy to help you and your development team get up and running.

For Downloads go to **www.parasoft.com/soaptest**

Call 888-305-0041 x1209 or email: buzz@parasoft.com

PARASOFT®
We make software work.™

FEATURES

- WSDL schema verification and compliance to standards
- Automatic test creation using WSDL and HTTP Traffic
- WS-I Conformance: Basic Profile 1.0
- WS-Security, SAML, Username Token, X.509, XML Encryption, and XML Signature support
- MIME/DIME Attachment support

BENEFITS

- Uniform test suites can be rolled over from unit testing to functional testing to load testing
- Prevent errors, pinpoint weaknesses, and stress test before deployment
- Ensure the reliability, quality, security and interoperability of your web service
- Verify data integrity and server/client functionality
- Identify server capabilities under stress and load

PROTOCOL SUPPORT

- HTTP 1.0
- HTTP 1.1 w/ Keep-Alive Connection
- HTTPS
- TCP/IP
- JMS

PLATFORMS

Windows 2000/XP
Linux
Solaris

Contact Info:

Parasoft Corporation
101 E. Huntington Dr., 2nd Flr.,
Monrovia, CA 91016
www.parasoft.com

www.altova.com
xmlspy
2004
NEW Microsoft® Visual Studio®.NET Integration
Enhanced Database Connectivity
XML Differencing
XPath 2.0 Analyzer
improved XML Schema/WSDL Editor
XSLT Debugger and Code Generator
XML
Schema
XSLT
WSDL
J2EE Station
CORBA
.Net Station
mapforce
Introducing mapforce™ 2004
New Visual Database-to-XML and XML-to-XML mapper!
Generate Java, C#, C++, and XSLT
mapforce 2004
FREE
30 day Trial

WebServices JOURNAL
.NET J2EE XML

INTERNATIONAL ADVISORY BOARD
Andrew Astor, David Chappell, Graham Glass, Tyson Hartman, Paul Lipton, Anne Thomas Manes, Norbert Mikula, Frank Moss, George Paolini, James Phillips, Simon Phipps, Mark Potts, Martin Wolf

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

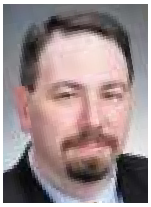
EDITORIAL
EDITOR-IN-CHIEF
Sean Rhody sean@sys-con.com
INDUSTRY EDITOR
Norbert Mikula norbert@sys-con.com
PRODUCT REVIEW EDITOR
Brian Barbash bbarbash@sys-con.com
.NET EDITOR
Dave Rader davidr@fusiontech.com
SECURITY EDITOR
Michael Mosher wsjsecurity@sys-con.com
RESEARCH EDITOR
Bahadir Karuv, Ph.D Bahadir@sys-con.com
TECHNICAL EDITORS
Andrew Astor aastor@webmethods.com
David Chappell chappell@sonicsoftware.com
Anne Thomas Manes anne@manes.net
Mike Sick msick@sys-con.com
EXECUTIVE EDITOR
Gail Schultz gail@sys-con.com
EDITOR
Nancy Valentine nancy@sys-con.com
ASSOCIATE EDITORS
Jamie Matusow jamie@sys-con.com
Jean Cassidy jean@sys-con.com
ASSISTANT EDITOR
Jennifer Van Winckel jennifer@sys-con.com

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

CONTRIBUTORS TO THIS ISSUE
Anjali Anagol-Subbarao, Robert Anderson, Andy Astor, Adam Blum, Fred Carter, Jim Carty, David Lady, Chris Lesar, James Markarian, Savas Parastatidis, Chris Peltz, Sean Rhody, Ahmed Sako, Jim Webber

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

Comply or Die!



WRITTEN BY
SEAN RHODY

At one point in my career I worked right across the street from Enron. I used to pass the big E on the sidewalk as I made my way to the account I was working on. Like most of us, the fall of Enron surprised and angered me.

One consequence of the Enron scandal and other accounting irregularities was the creation of the Sarbanes-Oxley regulations. This set of rules requires major corporations to ensure that their board is knowledgeable and aware of all of the ramifications of its financial position. It also makes the board members responsible and liable for financial proceedings within the company.

Accomplishing this task is no easy matter, especially for large corporations with multiple divisions and diverse management. Information that comprises corporate balance statements resides in hundreds of computer systems, none of which were designed with compliance in mind.

Which is where Web services comes to the rescue. The bad news of Sarbanes-Oxley is that the information required has to be very current – it can't take a year to put it together, it has to be timely. All of the old processes for generating corporate financial reports have to be sped up somehow. Web services promises to address this by removing the major cause of delay – impedance mismatch.

Accounting follows strict rules (well, okay, sometimes accountants play fast and loose with these rules, hence the Enron scandal, but the rules themselves are fairly straightforward). These rules allow accountants to produce standard reports (like a balance sheet) from myriad different sources. The problem that gives the accountants, and the systems guys, fits is trying to match up and correlate, federate, and aggregate data in different packages with different formats.

If this seems like a simple problem, think again. Even something as straightforward as a date can be problematic. Sybase, Oracle, DB2, and other databases all have different formats for dates. And SQL has been around over 30 years and has a standards body regulating it. Now imagine what it's like to try to get multiple accounting packages, ERP systems, management dashboards, and employee portals all pointed in the same direction and providing the same information. Most companies can't even come up with a single view of their customers. In some ways it's amazing that a balance sheet can be produced at all.

So the ability to remove many of these problems, assist in the mapping and correlations, and transform data from one format to another is a big aid in creating a timely flow of information. Of course, there's more to it than just compliance management. As a matter of course the improvements that occur as a result of compliance requirements flow not just to the board room, but to the bottom line. The ability to see deeper into the financial status of a corporation allows management to make realistic decisions and ideally helps to head off crises such as those that caused the accounting scandals.

But it's not just the board that benefits from system updates in the form of more timely financial information. The work necessary to implement better compliance management provides dividends in many systems areas. The ability to transform and begin to reduce the number of disparate views of things like customers and orders leads to shorter lead times and quicker fulfillment cycles – all benefits to the company and to the bottom line.

Web services won't solve all of the world's problems, whether we're talking about compliance management, order to cash cycles, or world peace (or alternatively, whirled peas). There's real work required to figure out what needs to be done with Web services capabilities, how best to expose services, and how to most efficiently combine them. Web services is a piece of the puzzle, but good design is still needed as well.

About the Author

Sean Rhody is the editor-in-chief of *Web Services Journal.* He is a respected industry expert and a consultant with a leading consulting services company.
■■■ Sean@sys-con.com

Mindreef News

Volume 2 April 2004 www.mindreef.com

INTRODUCING

Mindreef, WS-I, and Interoperability

WHITEPAPER

Q: What is Web Services interoperability, and why is it important?

A: Web services exist because there is a need for language and platform independent communication between computers. Ensuring interoperability between Web service components is essential to achieving this vision. The platform-independent specifications that make up the core Web services protocols (SOAP, WSDL, UDDI) are powerful, but expressive to the point of ambiguity. The same service can be represented with WSDL in numerous ways, and toolkits can encode the same message with SOAP in entirely different ways. Without commonly-followed guidelines that limit protocol use to an unambiguous subset, Web services would not interoperate.

Q: Who is the WS-I, and what is the Basic Profile?

A: The Web Services Interoperability Organization (WS-I) is an open industry organization that promotes interoperability among Web services across platforms, applications, and programming languages. Acting as a "standards integrator," the WS-I developed Basic Profile 1.0, a set of guidelines to limit the scope of what is acceptable Web service usage. The Basic Profile 1.0 describes how these core Web service specifications should be used to develop highly interoperable Web services. To make the Basic Profile 1.0 immediately useful, the WS-I recently released the WS-I Testing Tools, which enable Web services developers to generate a Basic Profile compliance report against any number of Web services artifacts.

Q: How is a Basic Profile compliance report used?

A: Generating a Basic Profile compliance report is excellent for pass/fail analysis. If you are developing a Web service, you can deploy a passing compliance report alongside your Web service to publicize its high quality, as well as support your interoperability claim. For clients, performing pass/fail analysis via a Basic Profile compliance report can aid in your choice of toolkit, and ensure that you are invoking Web Services with quality requests.

Q: Where does Mindreef fit in?

A: Mindreef understands the need for interoperable Web services and the Basic Profile. And while the WS-I testing tools may notify you of the existence of a problem, the report can be long and cumbersome, and specific errors give only a broad indication of where the problem might be. In an effort to help you discover and resolve your interoperability problems, Mindreef presents SOAPscope 3.0.

Q: What is SOAPscope and how does it go beyond the WS-I Testing Tools?

A: SOAPscope 3.0 collects messages and WSDLs in a variety of ways, and provides an intuitive framework for performing Web services diagnostics and analysis. SOAPscope is capable of analyzing Web service artifacts against a myriad of specifications and guidelines, including the WS-I Basic Profile, SOAP, and WSDL, as well as Mindreef Best Practices. SOAPscope provides an interactive UI to help solve problems, not just report them. As Web service artifacts are analyzed, SOAPscope pinpoints and highlights the XML fragments causing the problem, and presents it within a larger context, including extensive help, to assist you in solving the problem.

Q: How do I resolve errors as I discover them?

A: SOAPscope 3.0 addresses the resolution of Web services interoperability errors by providing a unified Web service diagnostic system, and supporting a process of iterative improvements. As artifacts are collected and automatically stored, they can later be examined, and analyzed for potential problems. If a problem is found with a SOAP message, you can edit the artifact, resend it, diff the result against previous results, and re-analyze the transaction. WSDLs can be viewed, dynamically invoked without writing code, diffed for changes, and re-analyzed as they evolve.

Q: Isn't analysis something that a tester does at the end of the project?

A: No. Of course you'll want to generate a compliance report at the end of your development to support your interoperability claim, but running analysis throughout the development process can quickly uncover latent bugs and issues at a time closest to when they were introduced. Uncovering those bugs automatically with intermittent analysis saves developers and testers from spending hours hunting down something that can be detected within seconds. SOAPscope 3.0 provides you with life-cycle value, performing analysis at any stage, and even generating the WS-I Basic Profile 1.0 compliance report when you are ready to deploy.

Q: Now that I've done all this, can I guarantee interoperability?

A: You can never guarantee interoperability, but by integrating analysis into your development and testing process, you can be assured that you've done everything possible to be interoperable.

Q: Where can I learn more?

A: Read more about Web services interoperability in our FREE whitepaper at http://www.mindreef.com/interop-wp.

> "SOAPscope rocks!"
> *- Don Box, XML Messaging Architect, Microsoft and co-inventor of SOAP*

Create software so brilliant
it can manage itself.

WebServices .NET J2EE XML JOURNAL

PRESIDENT AND CEO
Fuat Kircaali fuat@sys-con.com
VP, BUSINES S DEVELOPMENT
Grisha Davida grisha@sys-con.com
GROUP PUBLISHER
Jeremy Geelan jeremy@sys-con.com

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

SYS-CON EVENTS
PRESIDENT, SYS-CON EVENTS
Grisha Davida grisha@sys-con.com
CONFERENCE MANAGER
Lin Goetz lin@sys-con.com
NATIONAL SALES MANAGER
Sean Raman raman@sys-con.com

CUSTOMER RELATIONS/JDJ STORE
CIRCULATION SERVICE COORDINATORS
Shelia Dickerson shelia@sys-con.com
Edna Earle Russell edna@sys-con.com
Linda Lipton linda@sys-con.com

JDJ STORE MANAGER
Brunilda Staropoli bruni@sys-con.com

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

ACCOUNTING
FINANCIAL ANALYST
Joan LaRose joan@sys-con.com
ACCOUNTS RECEIVABLE
Charlotte Lopez charlotte@sys-con.com
ACCOUNTS PAYABLE
Betty White betty@sys-con.com

SUBSCRIPTIONS
SUBSCRIBE@SYS-CON.COM
1-888-303-5282
For subscriptions and requests for bulk orders,
please send your letters to Subscription Department
Cover Price: $6.99/issue
Domestic: $69.99/yr (12 issues)
Canada/Mexico: $89.99/yr
All other countries: $99.99/yr
(U.S. Banks or Money Orders)

Worldwide Newsstand Distribution
Curtis Circulation Company, New Milford, NJ
Newsstand Distribution Consultant:
Brian J. Gregory / Gregory Associates / W.R.D.S.
732 607-9941 - BJGAssociates@cs.com

For list rental information:
Kevin Collopy: 845 731-2684,
kevin.collopy@edithroman.com;
Frank Cipolla: 845 731-3832,
frank.cipolla@epostdirect.com

SYS-CON Publications, Inc., reserves the right to revise, republish and authorize its readers to use the articles submitted for publication.

SYS-CON MEDIA

Web Services as the Holy Grail?

WRITTEN BY
JIM CARTY

In order for companies to benefit from the strategic value of information technology, there needs to be much closer alignment between IT and the business side. So far, these two counterparts have had a difficult time seeing eye to eye. The problem is that despite earnest intentions by both to work together, the application integration tools that IT has had to work with have been cumbersome. This has resulted in too many projects being delivered late and over budget and has caused the business side to stop expecting IT to deliver what it needs when it needs it.

Integration is the bugaboo. Organizations are taking an enterprise-level look at their applications and this means integrating and sharing data between applications is a requirement in many projects. Gartner Dataquest predicts spending on integration projects will reach a staggering $10.6 billion in 2006.

So what companies need is a cheaper, easier, faster way to handle enterprise application integration. An *Infoworld* survey shows that application integration costs are at least 25% of the total IT budget at many companies. If Web services can help deliver on this requirement, they will play a role in improving business's perception of IT. As IT demonstrates greater responsiveness, the alignment with the business side will occur more easily. And IT is jumping on the bandwagon. The same survey indicated that 55% of the IT managers polled said Web services will make integration projects more viable.

The Never-Ending Quest

IT innovation is about making IT more accessible. By providing companies with the means to better leverage their IT assets Web services drives up the ROI for existing and new IT assets. Web services in and of themselves are not free and require their own cadre of specialists and consultants, but they are a more appealing option than older approaches. A recent survey of IT managers showed that 74% of them expect Web services to reduce their need for more expensive EAI software and services.

Relationship Building

Although IT departments are not known for their public relations savvy, they do realize the importance of having a good relationship with their internal customers. The problem has been one of difficult-to-implement technologies getting in the way of good intentions. The project delays and cost overruns posed by centralized, proprietary, EAI hub solutions hinder IT business-alignment efforts. Web services avoid many of the factors that make many EAI solutions so challenging by using standards that are widely supported. It's the business agility afforded by Web services that will bring the two sides closer and support the sense of a shared mission.

A necessary component in any business relationship is trust. In providing information solutions that accommodate managerial dynamics, management starts to see IT as a partner that "gets it." The value in promoting trust is that organizations will look at IT in new ways and try to integrate it into the fabric of the organization.

The World Is Not Flat

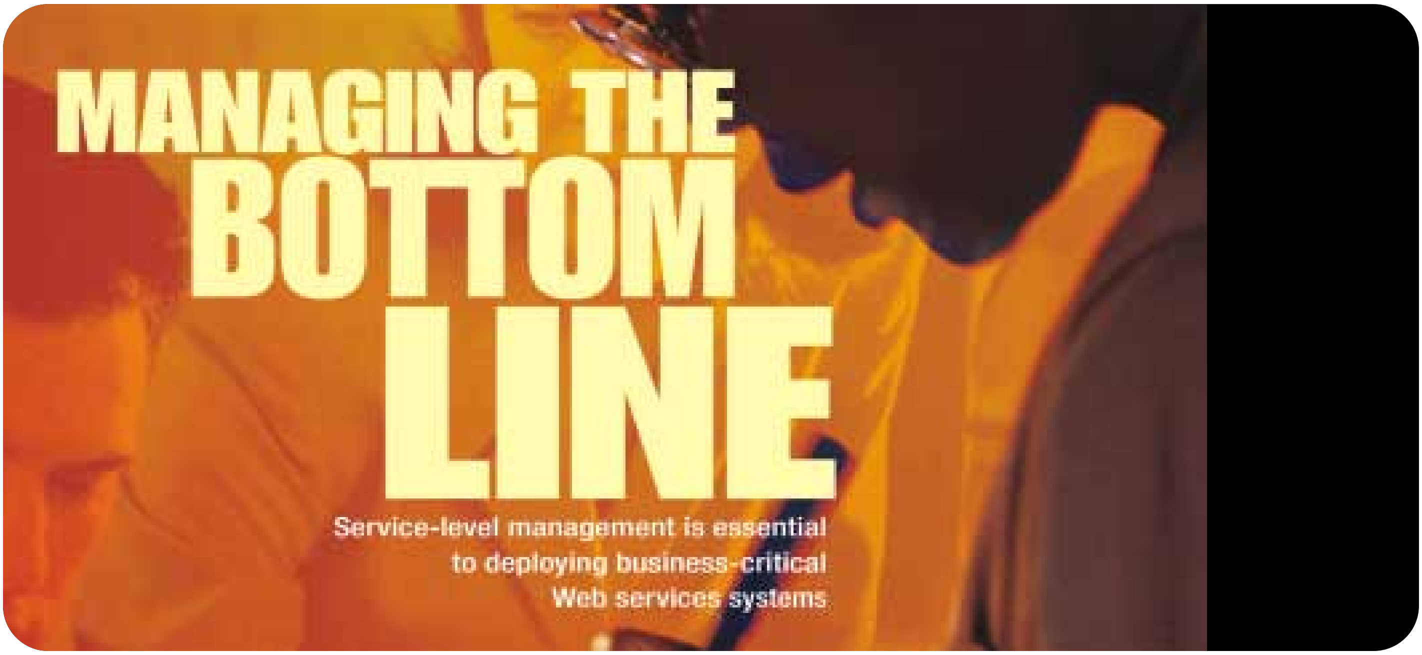
Increasingly, the most valuable applications are not internally focused but rather external facing. The competitive landscape brought on by the Internet requires companies to connect with competitors, supply chain partners, and marketing research databases. Web services help companies leverage their supply chain and other external connections. Until now, proprietary protocols have made the interchange of information and services difficult. Web services take advantage of the universal connectivity provided by the Internet to enable the sharing of applications in the form of services with any and all partners.

The Power of Agreement

The Internet may be the first thing all mankind has ever agreed upon. Web services has the potential to follow a similar track of standardization and global adoption. Of course it needs to avoid the pitfalls posed by vendors who have an interest in promoting "proprietary standards."

Standards, such as XML, are central to companies making more productive use of information. By investing in Web services, companies are able to leverage their IT capabilities into industry leadership. A well-known business practice is "giving to get." Web services enables a strategy of giving partners access to applica-

–Continued on page 18



MANAGING THE BOTTOM LINE

Service-level management is essential to deploying business-critical Web services systems

Web services are like your local auto repair shop. You don't want to do business with them until you have a clear idea of the level of service you can expect. Despite the many advantages of these new, standards-based systems, they will not become core business assets without capabilities for gauging and controlling their quality of service (QoS) attributes.

WRITTEN BY
FRED CARTER

Consider the example of a manufacturer that uses internal Web services for order fulfillment as well as extranet Web services to automatically replenish its inventory. To make informed decisions about order fulfillment, the manufacturer needs current and comprehensive information on the services and systems consumed by its inventory Web service. Is the warehouse management service running when inventory updates are needed? How long do suppliers' services take to respond with a promise date? How should service problems be addressed to minimize costs? If the manufacturer has Web services that are consumed by customers, it might want to prioritize orders by the customers' value to the business. Without the ability to see and control quality-of-service attributes across internal and external services, the manufacturer would be putting its business at significant risk.

This is where service-level management comes in. When implemented properly, service-level management, or SLM, provides a means of understanding the business impact service levels have on revenue and productivity, while also facilitating the diagnosis and mitigation of service problems within business processes. It's a continuous and closed-loop process of measuring, reporting, and improving the quality of service of systems and applications. Effective SLM includes a methodology for establishing and maintaining acceptable service levels to meet business objectives, streamline processes, and, of course, minimize costs.

SLM has typically focused on system management in terms of the physical or system-level behavior alone. With their many benefits, Web services offer the ability to expand that role.

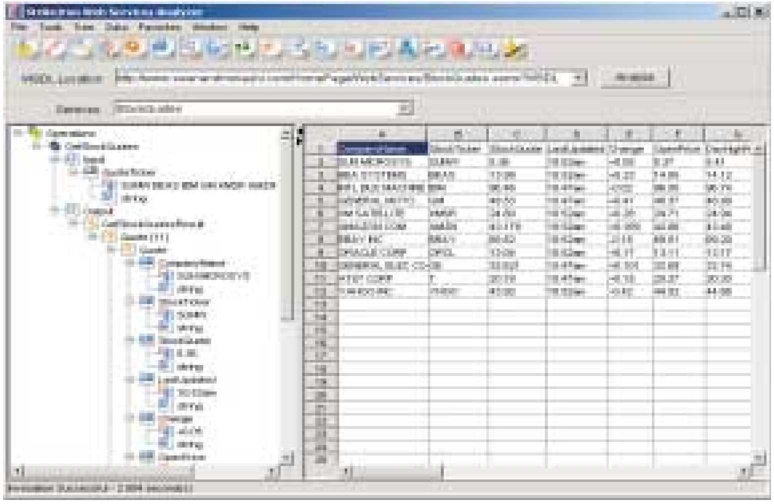
Traditional service-level management solutions focus on managing devices such as network routers, server equipment, and system software. Because these solutions are based on physical data and messages, they cannot be fully "aware" of Web services, to say nothing of logical- or application-level behavior. Moreover, they provide a limited perspective because they cannot see the business implications (such as the dollar value of orders) of the messages flying across the system. As a result, they are unable to proactively resolve quality-of-service problems or alter system behavior to meet underlying business objectives. (More on this point later.)

Abstracting the Management Layer

Whether you build your own SLM system or purchase one, you'll want to give some thought to the underlying mechanisms used to provide management capabilities. If you hard-code the logic into the Web services themselves you'll be challenged to maintain your management capabilities as your Web services evolve. Having your Web services and clients embed proprietary headers into SOAP requests and responses would limit your management capabilities to those Web services over which you have direct control. The best approach, therefore, is to abstract the management solution by using Web service intermediaries. With this approach, all the management policies reside in the intermediaries, enabling your Web services environment to grow naturally and without added development-time constraints.

A New Web Services Success Equation: Easier + Faster + Less Maintenance

By Bob Brauer
bob.brauer@strikeiron.com



The StrikeIron Web Services Analyzer for Java makes it easy to quickly discover, understand and utilize any Web service with a click of the mouse.

A new way to work with Web services

Now there really is a new and easier way to understand and utilize Web services that will accelerate your development and reduce your support time with users.

Advanced technologies, advanced solutions!

There is a new success equation for developers that brings together four advanced technologies for a new way to work with Web services:

- Graphic representations of the parsed code including data requirements, data structure and expected behavior.
- Dynamic invocation of the selected Web service requiring no additional software,
- Dynamic execution of the Web service regardless if it changes or is unknown at compile time, and the
- Ability to integrate this dynamic technology into applications.

Real benefits, real savings!

This translates into real benefits when working with Web services and provides opportunities for new types of applications:

- Reduce your time in half to decipher Web services.
- Quickly understand any Web service's behavior for faster utilization regardless of complexity.
- Significantly reduce your time and cost by eliminating the writing and compiling of additional code.
- Significantly reduce your support requirements through dynamic execution that is "self-adjusting". No need to re-write, re-compile, re-test and re-distribute.
- Spend more time creating applications and less time on maintenance and support!

The new success equation!

These technologies have been brought together by StrikeIron™ to provide an ideal set of companion products for every developer working with Web services. Now there really is a new success equation!

Point, click and use!

The StrikeIron Web Services Analyzer allows anyone to easily and quickly discover, understand and utilize any Web service with a click of the mouse. Everything you need is easily understood in a graphical interface with familiar tree and table views.

The StrikeIron Web Services Analyzer

- Easy-to-use point-and-click interface.
- Graphically visualize any Web service.
- Familiar tree and table views for input and output.
- Faster and easier discovery with a directory browser.
- Dynamic invocation of any Web service to test behavior.
- No time consuming programming or compiling.
- Save your output to Microsoft® Excel.
- No need for additional software to invoke or execute.

Dynamic applications: easier access, flexibility and lower support costs!

StrikeIron's Web Services Developer Kit accelerates application development and lowers costs by taking advantage of StrikeIron's Web Services Dynamic Object Technology (WSDOT). The WSDOT engine enables developers to *dynamically* load, invoke, aggregate and govern the execution of multiple Web services at runtime, without recompiling and without additional software. Reduce your costs and time to completion with a whole new level of freedom and flexibility!

The StrikeIron Web Services Developer Kit

- A full copy of the StrikeIron Web Services Analyzer and all its benefits.
- Access to StrikeIron's WSDOT API for dynamic invocation and execution without recompiling.
- Complete, standardized, executable Java code generation.
- An Eclipse plug-in for graphical analysis of Web services and code generation.
- A rich set of additional functionality for building applications including Web service monitoring, batch capabilities, multiple workspaces and more.
- No need for additional software to invoke or execute.

Delivering Web Services to the World!

StrikeIron is a pioneer of innovative products and subscription-based online services that provide a new level of ease of use and productivity for the utilization of Web services.

Free Trial! Please visit our Web site at www.strikeiron.com to download our newest products and to sign-up for our free newsletter.

StrikeIron, Inc.
919.405.7010
sales@strikeiron.com
www.strikeiron.com

Start Managing Service Levels Early

It's important to incorporate some service-level management into your Web services systems from the outset. Pilots and proofs of concept, which help to form your overall Web services strategy, call for careful evaluation and constant fine-tuning. With SLM, you and your team will benefit from a deeper understanding of system performance. What's more, you'll need SLM-generated data to quantify your successes in order to justify additional investments in Web services.

SLM can help you understand specific performance and availability issues of your Web service pilots. By recording and analyzing the service levels of your Web services experiments you'll have the knowledge and experience you need to ready your service-oriented architecture for prime time. For example, before exposing a new order-processing application via Web services interfaces, you'll first need to understand the impact the Web services will have on internal systems and the service levels you can expect to provide to users.

Rules of Thumb for Pilots

Keep in mind when planning and implementing SLM for Web service pilots:

- ***Be a discerning consumer of Web services:*** Service-level management will help you to distinguish third-party Web services by revealing QoS information such as performance and availability. You'll want to understand the QoS attributes of the Web services you consume over a period of time before incorporating them into business-critical applications.
- ***Differentiate the Web services you provide:*** You can actively manage your service levels to provide premium service to discrete user populations. For example, you might want to allocate your best-performing system resources to your highest-value customers. You'll want to practice such things with pilots before taking your Web services into production.
- ***Be aware of service quality:*** Pilots and proofs of concept are excellent opportunities to set informal quality-of-service goals for applications, and then track against those goals. For example, you can see if your application can provide 99% availability from Monday to Friday, 8 a.m. to 5 p.m. As you compile historic service-level data from your early experiences, you'll be better equipped to plan for growth and to respond to your user community.
- ***Create formal and informal service-level agreements:*** Service-level agreements, or SLAs, are the instruments for defining and enforcing SLM goals. An SLA sets the expectations of service providers and recipients alike. Formal SLAs could be contractual commitments that include noncompliance clauses, or they could be formal arrangements between internal departments that impact funds-exchange and charge-back policies. Informal SLAs represent a collection of internal service-level objectives based on internal goals designed to provide operations and IT departments with finer-grained control of their environments, beyond formal or legal commitments. An effective SLM solution will help you to ensure that your expectations are set on achievable goals, prevent service levels from becoming noncompliant with defined objectives, manage compliance failures according to the magnitude of the penalties, and minimize the cost of any noncompliance that occurs.
- ***Set objectives that match operational conditions:*** Each SLA has a set of service-level objectives, or SLOs, such as average response time or availability over a period of time. Take time into account when defining objectives. Factoring in hours of operation, peak hours of operation, scheduled downtimes, time zone adjustments, and so on, is key to ensuring the accuracy of your QoS data. This knowledge will also help you define trends and predict problems.
- ***Set objectives that match business conditions:*** As a service provider, you're likely to use the same service or application to support multiple clients. Because you'll probably commit to different service levels for those clients, you'll need to manage multiple SLOs for the same service. The more accurately your evaluation and mitigation strategies are matched to your business objectives, the more easily you'll be able to gauge and affect your service's impact on your bottom line. For example, any plan of action to mitigate problems should first address the most stringent SLAs. Even for informal SLAs that are tied to more generalized QoS goals, and not to specific customers, mitigation strategies should focus on business issues such prioritizing orders of higher dollar value.
- ***Continually evaluate service-level objectives:*** For greater understanding of your system's impact on your business, give consideration to the criteria for evaluating your objectives. For example, you might want to evaluate performance after the first 1,000 messages (to build a critical mass of data) and then every 30 seconds thereafter. Or you might want to track each order over $1,000,000. The more fine-grained the evaluation criteria, the more you'll learn from your experiments. Be sure to set warning thresholds in order to trigger alerts in advance of compliance failures.

To Unlock Business Insight, Content Is Key

XML provides standard syntax for representing information. This syntax makes it possible for an SLM solution to utilize the message content to play an active role in improving the system's quality of service. Tags, attributes, and element structure, as well as data from external sources such as LDAP, provide valuable contextual information that can augment use of the message content. By leveraging the content (e.g., order value) and context (e.g., customer priority) of the messages that are exchanged between Web services, the management solution can provide valuable business insight into service-based processes. Combining that business knowledge with system-level information, you can define service-level objectives and thresholds for any relevant set of business criteria.

Here are a few examples. By leveraging message content and context you can:

- Monitor the performance of an order-fulfillment process for all Platinum-level distributors
- Notify the account managers for all customers that are affected by a service failure
- Monitor response times by region
- Define service-level objectives by user roles, such as internal product managers, partner staff, high-value customers, etc.

Enforcing Service Quality According to Business Priorities

To optimize Web service systems and minimize risk, it's important to continually and proactively prioritize system usage according to business goals. An effective management solution not only measures, but actively enforces, end-to-end quality objectives that align with organizational goals. You can use the rich insight provided by service-based systems to optimize performance and availability in a manner that achieves the greatest business value. This ability becomes even more powerful with the opportunity to mitigate problems before they can impact business. For instance, if the performance of an order-fulfillment service starts degrading, you might prioritize customers with whom you have the most stringent SLAs. Or when a service is restored you might first process the orders of Platinum customers.

Managing Service Levels Across Complex, Distributed Processes

As you gain experience with Web services, you'll migrate to more mature environments, eventually integrating Web services that are distributed and federated across heterogeneous platforms. This brings forth business processes that span multiple divisions within the enterprise or cross corporate boundaries to include partners and customers. From end to end, the process will involve multiple, independently controlled Web services that might rely on other services and processes to complete a task. Typically, each Web service will interact with numerous other Web services, playing a provider role in some interactions and a consumer role in others.

Such complex architectures will also facilitate loosely coupled exchanges between multiple services. This will enable you to implement processes that can adjust to business conditions in real time by dynamically selecting the appropriate Web services. While this flexibility increases the effectiveness of the process, it also introduces a level of unpredictability into the system.

Consider the example of a company that would like to streamline its manufacturing and fulfillment processes across multiple factories and warehouses, as well as its partners' facilities. Depending on the available-to-promise inventory levels, the company will either fulfill the order from one of its warehouses or outsource the manufacturing to one of its established partners. This fulfillment process involves multiple Web services, both inside and outside the corporate enterprise, and must be very flexible. Managing such a process calls for:

- Gathering service-level statistics from distributed services, including inventory, manufacturing and warehouse services
- Monitoring services over which the manufacturer has no control
- Making real-time QoS decisions by monitoring each instance of the process (for instance, immediately notifying IT if the step that explodes the bill of materials goes down)
- Rapidly locating problems within a process and determining the impact on business
- Preemptively addressing issues before compliance failures occur

Combined, these capabilities provide the ability to successfully achieve business goals in a complex yet flexible system.

Rules of Thumb for Complex Systems

Some things to keep in mind about managing service levels across distributed, federated environments:

- ***Apply SLM to business processes:*** You'll need to manage service levels end-to-end across dynamic, multistep processes. Ideally, your service-level manager will be flexible enough to observe these processes without having to control them. This becomes especially important if an enterprise has multiple processes to track or there are numerous process engines within the organization.
- ***Determine the impact on business:*** By bridging system-level information (e.g., server availability) and application-level data (e.g., average orders per week), it's possible to get a clear understanding of the business impact service levels have on revenue and productivity. For example, if a service that calculates sales tax goes down, the IT group would want to know which business processes are affected, which customers are trying to access the service, and if there are any potential penalties for SLA noncompliance.
- ***Identify service problems early:*** IT and operations spend an inordinate amount of time trying to pinpoint bottlenecks in distributed, multistep processes. To ease this burden, they need a management solution that enables them to easily identify which operations or services are problematic. Better yet, given the opportunity to understand trends and predict failures with levels of confidence, they can avoid service problems altogether.
- ***Span multiple technical architectures:*** Your SLM solution must address the key architectural issues within a complex scenario of distributed and federated processes. It must aggregate measurements from distributed system components, including Web services hosted by other entities. It must scale as the logical network grows. It must be cross-platform to accommodate the heterogeneous nature of Web services and must collect data from a variety of sources (including WSDL-less Web services, non-SOAP XML, HTML, servlets, JSP, HTTP, and JMS). And it should leverage the system-level information that is collected by traditional management products such as IBM Tivoli or HP OpenView.
- ***Close the loop with historical information:*** Analyzing historical data provides added insight, such as an improved ability to identify bottlenecks, fine-tune service-level objectives, and anticipate changes in system performance. For example, by knowing the average order value during peak business hours you can better understand the revenue loss of a system failure. For more mature systems, this can also help identify new revenue opportunities – such as providing value-added services targeted at the most active users of a particular service.

Service-Level Management Is Taking Care of Business

In recent months, we've seen a great deal of momentum with Web services, as a growing number of companies are taking their new applications into production. What's more, we've seen Web service deployments from any variety of businesses – financial institutions, manufacturers, retailers, government agencies, telecommunications. The list goes on. One constant in all this change is that the companies deploying business-driving Web services have all implemented solutions that enable them to set service-level expectations, manage to those expectations, and resolve any issues that arise. To ensure the ongoing success and continual expansion of their Web service initiatives, the companies that are serious about Web services are using SLM to manage their bottom lines.

About the Author

Fred Carter is the chief architect for AmberPoint, Inc., a leading provider of Web services management software. Prior to AmberPoint, Fred was the architect and technical lead for EAI products at Forte, a role he continued at Sun Microsystems. Fred graduated from Northwestern University with a BS degree in computer science and received his MS in computer science from the University of California, Berkeley.

■ ■ ■ fcarter@amberpoint.com

Adopting Technology for Compliance

Risks and success strategies

■ With strict new regulations, such as Sarbanes-Oxley, that include serious penalties for those who do not comply, today's executives face more challenges than before. Lawyers, analysts, auditors, and corporate executives are confronting challenges they have not had to face in the normal span of their work in the wake of compliance regulations.

WRITTEN BY
CHRIS LESAR

According to AMR Research, Fortune 1000 companies on average will spend about $2.5 million just on Sarbanes-Oxley compliance tools in 2003. Technology tools that help speed the implementation and adherence to regulations can automate the process, reducing compliance-related headaches. Compliance with new federal regulations is not a one-time event and must be adopted with that in mind.

The Problem

In organizations today senior management needs to address three key areas of risk – controlled, inherent, and detection risk.

- ***Controlled risk:*** Can generally be classified into the environment, information systems, and procedures, all of which can be easily "controlled." They comprise attitudes, ability, awareness, and actions of management as they impact business operations
- ***Inherent risk:*** Cannot be controlled by the enterprise; the control environment created and supported by management can help ensure statements reflect economic realities. No structure can be 100% effective; therefore, some risk is normally associated in the normal course of business.
- ***Detection risk:*** The risk of not finding misstatements in reports.

The National Association of Corporate Directors (NACD) indicates that for corporate directors alone, time spent in areas such as this has gone up from 125 hours in 1999 to 250 hours in 2002 with an estimated increase of 20% in 2003. Technology must, and can, help solve this very important problem.

With looming Sarbanes-Oxley Section 404 deadlines and Section 409 requirements around the corner, time is of the essence. The application of technology is mandatory in order to provide the rapid return being sought.

Requirements

Today's standards require many controls, including a foundation for internal and external checks. Risk assessment by management of relevant risks and how they will be addressed is vital in the compliance process as is information and communication supporting all control components.

Technology is playing a key role in the compliance process. For example, the Internet is an ideal tool to monitor performance, risk, and compliance. Other common tools, like spreadsheets, are cheaper and easier to use but pose a control risk and reduce accuracy, agility, and transparency. Today's systems need to be streamlined to meet shorter deadlines and must manage documentation.

Senior executives should first determine how well their systems function and test the maturing of their internal controls by asking questions. Addressing the issue could be a function of process rather than technology. It is here that Web-based tools designed for simple access to large stores of data presented in user-friendly formats can play an essential role.

Simple areas where tools can aid in compliance are in enhancing internal controls. Today companies' recording and reporting systems remain messy. Ideally all of the reporting systems will be linked electronically to enable enterprise-wide drilldown to the smallest detail. For example, a compliance tool can help shorten the time between the end of the financial period and the government filing.

In today's business climate, where scandals have demanded companies focus on, and be able to demonstrate, good corporate governance, there is much more at stake than traditional cost and productivity issues. Rising D&O premiums (some estimate at 40% per annum), heightened personal liability for corporate directors and executives, potential reduction in access to capital, and lowered company valuations have broadened the need for greater flexibility in the technology required to support compliance initiatives. A simple initiative of providing corporate transparency has a tremendous ripple effect.

Content of various types must be easily aggregated, such as spreadsheets, Word documents, and other materials that might make up the notorious three-ring binder Board Book. External information, not just internal, to an organization, such as competitive and market intelligence, must be accessible without stepping through multiple screens and sign-ons, rendering the application unusable. Events from the ERP system of record that are deemed material must be pushed out to those needing notification. Overextended marketing budgets, excessive goodwill write-offs, out-of-control overhead expenses, and revenue items falling outside of normal guidelines are just some of the items drawing increased scrutiny from shareholders and regulatory agencies, not just the traditional internal company management. Streaming the detail and parsing via XML gives the appropriate level of user the optimal level of drill-down capabilities.

Everything OLD Is NEW Again!

The use of the same information, provided with appropriate security, can allow a corporate director to prove duty of care, a department head to maintain consistent levels of performance, and the knowledge worker the ability to adapt to changing conditions when appropriate.

Solutions

Fortunately, a reliable remedy is at hand. Business performance management practices and technology enable organizations to comply with today's stringent requirements. These practices and solutions can drive profitability through a planning, monitoring, and reporting cycle.

While automated compliance solutions are often installed in an organization, they are considered by IT departments as too difficult or are limited in their scope. In some cases, automated checking systems can be flexible and powerful enough to incorporate the entire compliance process.

In many cases, all of the data is available. In fact, most of the time there is an overabundance of data. The real challenge is in pulling it all together in a meaningful way by providing context and allowing powerful visualization and easy exploration.

Solutions evaluated need to be event- and data-driven and provide a near real-time effect. Solutions incorporating the current Web services standards in order to facilitate interoperability of disparate systems without the costly and time-consuming enterprise application integration (EAI) projects of the past will lead the way. Web services, SOAP, and XML are all technologies that need to be leveraged in order to provide just-in-time enterprise information integration (EII) at the point of use. This will facilitate timely decision-making and more rapid response to potential threats to the business.

The next step is to ensure that whatever solution organizations choose it supports occasionally connected functionality. This is a nontrivial exercise to support. The notion that an Internet connection is available everywhere just does not hold water. Being tethered to a server in an always-on mode goes against the notion of the highly mobile workforce.

A key element of the most appropriate solutions will be the ability to cut across departmental silos by pulling in information from multiple sources, very possibly external as well as internal to the corporation. The impact of these will provide tight linkage to the current trend of being business process focused. This is where it gets tricky. Do you use dashboards from the various ERP, business intelligence, and operational vendors you have in-house and try to stitch them together, or do you go for a common dashboard technology that you can connect to the myriad data sources you have in the organization and deliver a consistent look and feel across the enterprise?

The last major element of the solution must support extensive and easy end-user role-based personalization without constant programmer intervention.

Conclusion

With technology readily available to aid in the compliance process, organizations will now be able to compress reporting cycles, accelerate budgeting and planning processes, and gain the ability to capitalize on business opportunities as they arise. The potential is endless. ⓔ

About the Author

Chris Lesar is vice president of marketing and business development for Curl Corporation, based in Cambridge, Massachusetts. Curl is a developer of the industry's first rich client software suite to deliver Web-distributed enterprise applications.

■■■ clesar@curl.com

–Continued from page 11

tion capabilities in order to foster relationships that lead to more business.

An example of the business value of Web services is Amazon.com. Amazon.com is using Web services to make its system capabilities available to others who in turn use Amazon to sell products to their mutual benefit. It's give to get, and is an argument for companies to look at Web services not just as a tactical alternative to application integration but as a strategic initiative. In 2002, Amazon generated about 6.2% of their revenues, or $246 million, by using Web services to become an e-commerce platform.

Economic Principles

In addition to providing a significant ROI on their own, Web services boost the ROI of other IT investments by enabling greater reuse across the organization. Companies no longer need to incur the expense of duplicating functionality that exists elsewhere. And most important, you're speaking a language that the CEO and CFO understand and appreciate. Web services moves IT out of the expense category and into a long-term value proposition.

Is It Just the Latest Dance Craze ?

Every IT investment decision comes with the question of how long before the technology becomes obsolete. To the chagrin of CFOs, too many technologies become outdated before they're fully depreciated. A natural question is whether Web services are a passing fad or the building blocks for future growth and value. Web services are not the "end all" for a company's IT challenges, but they are an important next step for companies to take. They let companies leverage existing standards such as HTTP and the Internet, and lay the groundwork for taking advantage of future standards-based advances. ⓔ

About the Author

Jim Carty has been in the IT-related industry for 22 years in a variety of roles including development, marketing, management, and operations. Most recently, he has taught New Web Technologies at Columbia University and been the CIO of one of PNC Bank's divisions. See www.isvaluecorp.com for more information.

■■■ jim.carty@isvaluecorp.com

Lowering the cost and complexity of developing secure, manageable enterprise-class Web services with BEA WebLogic Workshop™ and Confluent for BEA

With Confluent for BEA, Web services developed in BEA WebLogic Workshop™ 8.1 are automatically monitored, managed and secured at development time. Confluent's Control is bundled with WebLogic Workshop 8.1.

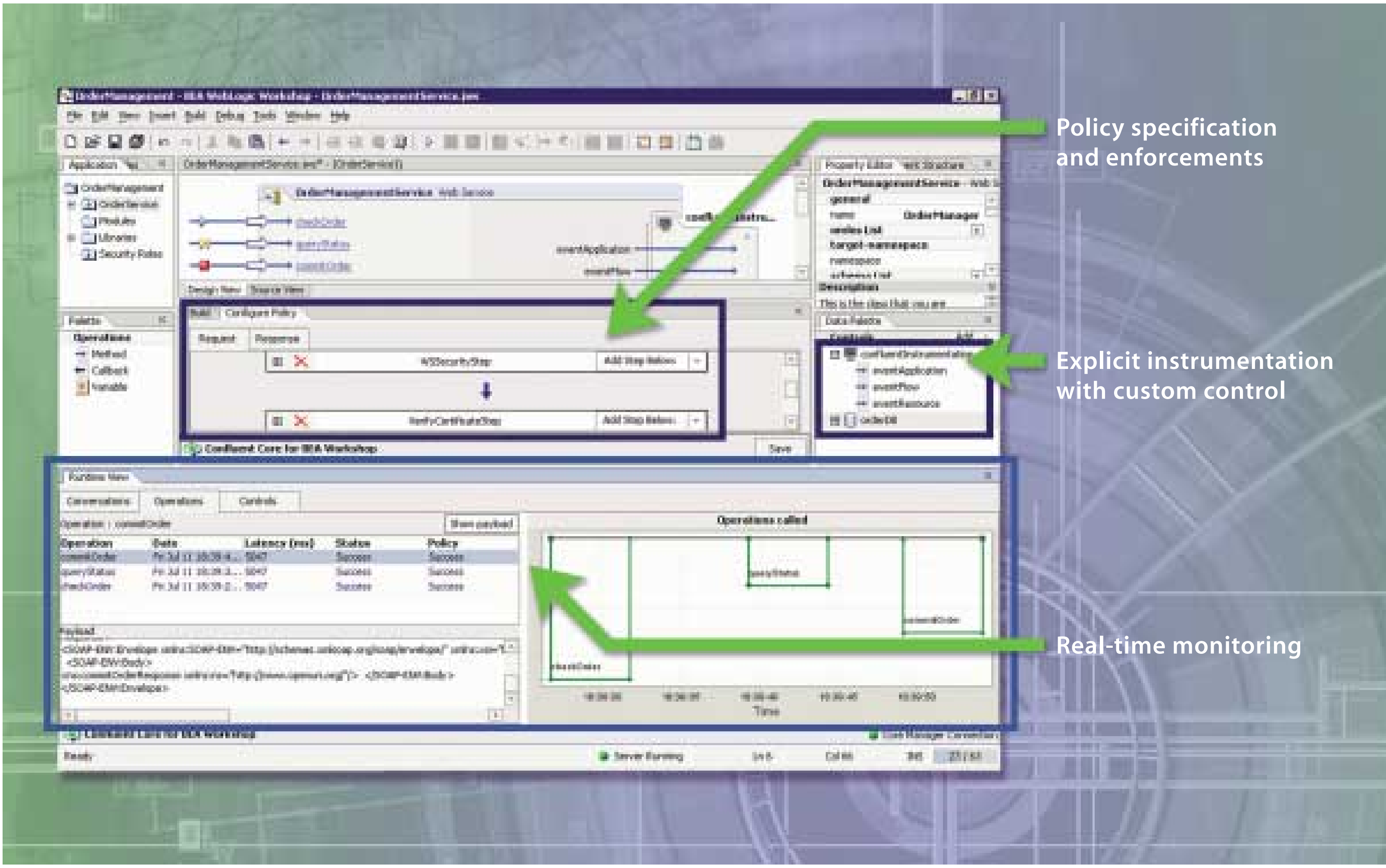
- **Real-time monitoring**
  No additional coding required, turn on instrumentation for a Web service, view all conversations as they execute, and drill down to views of individual operations and controls
- **Policy specification and enforcement**
  Define operational policies including security, logging and monitoring for Web services within BEA WebLogic Workshop 8.1
- **Explicit instrumentation with custom controls**
  Use Confluent Control to explicitly call operations such as monitoring and security from Web services

BENEFITS:

If you're a Developer—focus on application logic not infrastructure services

If you're an Architect—consistently implement security, change and other operational policies

If you're an Operations Manager—efficiently monitor, manage and evolve increasingly distributed applications



Confluent Software

Download Confluent for BEA at:
http://www.confluentsoftware.com/solutions/download/beacib.html

For more information on additional Confluent Web services management solutions, visit: **http://www.confluentsoftware.com**

Why WSDL Is Not Yet Another OBJECT IDL

Maintaining a loosely coupled service architecture

There has been much debate lately on what exactly WSDL's purpose is, and much of that debate has focused on whether WSDL is an interface definition language (IDL), or whether WSDL is better used to specify message-level contracts (without any associated operational semantics).

WRITTEN BY
SAVAS PARASTATIDIS &
JIM WEBBER

In this article we present an argument that dealing with WSDL as a message-level contract description language is the right way to go for building loosely coupled Web services.

Interfaces and Contracts

Before we delve into the specifics of how WSDL should be used, we need to understand the difference between a classic interface and a contract. By understanding the different problems that each solve we can begin to form a picture of when it is appropriate to use one or the other.

Let's start by briefly recalling exactly what is meant by an interface.

In an object-oriented view of the world, applications are composed from objects – entities that encapsulate state and have a well-defined, publicly visible interface. The interface describes the operations or methods that an object supports and that other objects can invoke. Each method represents an action that, when invoked, may or may not result in the state of the object being changed.

In this object-oriented world, we assume that a shared understanding of how the interfaces of the objects are described exists, that there is a common language for defining them; that's the *Interface Definition Language*, or IDL. An IDL provides us with a collection of primitive abstractions that we can use to define objects in terms of their interfaces but also with constructs that enable us to create new abstractions (sometimes called "classes" or "interfaces"). The collection of abstractions makes the *type system*, which is shared throughout our entire object-oriented application. In fact, the type system characterizes the application since more than one type system cannot coexist (e.g., the type system of CORBA, DCOM, Java RMI, etc.).

An abstraction defines the in-memory representation of an object. The interface of an object is the entry point to this in-memory representation. When building object-oriented systems, it is easier to reason about objects in terms of their interfaces since we only need to know how to access the objects and not their actual layout of the in-memory representation. Nevertheless, there is always a conceptual association between an interface and an in-memory representation of an object.

In a service-oriented view of the world functionality is exposed via services that are entities that can send and receive messages. Applications are thus composed by combining services that interact through message exchanges. To enable services to be interoperable, the format of those messages and the message exchange patterns that make the service-to-service interactions have to be agreed on in advance through a contract; thus we need a language with which to express such contracts – a *Contract Definition Language*, or CDL, as opposed to an IDL.

Unlike an interface, a contract is about the information on the wire and not about how the in-memory representations of objects are accessed. (*Note*: Throughout this article, we leave the details about the concrete binding of WSDL messages aside and assume that the XML infoset described by the abstract messages is honored by any underlying bindings.) Contracts allow systems to be built from components (services) that have well-defined boundaries (see "References" section) . These services can be developed and evolve inde-

pendently since they do not share any abstractions. No single type system spans all the components. All that is agreed upon is the information that is exchanged and not how to access any in-memory representation of the objects.

In the Web services arena, a WSDL document is the contract that a service advertises to its potential consumers. Since it is written in XML and (potentially) uses XML Schema, there is no need for shared abstractions or interfaces of in-memory representations.

The big win for this approach is that in any given system we must decide only upon what information needs to go onto the wire. This is a considerably simpler logistical problem to solve than ensuring that all components of a distributed application are able to share a common type system and that all developers (who may well be working for different organizations) understand the operational semantics of the various types in that system. In short, this makes developing distributed applications, particularly those that span organizations, easier and thus cheaper and more robust.

Web Services Description Language

In WSDL 2.0, the wsdl:portType element has been renamed to wsdl:interface (this is true at the time of this writing as per the November 2003 draft of the WSDL specification). Although its name is suggestive, the wsdl:interface element should not be treated the same as a Java interface/class, a C++ class, or a C# interface/class. As we argued earlier, a WSDL document is a contract that describes message exchanges and not a description of an interface for an object.

While it is usually a mistake to do so, consumers of a WSDL document may choose to use it as if it were an IDL document. In fact, many of the current Web services development tools do just that by converting WSDL documents to classes with methods and presenting an object-oriented view of the world to developers.

The wsdl:interface element should be seen as a group of message exchanges in which a Web service is prepared to participate. It defines the set of interaction patterns that are part of the contract a Web service advertises and to which it adheres. The WSDL working group might well have chosen to name this element wsdl:interactions to better represent its purpose.

The wsdl:interface element may contain any number of wsdl:operation elements. (*Note*: Renaming the overloaded term wsdl:operation as wsdl:exchange while keeping its semantics was discussed by the WSDL working group, but ultimately rejected.)

Again, these should not be treated like the methods of a Java/C++/C# interface/class although once again current tool support tends to treat them as such. Instead, a wsdl:operation represents the set of messages that are exchanged as part of one of the supported interactions. The alternative approaches are shown in Listing 1.

> “Evolving the WSDL document, on the other hand, does not need to break existing consumers”

It is important to understand that a message exchange is described in terms of the information that is on the wire. A WSDL interface makes no assumption about any state changes or correlation between messages, nor does it make any assumptions about what processing occurs in the service's implementation during a message exchange. In that sense, a WSDL interface is far less tightly coupled to its implementing service than the interface of an object that is intimately bound with its instances.

Example

To illustrate these issues, we'll look at a simple library application built using both an object and a service-oriented approach. First, the IDL documents for very simple library and book objects are shown in Listing 2.

It is assumed that the in-memory representations of the Library and Book objects are accessed through the methods available. It does not matter whether objects are passed or returned by value or by reference. There is always the shared understanding on how these objects are represented in memory and how they can be accessed.

Irrespective of different platforms or programming languages that are used to build the system, it is always the case that there are objects that can be accessed through the defined interface. For example, when CORBA is used the CORBA type system abstracts the use of different implementation technologies. The CORBA type system defines the shared abstractions between all the components of the system we are building.

A Library Web service advertises the contract that it expects its consumers to honor. The contract in Listing 3 defines a possible interaction with the service and the formats of the two messages that make the exchange pattern. The contract only talks about the on-the-wire representation of a book and the ISBN. No information is given about the way in which the in-memory representation of the book and the ISBN are accessed. The Web service and its consumer share only the knowledge of XML and XML Schema and so their boundaries are respected. Moreover, when we decide to evolve the application, we don't necessarily have to break existing consumers.

Let's say we want to get the title from a Book object. In order to add this new feature to our application, we will have to change the IDL (see Listing 4) and recompile the entire application since a shared abstraction has changed. It will not be enough to compile only those parts of the system that use this functionality. The change in the shared abstraction may impact the in-memory representation of every Book object in the system.

Evolving the WSDL document, on the other hand, does not need to break existing consumers. That's due to the flexibility of XML Schema and the fact that we do not share abstractions (see Listing 5). While it's true that there will have to be engineering work to upgrade the service to support the notion of a book title, this does not break the application as a whole since no in-memory representations or references to such are being passed around – only conformant XML messages.

Conclusion

Like XML Schema that can be used as a document validation mechanism or a meta-level type system, WSDL is broad enough in its scope to be (ab)used as something akin to an object IDL or as a contract description language. While there are proponents of the former, the simplicity of the contract view of WSDL is compelling since it is a natural fit with the loosely coupled nature of Web services–based applications.

Though WSDL 2.0 may yet provide the features necessary to support both views, we advocate the use of WSDL as a means of describing message exchanges in which a Web service participates. We believe that this approach yields the most loosely coupled system architectures, which are more robust in the face of changing requirements than those systems that impose an object-like view of the world onto a service-oriented application.

References

- Box, D. "Service-Oriented Architecture and Programming (SOAP)," - Parts 1 and 2, talks from the MSDN TV archive: http://msdn.microsoft.com/msdntv/episode.aspx?xml=episodes/en/20030827SOAPDB/manifest.xml, http://msdn.microsoft.com/msdntv/episode.aspx?xml=episodes/en/20030902SOAPDB/manifest.xml. 2003.
- Box, D. PDC 2003. Talks on Indigo and Web services

About the Authors

Dr. Savas Parastatidis is the chief software architect at the North-East Regional e-Science Center (NEReSC), Newcastle upon Tyne, UK, where he is their expert in Web services and grid computing technologies and standards. Previously, he co-led the research work at HP's middleware division that led to the development of the world's first XML-based transactioning system, and represented HP during the early stages of the OASIS BTP standardization effort.

■■■ savas@parastatidis.name

Dr. Jim Webber is a former architect with Arjuna Technologies, where he worked on Web services transactioning and grid computing technology. He is the former lead developer with Hewlett-Packard on the industry's first Web services transaction solution. Jim is now taking time to travel, and to think and write about Web services technology.

■■■ jimwebber@hotmail.com

Listing 1: WSDL 2.0 element names and the alternatives that were discussed

```
<wsdl:interface ...>
  <wsdl:operation ...>
     ...
  </wsdl:operation>
</wsdl:interface>
<wsdl:interactions ...>
  <wsdl:exchange ...>
    ...
  </wsdl:exchange>
</wsdl:interactions>
```

Listing 2: Library and Book IDL documents

```
interface Library
{
   Book   borrowBook(in string isbn);
}

interface Book
{
   string getISBN();
   string getAuthor();
}
```

Listing 3: Library service WSDL

```
<wsdl:types>
  <xsd:element name="isbnMsg" type="xsd:string"/>
  <xsd:element name="bookMsg">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="isbn" type="xsd:string"/>
        <xsd:element name="author" type="xsd:string"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</wsdl:types>

<wsdl:interface name="LibraryInterface">
  <wsdl:operation name="borrowBook">
    <wsdl:input message="isbnMsg"/>
    <wsdl:output message="bookMsg"/>
  </wsdl:operation>
</wsdl:interface>
```

Listing 4: Evolution of the Library and Book IDL documents

```
interface Library
{
   Book   borrowBook(in string isbn);
}

interface Book
{
   string getISBN();
   string getAuthor();
   string getTitle();
}
```

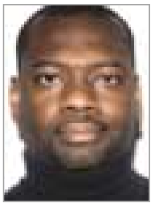
Listing 5: Evolution of the Library service WSDL

```
<wsdl:types>
  <xsd:element name="isbnMsg" type="xsd:string"/>

  <xsd:element name="bookMsg">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="isbn" type="xsd:string"/>
        <xsd:element name="author" type="xsd:string"/>
        <xsd:element name="title" type="xsd:string"
minOccurs="0"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</wsdl:types>

<wsdl:interface name="LibraryInterface">
  <wsdl:operation name="borrowBook">
    <wsdl:input message="isbnMsg"/>
    <wsdl:output message="bookMsg"/>
  </wsdl:operation>

</wsdl:interface>
```

Workstation Server Enterprise and Developer Editions from General Interface

An attractive alternative to the traditional Web tier

■ As Web services become more prevalent in the application tier and Web application sophistication increases, alternative approaches to the traditional Web tier are being developed. The Workstation Server product suite by General Interface provides a refreshing standards-based alternative for creating rich browser-based, event-driven interfaces to Web services (in IE 5.5+ and similar browsers).



WRITTEN BY
AHMED SAKO

The resulting applications look, feel, and perform like thick-client software (except there's nothing for the end user to install) by leveraging JavaScript Objects, XML, XSL, DHTML, and Web services. For this review, we will investigate the product suite components along with some patterns of use.

Product Overview

The General Interface Workstation Server suite consists of two components. The core component is the Workstation Server, a fast, lightweight (less than 100K compressed!), client-side container (written in JavaScript) for deploying and executing rich two-way Web applications in a browser. The development paradigm within the Workstation Server is based on a Model-View-Controller (MVC) architecture. The model is maintained in properties of JavaScript objects and a client-side XML data cache; the view is rendered via client-side generation of JavaScript, DHTML, and XSL (from the GUI objects); and the controller is the collection of event, data, and communication components implemented in JavaScript object classes.

The container provides, among other things, facilities for accessing Web services (via the "socket" metaphor), caching data (the cache is used for quick access to stateful application data), rendering GUI objects, and event dispatching.

Workstation Server sockets abstract interactions with the server by providing a uniform invocation mechanism for different request types (e.g., SOAP, XML GET, HTTP FORM POST, etc.). Sockets encapsulate the communication semantics, query parameter binding, and result-data interaction with the cache. Also, sockets allow for disconnected operations by either relying on cached data or using a specified XML document (e.g., to support cases when the Web service that the socket will ultimately communicate is being developed). The XML data, typically returned by sockets, is placed in the client-side data cache where it can be accessed by all aspects of the application.

Workstation Server applications are stateful and do not require an application server to repeatedly generate redundant HTML pages and deliver those to the browser. State is maintained on the client via instance variables and cached XML documents. A rich API is available to manage this cache but is beyond the scope of this article.

The Workstation Server includes a set of standard GUI objects (ListView, TreeView, ContextMenu, etc.). These are JavaScript objects (with their instance data stored in XML documents) that generate the appropriate DHTML code at runtime, and may be manipulated at runtime via a robust object-oriented API. An object inheritance framework allows the GUI object set to easily be extended to create new objects and subclasses of your own. The GUI objects support several nontrivial features, including drag/drop functionality, with some objects that have the ability to be bound to a socket directly. To wire an individual event handler to a control, the JavaScript code is assigned to the desired event in the properties editor for the control. Because JavaScript is the foundation of the tool, it can be entered almost anywhere (e.g., JavaScript can be entered directly into some properties along with HTML).

Building on the power of General Interface's object inheritance framework, the Workstation Server includes an object-oriented JavaScript API for managing instances of in-browser objects created by the Workstation Server. Object-minded programmers will appreciate the familiar feel of the APIs that enable high-level and low-level access to the object properties and methods.

The other Workstation Server suite com-

Company Info

General Interface Corp.
212 Sutter Street
Fourth Floor
San Francisco, CA 94108
www.geninterface.com
Phone: (415) 394-6150
Fax: (415) 394-6151
Sales: info@geninterface.com

Download Information

A free trial of the General Interface Workstation Server is available at www.geninterface.com/eval ponent, available in the Developer Edition, is a lightweight IDE used to develop the UI and integrate the application components. Using the IDE, a developer can lay out UIs, create new GUI Objects, assemble applications and connect them to XML/Web services. The IDE provides a JavaScript editor, XML/XSL editors, a JavaScript debugger, several wizards, inspectors (e.g., a GUI Object HTML viewer to inspect the rendered HTML) and a socket wizard. This wizard includes an integrated WSDL inspector that auto-creates the SOAP input structure.

It is important to note that while all development could be done without using the developer edition and leveraging the Workstation Server's extensive API, hitting "CTRL-J" while running an application from the Developer Edition will bring the application up in the IDE, allowing you to view and modify the instance data, and even the currently executed code (see Figure 1). This enables you to perform on-the-fly debugging and incrementally build applications with changes being immediately reflected in the application.

Installation and Application Deployment

Installing the Workstation Server is as simple as unzipping the software and placing the contents into a directory of your choice. The Workstation Server and the applications developed can be served from any industry-standard HTTP server, but a Web server is not required as applications can be run directly from the disk.

Developing Using the IDE and Wizard

Developing an application typically involves developing the UI, accessing the application tier using the socket wizard, and integrating the two using JavaScript or XSL/XSLT. (*Note:* The first two steps can be performed in any order.) For example, I've created a simple Web service that allows you to obtain weather information for a given zip code. This weather information would then be displayed on the contact screen (see Figure 1) via an additional text box when the "Get Weather" button is clicked.

To add this functionality, a "Get weather" button, a text box (to display the result), and a hidden blockX component (used to communicate with the Web service) were added to the screen. The socket wizard for the blockX was then invoked, which streamlined the creation of the SOAP message to be sent to the server. The wizard also allowed me to define where to cache the result data and use some JavaScript to access the value in the zip code text box.

Afterwards, an XSLT file was created to extract and format the information. This XSL file and a reference to the XML document (created when the blockX object is refreshed) were then assigned to the text box using the property editor. Finally, the JavaScript code to refresh the blockX object was placed in the "onClick" event for the "Get weather" button to complete the loop (see Figure 2).

Summary

The Workstation Server Enterprise and Developer Editions from General Interface allow the creation of enhanced user interfaces for Web-based applications, which enables richer user interaction akin to that experienced with desktop applications.

The IDE and API take a little getting used to and are geared towards more seasoned object-oriented developers with some XML/XSL know-how. However, the rewards are worth the effort. For rich Internet Explorer–based applications, the General Interface Workstation Server Enterprise suite is an attractive alternative.

About the Author

Ahmed Sako is CTO of Rosenblatt Securities. His areas of expertise include distributed systems, transaction processing, application architecture and development, system analysis, and Web design.

■■■ asako@rblt.com

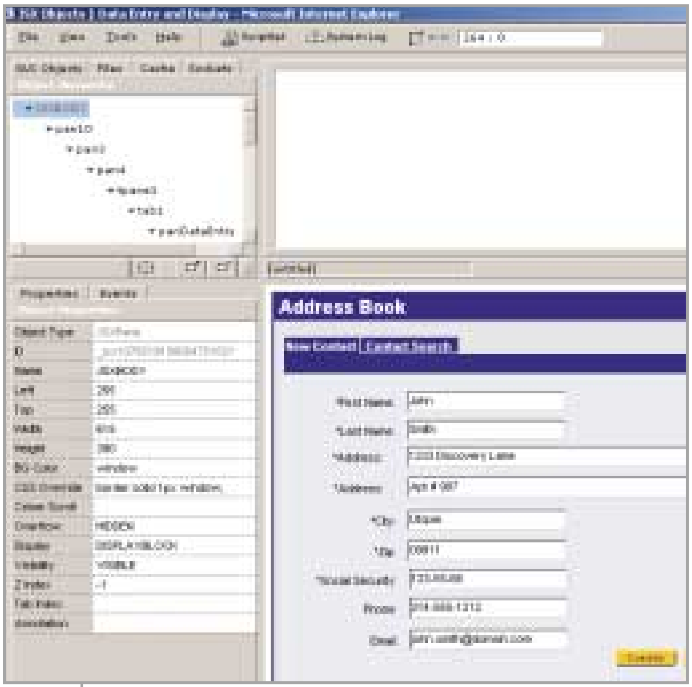


FIGURE 1 Contact application after keying 'CTRL-J' to access the IDE

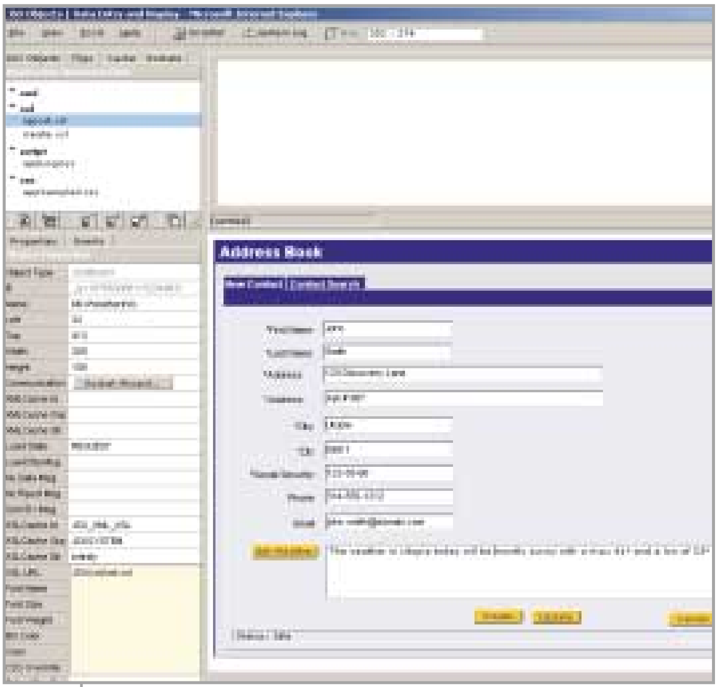
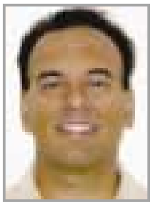


FIGURE 2 Contact application with weather information

Design Strategies for Web Services Versioning

Adapting to the needs of the business

Application versioning has always been a challenge for the developer community. With the introduction of Web services, this issue becomes even more difficult as developers are dealing with a more distributed set of components that aren't necessarily under their control.



WRITTEN BY
CHRIS PELTZ &

ANJALI
ANAGOL-SUBBARAO

A robust versioning strategy is needed to support multiple versions of Web services in development. This can allow for upgrades and improvements to be made to a Web service, while continuously supporting previously released versions. The right versioning strategy can maximize code reuse and provide a more manageable approach to the naming, deployment, and maintenance of your Web services.

The issue of versioning is a complicated one, and this article does not attempt to answer every question surrounding the versioning of XML and Web services. However, there are some key approaches and design practices that have been helpful in our development that we will share with you here. The approach we will take is to start at the component level (XML Schemas) and work up to higher service-level abstractions, including facades and service-oriented (SOA)–based architectures. Along the way, we hope to impart some important best practices that can be applied to incorporate versioning techniques at various levels of the design.

Let's begin by looking at the importance of XML Schemas…

The Importance of XML Schemas in Versioning

Versioning for Web services should first consider how the XML Schemas being used are versioned. A number of approaches can be taken to version XML Schemas, including using a schema version attribute, changing the physical location of the schema, and leveraging XML Namespaces.

XML Namespaces provide a scoping mechanism for XML, where elements defined in that namespace can be uniquely identified. Namespaces are typically used to reduce name collisions between schemas, but they can also be used as a version control mechanism. Here's a simple example showing how Namespaces can be used for versioning:

```
<xsd:schema targetNamespace=
http://www.acme.com/types/pcconfig/v1
```

A Web service referencing a versioned schema would be required to have knowledge of the version because it is built intothe targetNamespace. The benefit of this approach is that once the targetNamespace is updated, any clients that reference it are required to do something. This automatically enforces a level of version control, requiring some action on the client's part.

The downside of this approach is that any incremental change would require the client to also change. Clearly, a noncompatible change such as removing a WSDL operation being used by the client requires client modification. But, a change such as adding an optional attribute to the schema would still be compatible with the existing code. One key question that must be addressed in your design is: What changes should constitute a new Web services version? If you take the approach that any change results in a new Namespace, this will place a great burden on the developers using the schemas.

One proposed hybrid solution is to use a combination of Namespace and version IDs. The targetNamespace would only be updated for non-compatible changes. For any incremental, compatible change, the version ID attribute could be used to identify

the new revision. The following approach can significantly reduce the amount of maintenance required by the developer when new versions of a Web service are released.

```
<xs:schema xmlns=
http://www.acme.com/types/pcconfig/v1
targetNamespace=
http://www.acme.com/types/pcconfig/v1
version="1.2">
```

Version compatibility can be a very difficult thing to determine. Without any robust development tools available for this, it is the responsibility of the development team to determine whether a new release maintains compatibility. In this section, we've presented a number of approaches to versioning XML Schemas. If you're considering versioning at this level of your architecture, we would recommend use of XML Namespaces in a limited capacity to indicate major version upgrades to the XML But, it is a good practice to fully test your services to verify version compatibility.

Naming Conventions for Versioning

Having a name for each release of a Web service is essential for simplifying the management of the releases. The naming strategy should facilitate recognition of multiple versions of a Web service, including both existing and new interfaces. Different strategies can be implemented here, including using a sequential naming strategy or date stamps.

When you are sequentially naming your version you can use a convention of "vMajor#.Minor#/SERVICE_NAME, where Major# is the major version release and Minor# is the minor number release. A major release would most likely require a change in the client code to use the new version, while a minor release would attempt to maintain backward compatability with the client. A minor version could constitute a modification to a Web service that would impact the internal system only.

The standard method name for a Web service has "_v#_#" appended to the end of its name. Thus, for the getProductCatalog service we might define the operation as "getProductCatalog_v1_1". Or, if you are applying this to XML namespaces, it might look like:

```
<xsd:schema targetNamespace=
http://www.acme.com/types/pcconfig/v1/1
```

Another way of naming versions is to use date stamps as part of the namespace. The date helps to determine the sequencing of the version. This naming convention does not show a major or minor release. Here's how this might look in our definition of the schema:

```
<xsd:schema targetNamespace=
http://www.acme.com/2004/03/01/pcconfig
```

The important thing to remember is that there isn't one best solution in how you name the versions. Your approach will depend on the deployment strategy you use and the specific requirements your clients might have to distinguish versions and maintain compatibility.

It should also be apparent that these techniques can become very unmanageable if you have to apply them to every component in your design.

To address this, let's turn to a few important design practices for building more coarse-grained services with design patterns and service-oriented architectures.

Using Web Services Facades

As organizations build, deploy, and orchestrate Web services, it becomes apparent that a higher level of versioning and management is required. You must begin thinking about services from a business level, rather than from the technical interfaces being exposed. However, many organizations investigating Web services are still looking at them as extensions to object-oriented paradigms. They make the mistake of taking their existing objects and exposing them directly with a Web services interface. This will most likely deliver Web services that are fine grained and difficult to use. It can also make it difficult to incorporate a consistent approach to Web services versioning.

Instead, think from the perspective of the WSDL first, and map to a back-end implementation later. This approach provides a more loosely coupled architecture, minimizing dependencies on specific implementations. You should also look to design patterns, such as the Web Services Facade, to assist in the creation of coarse-grained components. This pattern takes the complexity out of the interfaces being exposed. Rather than having multiple interface points to a collection of services, composite Web services can be designed with simpler, well-understood business interfaces.

As Figure 1 illustrates, the facade can be used to simplify the steps required to place a Change Order request. While this picture shows the facade being used from the perspective of the provider, a consumer of could also design facades to simplify access to services.

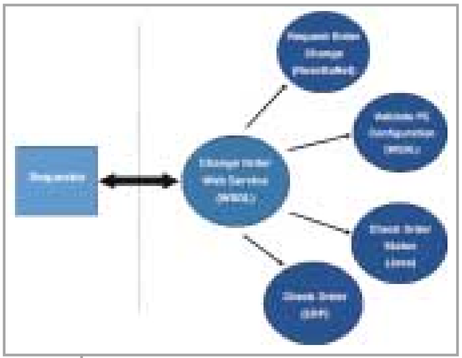


FIGURE 1 | A simplified Change Order request

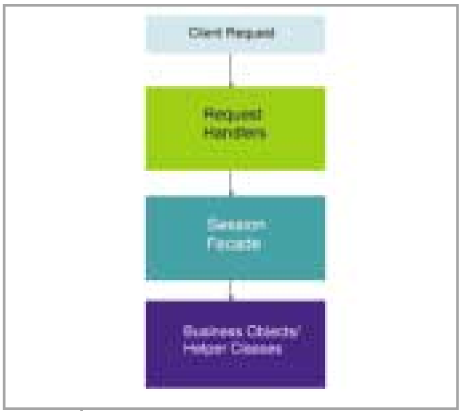


FIGURE 2 | Facade pattern implementation

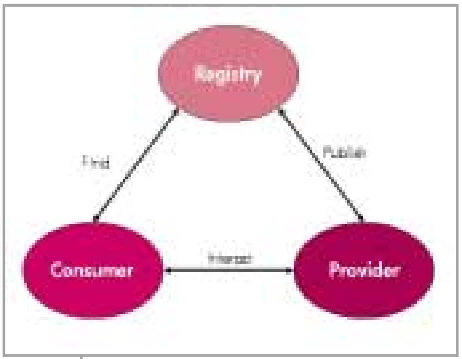


FIGURE 3 | SOA model

The use of this pattern can offer a number of benefits, including minimizing the level of coupling between consumer and service, creating a control point for manageability, and improving overall performance by reducing the number of network calls. The facade pattern can also provide a framework for managing the inherent complexity of supporting multiple Web services versions simultaneously.

One implementation of this pattern, shown in Figure 2, creates levels of abstraction by separating the code into three distinct tiers of classes: request handler classes, facade business flow classes, and business object classes.

The managing class, the session facade class, is responsible for chaining the smaller business object classes together. Each business object class contains a subunit of processing logic, and chaining of all these business object classes provides the core business logic. Each session facade class by itself contains no processing logic, but by concatenating business objects together it implements the desired business logic.

> “Life-cycle management is much more than just managing the deployment of Web services”

This architecture decouples processes from each other, allowing reuse of components. It also helps in managing the complexity of Web services as business requirements change in different versions of the service. Processes leveraged to execute business requirements are abstracted into individual pieces of software code that one managing class chains together to execute in sequence. By chaining these pieces of code together, required business processes are implemented to adhere to a specific business process flow. Thus, by building multiple managing classes, different business flows can be implemented and maintained simultaneously.

Importance of Service-Oriented Architectures

The design of coarse-grained services is an important aspect of a versioning strategy, simplifying the client interfaces and maximizing reuse. Designing facades at the implementation level can help, but it still requires a significant change to the underlying infrastructure at various levels that someone has to deal with. What's truly needed is an SOA-based platform that can assist in the management of the entire life cycle of Web services.

An SOA-based approach offers software components as a collection of distributed business services. A business service plays the role of a single entity that represents a business application, and may include multiple IT resources, platforms, and components. The model, simplified in Figure 3, separates the roles of consumer, provider, and registry. A provider registers available services in a registry that the consumer can later discover and invoke. Consumers aren't directly aware of a specific service endpoint or the implementation details. The SOA provides a greater degree of separation between the provider and consumer, and as changes are made to a service, the SOA can help minimize the impact on the consumer.

An SOA typically requires some type of framework or platform that can act as an intermediary between providers and consumers. Without this intermediary in place, consumers would have to build a more tightly coupled integration with a service endpoint. This intermediary could be implemented as a proxy, a gateway, or a broker, handling issues such as security, routing, SLA management, and life-cycle management.

The life-cycle management features offered by the platform are key to our discussion of versioning. Some Web services and SOA management platforms on the market today offer support for life-cycle management. This might include:

- The ability to handle multiple versions of a service at the same time
- Deprecating or retiring services in a graceful manner
- Routing requests to the appropriate service endpoint based on a request
- Transforming requests and responses to maintain backward compatibility (e.g., using XSLT)
- Automated deployment of new versions without requiring system shutdown
- Managing dependencies that exist between different Web services

Life-cycle management is also much more than just managing the deployment of Web services. From the perspective of a business service, life-cycle management might include many platforms, application servers, and database instances. To truly manage this level of a service, you must consider how all dependent application components and infrastructure elements are deployed and managed.

Let's look at an example showing a common versioning problem. In Figure 4, a provider has offered an initial version of a Web service, V1.0. The platform would automatically route requests for this service to an appropriate service endpoint. Routing could be determined based on information placed in a UDDI registry, or the underlying management model could manage this (e.g., through WSDM interfaces).

At some point, a second version of the service is deployed that maintains backward compatibility. Within the SOA, the new service is identified as supporting both the original business service and the new business service. Requests made for the original business service could be brokered to either service endpoint. Consumers wishing to access the new functionality would have to make the appropriate programmatic changes.

This approach to versioning allows an IT organization to better adapt to changes required by the business. Multiple versions of a service can be maintained simultaneously without impact to the consumer. Old versions can be retired gracefully, and consumers can upgrade on their own time, rather than being forced to when new versions are released.

Deployment Considerations

Even with a solid design and robust architecture for versioning, there is still a human factor that must be considered in this process. Understanding the versioning life cycle will help you to implement new versions and deprecate the older versions effectively. The first step in this process is to put a solid plan together describing the life cycle of Web services being supported. This plan might include the following steps:

1. ***Determine how often versions are to be released.*** When considering frequency, you should consider how many versions of the Web service you want to support in parallel.
2. ***Understand the timeframe within which you expect consumers to move to a new version of a service.*** The Web services management platform may be able to provide guidance on service usage to determine the appropriate time to phase out older versions.
3. ***Consider releasing a pilot or an early release of a new version***. Give consumers an opportunity to test compatibility and determine potential code impacts.
4. ***Approach Web services versioning the same way software packages might be released.*** Changes to your service, either as a result of bug fixes, partner requests, or specification upgrades, should follow a specific release cycle.
5. ***Clearly communicate your Web services versioning strategy to users of your Web service.***

Once the versioning strategy is laid out for each release you should then consider the following steps:

1. After making the appropriate changes to your services, do unit and functional testing of the service.
2. Deploy the new service through appropriate schema, WSDL, and service changes. This step might include registration to a UDDI registry or the Web services platform.
3. Notify the consumers of your new service and pilot the new versions with one of your consumers.
4. Run the new and old versions in parallel for the timeframe you have allocated in your versioning plan.
5. Notify the consumers of your service of the date you expect to deprecate old versions.
6. Remove old versions from WSDL descriptions and UDDI registries to prevent new consumers from discovering and using the older versions.
7. Remove functionality of the old service, and add in appropriate functionality so existing consumers are properly notified (e.g., through a SOAP fault) that the old version is no longer supported.

One approach to deployment is shown in Figure 5. The calendar is shown in quarters of a company's financial calendar. The new version can be introduced at the beginning of a quarter (e.g., Q1 of 2003). It is valid for four quarters and then is sunset at the beginning of Q1 of 2004. In this case, the consumer of the service could be given six months to convert to the new version. You can also have minor versions released in between the major versions. These minor releases can then be folded into the Web service version 2.0 when it is released.

Conclusion

This article has presented a number of considerations when versioning Web services. Whether it's developing a versioning approach for schemas or building a full-scale versioning system using an SOA approach, there are a couple of key points that we would like to leave you with.

First, we hope it's obvious that versioning can become a nightmare as you deploy more and more versions of your service, especially if the service is being offered to different consumers. The goal of your design should be to build reusable components that can be leveraged and reused across the versions being managed. In the long run, this will reduce maintenance and deployment costs.

Second, don't underestimate the impact these changes might have on the consumer. While the design you introduce might reduce your development costs, you must also consider how these changes impact the end consumer. Service compatibility is key to this, and if new services maintain backward compatibility, consumers wishing to use the existing functionality shouldn't be required to change. The architecture should provide an easy migration and upgrade path for them.

Third, you want to strive to isolate the technical implementation and provide a clear separation between the consumer and provider. This might be implemented using design patterns such as the Facade, or it could require an SOA-based approach. Either way, the notion of coarse-grained business services offers a more loosely coupled architecture where consumers and providers can change without impacting the other party.

The last piece of advice we could offer is to not attempt to build the entire versioning infrastructure yourself. If you look at the requirements for manageability, they include a wide range of features such as routing, transformation, versioning, and security. Leveraging a vendor offering, especially one that provides a service-oriented approach IT management, can go a long way toward offering an architecture that can truly adapt to the changing needs of the business.

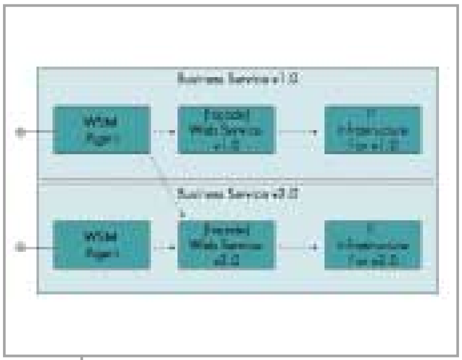


FIGURE 4 Initial version of Web service

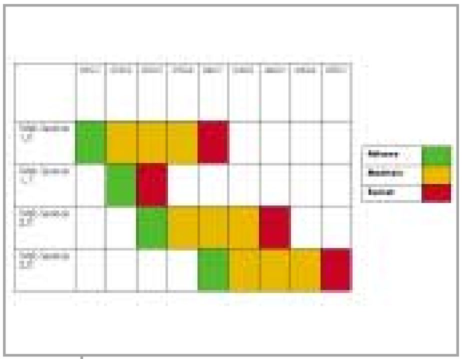


FIGURE 5 Version scheduling

References

- Costello, Roger. *XML Schema Versioning.* www.xfront.com/versioning.pdf
- Costello, Roger. *Multi-Schema Project: Zero, One, or Many Namespaces?* www.xfront.com/zerooneormanynamespaces.html
- Orchard, David, and N. Walsh. *Versioning XML Language*. www.w3.org
- OASIS Web Services Distributed Management TC: www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsdm"
- HP Management Integration Platform: www.managementsoftware.hp.com/products/mip/.

About the Authors

Chris Peltz is a software architect in HP's Developer Resources organization (devresource.hp.com), providing technical solutions around Web services, service-oriented architectures, and adaptive management.

■■■ chris.peltz@hp.com

Anjali Anagol-Subbarao is an IT architect in the HP's Personal Systems. Her team has successfully deployed Web services to enable real-time business transactions between major retailers and HP.

■■■ anjali.anagol-subbarao@hp.com

untangle
untangle your integration woes with
Assande
EAI Messaging Suite

Sarbanes-Oxley and Web Services

Act now to get an early grasp of the inevitable

This article makes the case that Web services provide a significant benefit to Sarbanes-Oxley compliance projects, and that they will therefore be used extensively on these projects. We begin with a very brief primer on the Sarbanes-Oxley Act, then describe the connection between SOX and Web services, including an outline of how most Sarbanes-Oxley projects are conducted, and where Web services fit in. Finally, I offer some specific actions you can take today to get yourself ready for Sarbanes-Oxley.

WRITTEN BY
ANDY ASTOR

A Sarbanes-Oxley Primer

The Sarbanes-Oxley Act of 2002, which applies to all companies traded on U.S. stock exchanges, was enacted into law in response to financial scandals such as Enron, MCI, and others. The law puts into place tough requirements and penalties to ensure that companies' financial statements accurately represent their business position.

There are numerous sections in the Sarbanes-Oxley Act. However, the three that concern us here are Sections 302, 404, and 409. As shown in Figure 1, these are successively steeper hurdles that are being phased in over time.

- Section 302 states that CEOs and CFOs must personally sign off on their companies' financial statements. Few specific controls are required by Section 302. The point of it is to establish CEO/CFO accountability for the rest of the Act's sections, with the possibility of prison for noncompliance.
- Section 404 mandates that well-defined and documented processes and controls be in place for all aspects of company operations that affect financial reports. Furthermore, executive management and a company's auditors must each state in writing that these processes and controls have been examined and are effective. Any findings of ineffectiveness must be publicly disclosed. For companies whose net worth exceeds $75 million, this rule goes into effect beginning with fiscal years ending June 2004. In other words, right now.
- Section 409, which is not yet in force, will soon require real-time public disclosure of all events that could materially affect company financial performance. To comply with this rule, companies will first have to recognize in real-time that significant events have taken place (e.g., a cash-flow problem), and then get that information into a public reporting system.

Sarbanes-Oxley and Web Services

At first glance, the connection between Web services and Sarbanes-Oxley is not obvious. After all, what do Web services have to do with financial reporting? Indeed, most of today's Sarbanes-Oxley compliance projects are being run by finance departments, with little or no IT involvement.

But recall that Section 404 dictates that controls be in place for all material impacts to financial statements. In other words, every significant business transaction – most of which span multiple corporate systems – must be modeled and inspected for risk; if risks are found, they must be mitigated. Furthermore, Section 409 requires real-time recognition and disclosure of material events. How will these requirements be met?

As Figure 2 implies, most public companies are large, distributed organizations, with diverse systems that have been built up independently over time. To bring information from these systems together, such as for producing financial reports, multiple strategies are generally used, often including point-to-point connections and manual systems. For example, a distribution center in Ohio might send a file or a set of transactions to a headquarters accounting system in Dallas. Or spreadsheet summarizations might be used to consolidate output from multiple plants. Even pencil-and-paper manual controls are common.

These point-to-point and manual systems often get the job done, but they are not up to the requirements of Sarbanes-Oxley. They can expose an organization to problems such as:

- Inconsistent policy and control implementations
- Rekeying errors across multiple systems
- Omitted or double-posted transactions due to failed processes
- Inconsistent or nonobjective manual reviews and approvals
- Lack of reconciliation between unintegrated systems
- Unusual events not flagged for follow-up

In a nutshell, these point-to-point and manual controls (a) fail to document the specific links between systems; (b) do not enforce controls with enough rigor; and (c) can be extremely error prone.

The simple truth is that most corporations cannot achieve the requirements imposed by Sarbanes-Oxley without a strategy for automating the integration of the diverse business processes and systems throughout the enterprise. Web services and other integration technologies can be that link, providing the capability to establish control and documentation, reduce risk and error potential, and lower control costs.

Sarbanes-Oxley is often viewed as a burdensome business regulation that provides little or no business value. Consider, howev-

Gartner Application Integration & Web Services Summit

May 17-19, 2004
Los Angeles, California

FREE to every Summit attendee: Gartner 2004 Guide to Application Integration and Web Services Products and Vendors

Making Business Processes and Service-Oriented Architecture Pay Off

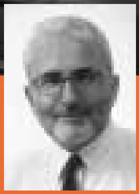
Topics covered:

- Application Integration
- Web Services
- Service Oriented Architecture & Applications
- Business Process Management
- Enterprise Information Integration
- Aligning IT and Business
- Portals
- Event Processing
- Measuring and Improving ROI
- Security Issues

and more!

Keynote Speakers

James Burke
Science Historian, Author, Producer



Peter Weill
Director of Center for Information Systems Research, MIT

Roy Schulte
Vice President and Distinguished Analyst; Conference Co-Chair

David Smith
Vice President and Research Fellow; Conference Co-Chair

Jeff Schulman
Group Vice President

- 40 in-depth sessions led by 30 leading experts
- Real world case studies
- The latest unbiased research and best practices
- Solutions from 50 leading vendors

Plus!

New Special Program
Enterprise Architecture:
New Challenges, New Solutions

and more!

Register Now: gartner.com/us/aiws and use Promotion Code APNWSJ

er, a different view: Sarbanes-Oxley compliance efforts can transform an enterprise into one that ties together every person, computer system, and business process; one that routes information to and from the right people, in the right places, at the right times. In other words, the Sarbanes-Oxley Act can be seen as a compelling event that provides the impetus to accelerate business responsiveness, streamline supply chains, and enable better decisions. If this isn't a job for integration and Web services, what is?

IT and Sarbanes-Oxley

If Web services are so important to Sarbanes-Oxley compliance efforts, why aren't we seeing them (and other integration technologies) in more of today's compliance projects? Indeed, most of today's Sarbanes-Oxley compliance work is being driven by finance organizations, company auditors, or both. IT is rarely involved, except to execute the plans created by Finance/Audit. The reason for this conspicuous absence is twofold.

The first reason is simply a deadline issue. For most companies, Section 404 requirements must be met in the current fiscal year, and executives face jail time if their firms are found out of compliance. This means that finance organizations are scrambling just to get their systems documented and to plug the biggest risk areas, usually with manual fixes. Prison is a powerful motivator for getting a job done, even if the solution is not particularly elegant on the first round.

A second reason that IT is not yet involved in many Sarbanes-Oxley projects is that these projects tend to follow a natural progression of three phases, and the value of integration and Web services is not evident until the third one. In other words, many companies simply haven't gotten around to the point where Web services are needed. The three phases common to most Sarbanes-Oxley compliance projects are:

1. ***Assess and prioritize subject areas:*** Financial statements are reviewed to identify line items at risk for fraud or error. These items become candidates for immediate evaluation and, where necessary, remediation. For example, salaries might be deemed a low-risk item since they are tightly controlled by a small group of people. Revenue recognition, on the other hand, might be deemed high risk because of loosely defined recognition procedures. This phase is really about analysis and prioritization.
2. ***Document and evaluate business processes:*** The business processes deemed most critical in phase 1 are documented and then evaluated for fraud and error potential. Several technology-based solutions are available to enable companies to graphically model these processes and to use these models to evaluate missing or inadequate control points. But this phase is still mostly financial analysis.
3. ***Remediate and improve control systems:*** As control weaknesses are discovered in business processes, system changes and/or automation are added. This is where Web services come in.

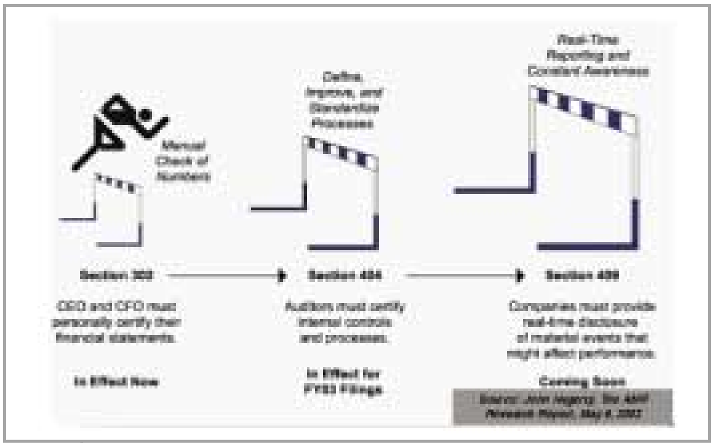
What Do Web Services Bring to the Table?

As projects enter this third phase, Web services and other integration technologies become key implementation enablers. Using Web services, for example, manually prepared spreadsheet summarizations can be eliminated in favor of direct system-to-system communication, yielding tightly controlled audit trails for Section 404 compliance. Similarly, real-time alerts can be defined via Web services to recognize and report on unusual events, to comply with Section 409. All told, Web services address three of the most important business drivers in Sarbanes-Oxley compliance projects:

- ***Control enforcement:*** Automated control of both computer-driven and people-based systems is far more effective than written procedures, since such controls cannot easily be subverted. For example, in the event of a credit risk override, a real-time alert sent to the risk management staff can prevent a poor decision from becoming a business disaster.
- ***Real-time reporting:*** The real time reporting requirements of Section 409 are coming soon, and businesses must have automation in place to handle them. For example, Section 409 will require real-time public disclosure of material events such as significant write-downs or bad debt recognition. Automated alerts provided by Web services can ensure that such events are communicated immediately to the appropriate executives.
- ***Cost reduction:*** Initial Sarbanes-Oxley compliance costs may be high simply because many of the controls put in place will be manual or one-off efforts. To reduce these costs, IT organizations will need to drive these controls into automated enterprise-class systems. For example, the cost of evaluating customer credit risk can be reduced by integrating credit history and external credit checks into the order processing system.

Actions to Take Today

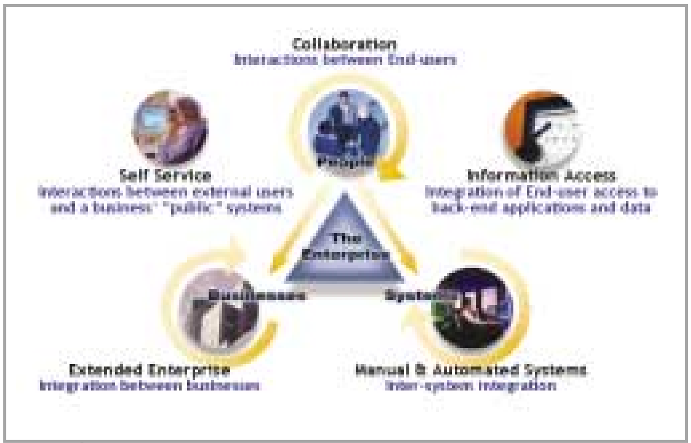
If you are involved in Web services development at a public corporation, be assured that Sarbanes-Oxley is a force headed in your direction. Within the next year, many



Real-Time Reporting and Constant Awareness
Define, Improve, and Standardize Processes
Manual Check of Numbers
Section 302
Section 404
Section 409
CEO and CFO must personally certify their financial statements.
Auditors must certify internal controls and processes.
Companies must provide real-time disclosure of material events that might affect performance.
In Effect Now
In Effect for FY03 Filings
Coming Soon

Source: John Hagerty, The AMR Research Report, May 8, 2003

FIGURE 1 | **Sarbanes-Oxley compliance hurdles**



Collaboration
Interactions between End-users
People
Self Service
Interactions between external users and a business' "public" systems
Information Access
Integration of End-user access to back-end applications and data
The Enterprise
Businesses
Systems
Extended Enterprise
Integration between businesses
Manual & Automated Systems
Inter-system integration

FIGURE 2 | **Interactions between end users requires collaboration** of your company's systems will need to be tied together in new ways, and Web services can be a critical architectural enabler. Be ready for these changes. Following are a few ideas for actions you can take today.

1. ***Get involved:*** Learn about the Sarbanes-Oxley efforts in your organization. Identify the people in your finance organization working on Sarbanes-Oxley (they are there, I promise), explain how IT will be important to them in the near future, and ask to get involved proactively.
2. ***Service-oriented architectures:*** Build your systems with a service-oriented architecture to maximize flexibility and adaptability. When the Sarbanes-Oxley projects come your way, you'll be ready.
3. ***Build-in auditability:*** Remember that Section 404 doesn't just demand integration; it requires controls over that integration. Controls generally translate to auditability: the ability to know exactly what happened, along with when and why. Build these capabilities into your systems today.
4. ***Tools:*** There are many solutions on the market today that claim to help with your Sarbanes-Oxley efforts. For the most part, they are designed for phases 1 and 2, which are primarily carried out by finance organizations. The tools that will be most helpful to you in implementing phase 3 (remediate and improve control systems) are Web services infrastructure and integration tools. Look for standards-based tool sets that are staying on the leading edge of Web services technologies.
5. ***Choose your partners carefully:*** Chances are, your systems integrators and auditors have already been chosen by your finance organization. Choosing your technology partners, however, will probably fall to you. Be sure that your provider is a public company, itself subject to the Sarbanes-Oxley Act. It should understand Sarbanes-Oxley "personally." It should also possess an unblemished track record of conservative financial practices without pending shareholder or other lawsuits, and should have pre-existing partnerships with your Sarbanes-Oxley advisors (e.g., auditors and systems integrators).

Final Thought

It's important to recognize that compliance with Sarbanes-Oxley is not a one-time event or project. This is a process that will be ongoing for many years to come. Sarbanes-Oxley compliance is here to stay, and it will impact every major system in every public corporation for the foreseeable future.

It's sort of like Y2K, only this time there is no end in sight.

About the Author

Andy Astor is vice president, strategic solutions at webMethods. In this role, he is responsible for driving the company's strategy and execution in key strategic areas, and for evangelizing webMethods' position in the marketplace. Some of his focus areas since joining webMethods in 2002 have been Web services and the Sarbanes-Oxley Act. Andy was recently elected to the Board of Directors of WS-I and is also on the International Advisory Board for *Web Services Journal.*

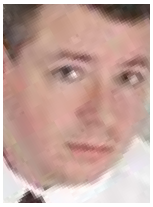
■■■ aastor@webmethods.com

HostMySite.com
TM
Built by Development Pros for Development Pros
• 24 / 7 / 365 Phone Support
• 99.9+% Uptime
• "Full Control" panel
• All Major Scripting Languages
• SQL Server, ACCESS, MySQL
• Turnkey Ecommerce Solutions
• Reseller Incentives
plans from $8.95 / mo.
FREE Domain Name*
FREE Setup
FREE 2 Months
Visit www.HostMySite.com/wsj for:
2 Months Free
FREE Setup and FREE Domain Name
*on any annual shared hosting plan
call today
877•248•HOST
(4678)

A Web Services Value Proposition

Lower costs, much higher benefits

Even if you work for a great company, if it isn't a technology company it will mean that no matter how cool, obvious, or forward thinking technological advances are, you will always have to be able to translate their benefits into real business dollars and sense.



WRITTEN BY
BY DAVID LADY

Web services presents one of the simplest and most innovative technologies I have come across. Its potential looks amazing and it offers little down side that I can see. It makes sense to adopt it as a systems integration platform of choice because it represents two things that our business is interested in:

- Cost savings
- Agility

I'm going to talk about Web services as an integration technology, generally focusing on using it to pull legacy systems into a service-oriented architecture model. As I have gone through this journey, I can recommend dozens of things that you should have an understanding of if you are looking to adopt Web services for integration use; however, there are three things that I think are at the top of the list:

1 ***Have a value proposition:*** Be able to show how the Web services integration architecture saves money in your company.
2. ***Have a plan for all of your systems, not just the new ones:*** Including legacy systems in your Web service architecture has many benefits.
3. ***Know what your strategy is going to be for data before you make any other decisions.*** This strategy will have substantial impacts on your design and scope.

I want to share what I believe to be a sound way of presenting these important business ideas to senior management. In this article I'll go over the models I have built to help translate the obvious technology benefit of Web services into language that works to cover the business costs of putting them in place. After that, I'll talk a little bit about how to get creative so you can use Web services for older legacy systems that needed to be included in a service-oriented architecture. Finally, I'll touch on some enterprise-wide data integration basics to get you thinking about a strategy.

The Problem Statement

Many companies today have a lot of different technologies, and a substantial part of there enterprise data in legacy systems. The industry is evolving, slowly, into more modern commercial-off-the-shelf (COTS) packages. Companies that are all COTS based I call an Application Oriented Architecture. This may have been the grand industry vision at one point, but somewhere along the way the vision changed (or was incomplete to start with), and has left many with new and legacy systems that do kind of the same thing. This creates a lot of integration needs to give the business a complete view of their lines of business.

Systems need data and we have to come up with cost-effective solutions for a blend of technologies. Additionally, we have to justify the cost of these solutions against a preestablished business benefit that has only moderate tangible (save money now) benefits, but an overwhelming set of elusive (save money later) benefits.

So as technologists our problem is twofold: find a set of integration technologies that offered a fit for a very wide array of systems, and describe the tangible benefits in a way that can be measured against existing financial cost expectations.

A service-oriented architecture (SOA) using Web services makes a lot of technical sense. It makes good business sense too. Since we are sitting somewhere between no architecture and an application-oriented architecture, the SOA model is the most efficient model to tie it all together. I think the SOA is the most complete vision of business application technology today.

Adopting a strategy to move to an SOA allows you to focus on the elements of business technology that matter most to the business.

A foundational model for maturing the enterprise integration architecture is:

```
Connect ‡ Integrate ‡ Optimize
```

One danger we have faced over the past couple of years is that the tactics used to connect systems, without a vision for an optimized environment, lead to point-to-point madness. Systems cannot be integrated, only connected, and reuse or other optimization is not possible.

A service-oriented architecture promises to speed development time and reduce integration costs. But for this to happen the services must be understood and implemented correctly. They must be understood from an enterprise perspective and organized so they can be reused. This is where the big payoff comes, as I'll talk about later. If planned properly, Web services for integration can be a very good answer indeed.

Now, how do we wrap this into language that can be used to communicate to a fast-moving, bottom line business?

Have a Value Proposition

Okay, I have the technology solution; how do I justify that this is a better business choice then a different architecture?

I've done a considerable amount of research analyzing how people spend their time on projects. Sometimes this can be a little bit of a challenge day to day, but it provided me with data I could analyze and determine how much on average current integration implementations were costing.

www.LinuxWorld.com
Taking the Cost Out of Firewalls: Knowing Your Linux Pays
LINUXWORLD
MAGAZINE
PREMIER ISSUE!
Why Migrate to Linux?
A comprehensive solution for enterprise IT
Facing Microsoft's Fury...
Taking the Geek Out of Linux
Exclusive Interview with Eric Steven Raymond
FREE
NOW
The Rise of Linux in the Enterprise
Big Gains in Small Business
The Leading Magazine for Enterprise and IT Management

For simplicity's sake let's assume you have the standard supply of FTP's, point to point, ETL's, users performing multiple entries, screen scrapping, and message queuing. If your inventory turns up similar to mine. You should now able to reduce your costs down to a quantifiable value that can be scaled by the number of touch points there are between systems across the enterprise.

I call this the Consumed Integration Point (CIP) Index. I use this term to identify points where an application will need to connect to or integrate with other applications to share data or processing. The CIP Index is calculated by counting the number of data consumption needs, per application, for data outside of the applications domain.

This index can then be tracked historically and trended forward over time.

> “Web services have lowered my integration development costs and positioned my architecture around agility”

Efficiencies to the integration architecture can be measured and costing trends established against those efficiencies. And here is the great part: when you start to forecast your CIP Index on existing technologies against a Web service–based integration architecture you will see that the costs went down – by a ***lot***!

The industry average for a new CIP pre–Web services costs just over $23,000. Since in our pre–Web services architecture an application needed to request and then receive the data, the complete work flow comprises a minimum of two CIPs, one for the request and one for the response. Since we adopted a request/response metaphor for our Web services generic scenario we get the benefit of two different reductions: first, the cost per CIP goes down just over 20%; second, we eliminate one of the CIPs. This is a substantial cost savings. Integration messaging that used to cost us $46,000 to implement now costs only $21,000.

This is a real savings and a strong business reason to look into Web services. The model below assumes that you have multiple ways of moving data that you will consolidate; that you can collapse five CIPs into three per; and is for the build and test phases of your SDLC only. The model becomes more accurate with more CIPs.

```
Cost Savings = Current Build Costs –
Improved WS Integration Costs
Current Build Costs = Cost Per CIP * #
of CIP
Improved WS Integration Costs = (Cost
Per CIP * .8) * (# of CIP * .6)
```

For example, assume that your sales executives receive awards and recognition based upon their sales performance. This sounds obvious and straightforward. In reality, the system they use to sell your product, the valuation of the quality of the sale, and the criteria for the award are three different systems. For the awards and recognition system to produce output it takes six CIP’s, which are:

1. Ask for Sales Executive’s information
2. Receive Sales Executive’s information
3. Ask for inventory information
4. Receive inventory information
5. Ask for sales to be evaluated
6. Receive sales to be evaluated

I have adopted the request/response style of using Web services for integration – there are technology reasons why this needed to be the model used. In this example, with the request/response method four interesting things happen. First, I have a tangible build cost reduction for each CIP. Second, I have a tangible elimination of three CIPs. Third, I have an elusive benefit of reusability, meaning that if I ever need this information for a different purpose or system I can completely reuse the Web service. Fourth, if I am adding to an existing Web service’s infrastructure no additional support costs accrue.

Given all of the tangible benefits, my cost per CIP goes down, and the number of CIPs is cut in half. If you assume that you might have hundred’s of scenarios like this one you can be talking about millions of dollars of savings, which should justify the costs of moving to Web services and an SOA.

Another obvious place to look for tangible cost efficiencies is in product maintenance fees. This article isn’t about preaching one Web services platform over another, although I will tell you that the one I selected doesn’t carry any additional maintenance costs than were being paid already. This means that my cost per CIP can go down just by changing over to a Web service and SOA platform. Eliminating a couple of products from the infrastructure is always a good goal when you have hundreds.

Have a Plan for All of Your Systems

By using the CIP index you should be able to cost-justify a project to implement an SOA using Web services and include your legacy systems. One of the first design challenges is the fact that the industry average shows that about 80% of the data and processing you need to provide to the enterprise as a service is still housed in the oldest and least SOA-friendly systems. The goal is still not to write too much new code in these systems because the trend is for organizations to focus on other technologies going forward. I wanted an example that would work for all types of systems.

What I landed on was a bit of a hybrid idea between a pure SOA and the older hub-and-spoke messaging model. Basically, the idea is to have a central hub where the majority of the Web services for legacy systems can be deployed. Other benefits come from this, such as a single point of audit and using SSL, a very graceful transport solution that can be used to meet all of the broader security compliance regulations and legislation. I call this hub the Network Enterprise XML Universal Service, or NEXUS.

I recommend adopting the W3C.Org as the standards body and to only write code for the NEXUS that is W3C compliant. You can use this hub to service intra- and extra-systems needs.

One of the down sides of a hub is that it can be a single point of failure. To address this I recommend a redundant, clustered-server environment that also serves as a load-balance platform for performance, so nothing goes to waste.

Have a Data Strategy

This last part, a data strategy, was a lesson learned with some pain. Decisions need to be made right up front about what your goals are. Do you want to have a system of record for every data element? Do you want the data to be accessible locally to each system that uses it? Does some data need to be more secure, or have audit trails? The questions go on and on. A nifty little way to start to talk about the decisions involves taking a high-level view and categorizing your goals.

	Data Everywhere	Pre-Sync
Strategy #1	Yes	No
Strategy #2	Yes	Yes
Strategy #3	No	No
Strategy #4	No	Yes

TABLE 1 **Decision Strategies**

The model I came up with has two elements that decisions need to be made on, for a total of four possible strategies (see Table 1):

- ***Data Everywhere.*** This decision point established whether you want data to be locally available to each application. Data everywhere would push your SOA into a Transaction Of Record state, meaning that as data changes, or is needed, integration services need to keep the enterprise in sync. The opposite of data everywhere is systems of record, which would push your integration services to go get data as needed from the systems of record.
- ***Pre-Sync.*** This decision point establishes whether you want to scrub and align your data before turning on integration services, or if you want the business rules created around the integration services to synchronize the data as it is accessed.

Looking Forward

Web services have lowered my integration development costs and positioned my architecture around agility. There is no question about it. With Web services, I can start to see the promise of technology that seemed so prevalent just five years ago. I am excited about a number of future plans and ideas I have for optimizing our systems on this platform and look forward to evolving uses of this unique technology. ⓔ

About the Author

David Lady is the senior director of enterprise architecture and planning for a fortune 100 company. He is an industry-recognized expert on systems integration and IT initiative cost forecasting. His current crusade is to help companies understand the true enabling value of system integration and a solid service-oriented architecture vision.

■■■ dlady@tampabay.rr.com

Web Services Management – Plotting a Course for Success

Slow adoption requires careful planning

Web services are no longer a collection of buzzwords in the field of e-commerce. Instead, Web services technology enables companies to effectively integrate applications from disparate platforms and partners into a composite application built on business processes.

WRITTEN BY
ROBERT ANDERSON

In today's early stages of deployment, many companies across a variety of industries are experimenting with Web services. However, we are still not at the point where Web services are the standard for doing business with customers, suppliers, and partners. As a result, most companies won't need a complete Web services management framework until business-critical services become more pervasive – through the end of 2004 and into 2005. This article examines the process of implementing both early-stage and advanced Web services management.

Among the techno-savvy in today's corporations, Web services are being touted as the next big driver for the technology business. Arming companies with the ability to do business securely outside of the corporate firewall is an important step on the road to cutting costs and improving productivity. However, while Web services are a step forward, technology buyers are still left with important unanswered questions. Web services management companies competently addressing security, performance, and reliable messaging of Web services will be best positioned to answer these questions and capture the early investment in management.

As companies begin to re-architect their applications around a service-oriented architecture (SOA) using Web services as the process backbone of their enterprise, it is clear that software to ease this transition should play an integral role in their overall management strategy. The early introduction and effective use of Web services management can help ensure companies reap the simplification and cost benefits of Web services and meet broader business goals.

What Is Web Services Management (WSM)?

Web services management stands apart from the traditional management of most systems, networks, and applications. While traditional network and systems management (NSM) tools typically examine managed components without taking part in or directing their actual production operation, WSM software may take a much more active role in assuring the production operation of business services. This active management is essential when you consider that Web services crosses the boundaries of so many different systems, and networks and applications use open and potentially immature standards.

The componentized and dynamic nature of Web services introduces new considerations for management that are not prevalent in traditional NSM applications. The network of services in a composite application can change frequently, and mature management solutions must be able to answer questions like who's using the service and how often. Companies cannot accurately measure the liability of failed services unless they can answer these questions. These problems dictate that Web services management software must deliver a service-based view instead of the silo-based views provided with traditional NSM software.

The many different components that make up the composite applications of Web services also mean that the best Web services management software will not be limited to services implemented by one vendor or platform, but will instead manage any services adhering to open standards. The complexities of SOA suggest that management requires investing in a company with established experience in enterprise management. Traditional NSM solutions are concerned with making sure individual application components are available and responsive, but Web services solutions must tackle more complex concerns, including security, provisioning, orchestration, load-balancing, and change control. With Web services, a failed service at one provider does not necessarily mean the production business application is broken, adding complexity to the management task (see Table 1).

Where to Start

Most companies won't begin seeking a complete Web services management framework until close to the end of 2004. However, in the meantime they need to look for software to help them during the development phases, including tools, for example, that will help them check for proper WSDL descriptions of their services, perform load testing, or otherwise help orchestrate services. Many of these early needs can be met by the Web application server platform vendors but there are other providers as well. The following are some of the solutions available today.

Web Services Startups

These companies are the ones creating the pioneering advanced Web services management solutions. They offer the most granular control of Web services and secure important gaps in standards like security, but at the same time usually require that services be architected around the management software, which can be difficult to decouple if problems arise with the vendor or its software. They also require significant effort and cost to deploy or redeploy. Some of the key independent Web services vendors include AmberPoint, Actional, Blue Titan, and Infravio.

Larger, more established enterprise systems management vendors are partnering or acquiring many of these players at a rapid clip, and it's questionable how many of these startup companies will survive intact as Web services adoption rates increase through the next 18 months. Hewlett-Packard acquired Talking Blocks in an effort to strengthen its adaptive management approach to linking IT to business services. Computer Associates appears to have decided to more directly target the basic problems of Web services management by acquiring Adjoin and releasing an extension to its flagship Unicenter product line, CA Unicenter Web Services Distributed Management.

Web Services Platform Vendors

Microsoft, IBM, BEA Systems, and Sun Microsystems are racing to add basic Web services management and development aids to their offerings, usually at no additional cost, in order to compel wider adoption of their platforms. Where it makes sense to them, these vendors are also seeking integration with some of the start-ups.

For these vendors, their biggest strength is also their weakness – their solution is tightly integrated with their application server and development environment. Their solutions meet very specific needs based on the platform suite but focus only on one vendor's solutions. Additionally, they currently do not offer solutions to many of the important challenges outlined in Table 1.

Traditional NSM vendors

Companies tend not to need new infrastructure to implement Web services, and, as such, traditional management vendors, who already manage this infrastructure, are well-positioned to add Web services capabilities to their existing offerings. Any additional capabilities can be integrated immediately into the enterprise nerve center, allowing Web services to quickly reach production status. However, these vendors are being pushed to commit to standards that are still subject to change and that may or may not be adopted, slowing the overall maturity of Web services management software.

OASIS (www.oasis-open.org) has a technical committee (TC) dedicated to proposing a standard for Web services management, the OASIS WSDM TC. This is the only body proposing a standard for the NSM vendors to use but it is months away from ratifying an initial proposal. It will take considerable time for the TC to address the complex challenges outlined above.

Recommendations

Looking to the future, companies should focus on the early stages of Web services management and begin to move management of these services into the enterprise nerve center.

Companies should also take advantage of the various tools available within their Web services platforms. Much of today's Web services development is taking place within the J2EE application servers and, as such, the availability and optimum performance of J2EE platforms – like IBM WebSphere or BEA WebLogic – is of the utmost importance. Just as important is the overall availability and performance of Web services and the applications that are executed on these platforms.

New management capabilities are being incorporated into these application platforms as their Web services capabilities mature, and enterprises should look specifically for those with in-depth J2EE management facilities. These should expose more execution details about applications, enabling IT operations personnel and application developers to identify and resolve problems that occur within Web services applications.

Conclusion

Except for truly leading-edge implementers, most companies using Web services today are focusing on integration of heterogeneous applications. They are not ready for enterprise-wide adoption of Web services, but do need to get a handle on the availability and performance of their current services. In some cases, their NSM vendors are fulfilling that need today. Meanwhile, early adopters using Web services to create composite applications must continue to take an opportunistic approach to Web services management – making investments in technologies that they know may have to be replaced as standards mature. ⓔ

About the Author

Robert Anderson is a product manager for PATROL Applications Management at BMC Software, Inc. He focuses on market strategy and the development of Internet and Web services management products.

■■■ rma@bmc.com

Provisioning, orchestration and change control	One of the greatest challenges of moving Web services into production is answering the question of which services are in use and by whom, and understanding the liability associated with those services. By implementing a management tool that provisions services under published guidelines, companies can understand what is being used and make certain that changes to the services are seen by all service endpoints.
Security management	Security management includes providing the single sign-on capabilities that are crucial to service interoperability and integrity. WSM vendors will not try to reinvent Web security, but will instead depend on integration with popular security solutions, such as those available from Netegrity and Oblix, to provide these capabilities for Web services.
Web service translation and transformation	This capability allows companies to add business logic to a service in-stream depending on the class of consumer or other business rules, giving the greatest flexibility in service implementation.
Service routing, fail-over, and load-balancing	The reasons for service fail-over and load-balancing are obvious, especially when you consider the sheer number of interconnecting endpoints that can exist in a true service-oriented architecture. A strong solution around service routing allows companies to move service endpoints around as business needs dictate.
Service monitoring, logging and reporting (SLA)	This is the aspect of Web services management most aligned with traditional NSM problems. What makes this problem unique for Web services is understanding when a failed service leads to a failed business process. Also part of this are more advanced initiatives such as business activity monitoring (BAM).

TABLE 1 | **WSM Challenges**

Web Services Converge with Data Integration

New ways to unlock the power of data

■ As data integration tools produce and consume information, a service-oriented architecture presents opportunities for information exchange, data-driven process automation, and business agility.

WRITTEN BY JAMES MARKARIAN

Web services proponents like to rhapsodize about a somewhat quixotic world in which Web services orchestrates disparate components of the enterprise in a seamless, real-time symphony. There's genuine promise in that vision. But as any IT professional worth his or her salt knows, the devil is in the details. When you begin to look under the Web services hood, it's clear that certain crucial distinctions tend to get lost amid the froth.

In this article, we'll drill down into one of the most compelling propositions offered by Web services – as an enabler of data-level integration. Then we'll explore how Web services plays with data integration to better support real-time information access delivery to the business users who need it.

Leveraging the Proven Value of Data Integration Technologies

Data integration helps organizations leverage their data across a variety of integration initiatives for improved operational efficiency and business performance. Whether a project involves data warehousing, data migration, building master data hubs, improving data synchronization, or implementing business activity monitoring projects, it requires reconciliation and integration of data, often on an enterprise scale, to provide a single view of customers, households, and suppliers. For more than 10 years, data integration has delivered some of the highest ROI of any technology set.

So now Web services comes along, and prompts two questions:

- What does Web services mean for data integration and business intelligence?
- What do data integration and business intelligence mean for Web services?

At first glance, they may sound like the same question. They're not. They're two distinct yet related issues that organizations deploying Web services and data integration technology – that is, virtually any mid-size to large enterprise – need to carefully assess.

Data-Level Integration with Web Services

Though data integration has delivered enormous value, many systems built piecemeal throughout the 1990s operate in relative isolation via proprietary APIs and hard-coded connectivity. Web services offers a way to liberate both the integration and analytic aspects of data warehousing, expose them as Web services, and embed this functionality into enterprise applications.

In the first iteration, most flavors of Web services do not address the issue of data-level compatibility among disparate systems. When we talk about Web services, it's usually in terms of connectivity and interoperability among loosely coupled, disparate applications via such standard protocols as XML, SOAP messaging, WSDL to describe services, and UDDI as a services registry.

Customers who plunge headfirst into a Web services integration initiative may soon find themselves confronted by the old apples-and-oranges dilemma – missing data, incompatible data (for example, different customer IDs between SAP and Oracle application systems that require the use of lookups and lookup tables), format discrepancies, unit of measure differences, and invalid data (data that is valid in one system but rejected by another). Data integration facilitates Web service integration by enabling application-to-application integration at the data level, thus providing solutions to all these problems via data mapping and transformation capabilities, integrated data cleansing, and automatic lookup techniques. A Web service call can invoke a data integration engine to return requested data (as XML) that is transformed from its native format and cleansed for consistency. If data is changed in an operational application, the data integration engine can function as a Web service that propagates data-level changes across target applications.

For example, a wholesale customer phones an inbound call center to order a large quantity of your company's product. With a data integration engine in the mix, the sales agent can compare the quantity of the customer order to historical averages and the amount in inventory. If the quantity is significantly different, the data integration engine can effect changes in the inventory system or notify the head of procurement about the situation.

With a Web services call to a shipping system, the data integration engine can capture that data, transform it, and propagate it to, say, an SAP financial application and an Oracle customer transaction history record. In turn, the data integration engine can extract, cleanse, transform, and aggregate data from those applications and load it into a data warehouse. Depending on predefined business rules, the system may push out an alert of a large order to the BI dashboard of a sales manager. These sequences are accomplished in minutes or less – in effect, in real time.

Data Integration Evolution: From Tool to Web Services Platform

This is an evolutionary step in the data integration market. Until recently, data integration tended to mean extraction, transformation, and loading, or ETL. It was also associated exclusively with data warehousing and executed in an isolated batch fashion. Gradually, the acronym is giving way to the more comprehensive term *data integration*. Data integration platforms are engineered to execute five key elements among multiple sources and multiple targets – movement, transformation, aggregation, cleansing, and profiling. They are also now engineered to operate in real time and feature high-performance engines that fully exploit SMP and grid architectures to offer fast performance on large data volumes.

> “Data integration helps organizations leverage their data across a variety of integration initiatives”

A key enabler for data integration is Web services, and the connectivity and interoperability its standards provide to loosely coupled applications. In the past year, leading data integration vendors have incorporated support for XML, SOAP, WSDL, and UDDI in their products, and delivered open Web services APIs within software development kits. These help support:

- Real-time updates across disparate applications and databases
- A stand-alone data transformation service
- Bidirectional data-level integration
- Ease of development, implementation, and maintenance of a data integration service-oriented architecture

But support for Web services standards in a data integration platform is only the start. Companies that look to couple data integration platforms with Web services should not take for granted that their vendor supplies the functionality necessary for fully optimized systems.

Prospective buyers should look for centralized metadata management support, configuration management, robust security, broad data connectivity including mainframe and AS400, and change data capture.

Metadata Management

Metadata – or data about data – was once something of an afterthought in the larger world of data management. But as systems and data have grown in volume and complexity, more companies are implementing metadata systems for visibility into integration and business intelligence processes, allowing for significant benefits in terms of reuse, productivity improvements, and reduced coordination costs.

Similarly, a metadata repository can serve as a secret source for tracking Web services publication and invocation mechanisms executed via WSDL, UDDI, and SOAP. Metadata provides a means to answer such questions as, “Where did this Web service originate? What’s the source of this data? What does it do, for whom? What formulas are applied to get it here?” It’s a powerful and incisive tool for keeping straight what would otherwise become a spaghetti-like snarl of services.

In a practical sense, suppose you expose a currency conversion program as a Web service. Millions of dollars may be at stake – but what is your assurance that the conversion rate is accurate and the Web service valid? Metadata enables you to monitor and validate the system. Similarly, metadata provides lineage and versioning to data exposed as a Web service that can be crucial in establishing data accountability and monitoring compliance with such mandates as Sarbanes-Oxley and HIPAA.

Best of all, it’s not necessary to redevelop Web service–based access to metadata systems. A metadata-driven data integration platform will automatically publish service-related metadata with negligible cost or risk.

Open Interfaces and APIs

A data integration platform should feature broad support for open standards (such as COM, XMI, and LDAP) and APIs to plug into applications from any number of vendors. This support (along with backwards compatibility) is also important in liberating legacy and mainframe applications for Web services.

Given that Web services is frequently about interoperability, buyer organizations will be well served by a data integration vendor that boasts a long history of support for standards and APIs. The last thing you want is a data integration platform that’s going to require hard-coding around proprietary interfaces. That can negate any productivity gains you achieve through Web services.

Robust Security

SOAP over an SSL transport is necessary, but it’s not enough to ensure data protection, particularly in multi-hop Web services systems. For security that’s suitable for the majority of enterprise applications, look for a data integration platform that supports authentication and authorization as well as in-transit encryption. This is an emerging area and involves acceptance and validation of authentication tokens, as well as encryption, decryption, signing and verification of SOAP messages.

The best data integration platforms will accommodate LDAP, or Lightweight Data Access Protocol, and support leading LDAP servers, such as Java System Identity Server from Sun Microsystems, or Active Directory from Microsoft. LDAP will govern which individuals and applications have access to which data, and where.

Change Data Capture

This technology is crucial to deploying a data integration platform in a service-oriented architecture. Change data capture is a technique by which only data that has changed since an application's last call is fetched and propagated to the requestor.

As such, it can reduce by orders of magnitude the impact on both a source application and the network in retrieving requested data. That's a critical consideration when architecting a real-time Web services infrastructure to support an initiative such as business activity monitoring (BAM).

Building Blocks for Visibility

Business intelligence (BI) has proven crucial in enabling organizations to analyze data on customers, sales and marketing, finance, supply chain, and other operational areas. Eager to build on that success, the demand from companies – and the focus of many BI vendors – is increasingly on systems that knit those various elements into a broad analytic fabric across the enterprise.

> "Until recently, data integration tended to mean extraction, transformation, and loading"

BI platforms have evolved incrementally. Over the years, companies have deployed tactical BI systems to attack key subject areas. From an architectural standpoint, the data warehousing infrastructure is closely coupled with the operational systems from which it sources data. However, it depends on fairly rigid processes for data access, collection, and analysis.

Web services offers an opportunity for greater fluidity between operational and analytical systems. Along with accepted Web services standards, a subset of protocols aimed at analytics and data management, such as XML for analysis, CWM, and the Java OLAP API, pave the way for developers to link operations and analytics through cost-effective component reuse.

In addition, Web services' ease of development and deployment gives IT organizations new mechanisms to rapidly extend analytics to meet short-term tactical goals. From a strategic standpoint, this dovetails with objectives for faster and more precise insights into business performance:

- Real-time monitoring of business conditions
- Metrics-driven alerts with analytic drill-through
- Better visibility into dynamics among interdependent processes

These goals are encapsulated in BAM. A key premise of BAM is its real-time monitoring of changes in business data and notification to decision makers. A service-oriented architecture using trigger-driven business rules neatly meets this goal, as in the following example.

A product manager would want to know immediately that a new product release has triggered a flood of complaint calls to a call center in order to identify the root cause and take corrective action. With many implementations of present-day technology, it may take a week or more before a pattern is discerned, and more time before corrective actions may be executed.

With the Web services approach, developers can readily couple the transactional data associated with the complaints to a business rule threshold defined in a BI system. The BI system's business rules function as an intelligent agent that continuously "listens" for anomalies and delivers alerts to a manager's desktop. This BI Web service may be described in WSDL, registered in a UDDI library, and invoked by the requesting application via SOAP messaging, with data exchanged in XML format.

In a related area, because a service-oriented architecture separates presentation and application layers, Web services enable developers to embed BI functionality within an operational application. For instance, a call center agent working from a Siebel call center interface can summon information on customer affinity program status and predefined cross-sell and up-sell opportunities stored in a BI system.

Dividends from a Three-Pronged Convergence

To date, much of the attention paid to Web services has been as an enabler of enhanced enterprise interoperability, and more efficient and cost effective deployment of services. Naturally, CIOs, IT managers, and developers are focused on building simple Web service request/response systems. At the same time, they are assessing the long-term viability of Web services standards, as well as the competing J2EE and .NET platforms.

Web services itself, though, is not an "if" – it's a when. Already it's evident that Web services will evolve in much the same way data management software evolved throughout the 1980s and '90s. First, companies deployed relational databases and ERP applications to crunch numbers, record transactions, and track customers. Once those systems were up and running, organizations turned their attention to data integration systems to analyze and understand the raw data.

Though Web services are bound to be deployed in phases, it's not too early to examine how to build in data integration capabilities at the ground floor. Ultimately, however, the open architecture of Web services will enable developers to plug and play data integration functionality as business needs and IT strategy dictate.

About the Author

James Markarian is the CTO of Informatica, where he leads their product strategy, defining the key technologies and themes that are instrumental for Informatica's industry-leading data integration platform. Prior to joining Informatica, James spent 10 years at Oracle Corporation, where he held a variety of positions including senior architect of the Oracle Tools division and development manager of the Oracle Forms product.

■■■ jmarkarian@informatica.com

INFORMATION
STORAGE+
SECURITY
JOURNAL
ISSJ
PREVIEWING
MAY 11, 2004
NETWORLD+
INTEROP
IN LAS VEGAS
Microsoft
RSA Security
SecurID for
Windows
SAN
NAS
Emerging technology trends
and market maneuvers
PLUS
Secure Service-Oriented B2B Architectures
Rebuilding the Foundation of Enterprise Security
Disaster Recovery with an iSCSI SAN
IP STORAGE • SAN NAS • DISASTER RECOVERY • ENTERPRISE SECURITY • INFOSECURITY

SOA Came to Boston at EDGE (East) 2004

edge 2004
EAST
Development Technologies Exchan
Conference February 24–26, 2004 • Exposition February 24–25, 2004 • Hynes Convention
Microsoft
intel

Perspectives on the February 24–26 Conference & Expo

■ (Boston) – The technical programs of technology conferences make very useful weather vanes for the state of the union in the technology space, and the i-technology devised by the advisory board for the seventh successive "Edge" conference was no exception.

After six shows under the "Web Services Edge" moniker, the decision was made early on that Web services has become so much of a given now – within the firewall and, increasingly, beyond it – that it made sense to de-emphasize this time round the newness of loosely coupled, distributed computing and concentrate instead on the chance that this show offered all comers the opportunity to do a side-by-side "compare and contrast" of leading Internet development technologies – Java, XML, .NET, Web services, and the new MX technologies from Macromedia.

Within a few weeks of announcing the Call for Papers one clear trend emerged – that service-oriented architecture (SOA) was going to be the next shoe to drop. Never has a single notion recurred so evenly throughout every session and all three keynotes.

The opening keynote, from Orbitz's CTO Chris Hjelms, featured SOA (see sidebar), as did the two other keynotes, one by Macromedia's General Manager David Mendels and the other by IBM's Robert S. Sutor, director of marketing for IBM's WebSphere Foundation Software as well as its Web services and SOA efforts.

What follows are a selection of perspectives and reports on the show, first-hand reports that give the flavor of what was another very rich technical program, complemented by the usual full two-day Expo that the "Edge" series is justifiably well-known for.

Keynote's Underlying Message: "Hire Bright Programmers"

"The CTO of Orbitz...gave a very interesting key note. Ostensibly, it touted the benefits of a service-oriented architecture (one of the biggest buzzwords of the day), but, as far as this listener could tell, a lot of what it added up to was: hire the brightest programmer you can find; set up a process which keeps out of their way; allow them to use open source tools if they like them. But, especially, hire bright programmers.

Sadly, this never seems to be management's answer to anything. Chris Hjelm…was very clear that their technology gives them a competitive advantage. If I had to sum it up, I would say that their key differentiators… are: (a) a very sophisticated search algorithm, developed by ITA Software (www.itasoftware.com) for Orbitz; and (b) a very short time-to-market for new features, made possible by small teams working with as little technical and administrative overhead as possible."

Dan Milstein, writing at Java.net: (www.java.net)

•••••••••••••••••••••••••••••••

I am here at the conference, getting ready to go into the session "Building a Rich EIS Dashboard with Macromedia Flex" given by Dave Meeker with WHITTMANHART. After this session, David Mendels (Macromedia SVP, Macromedia Tools and Platforms) will be giving the keynote titled "SOA + RIA = ROI".

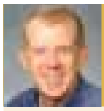
My first impressions of the conference are extremely positive. The attendees seem to be geared more towards the business side of the house and project leads. Much different than what I usually see at many of the Flash conferences I have attended, which seem to cater more towards those who are in the "trenches." The staff has been extremely helpful, and the overall show seems very organized and well thought out.

Daniel Dura, writing at www.DanielDura.com

•••••••••••••••••••••••••••••••

I attended 14 sessions, plus the Expo and Macromedia's Flex reception and walked away with these overall impressions:

1. I'm less concerned now with which technology is the best for developing Web applications or services (there's room for all of them, so long as they stick to standards), and more interested in the best way to intelligently connect them and orchestrate what they're doing throughout an enterprise. The session on "Service-Oriented Integration: Making the Right Choices to Support the Next



Keynote Chris Hjelm, CTO, Orbitz

Without SOA, says Orbitz's chief technology officer, his company – founded in 2000 and successfully IPO'd since then – wouldn't be able to leverage its main competititive advantage: speed to market.

Hjelm unpacked this broad-stroke observation for a packed keynote hall on Day One of EDGE 2004. Making this as much a case study as anything, Hjelm first established the parameters of the Orbitz operations: 800+ production servers (usually commodity PCs), 3.9 million lines of code on its Web site, and 110,000 lines of code being added or deleted on a weekly basis.

"Getting your architecture right enables you to do more things faster," Hjelm observed: in Orbitz's case, that above all means of course faster searches. Orbitz's consumer-friendly site, he explained, masks a complex infrastructure.

Capable of searching over 2 billion flight/fare combinations, including feature searches unique to Orbitz such as Flex Search and DealDetector, the Orbitz system validates the effectiveness of SOA - "Traditional search on old mainframes would be too expensive," Hjelm noted. Yet within just four years the Orbitz system has proven so effective, so growthful, and so user-friendly that Orbitz has in that short time captured some 17% of the entire online travel market (versus Expedia's 40% and Travelocity's 20%).

In the car space, Hjelm noted, Orbitz has captured a much larger market share. And on the messaging side, Orbitz sent out over 775,000 wireless messages in December 2003 alone.

Leveraging SOA, the company is able to deploy new SL implementations dynamically with zero impact to existing systems. "When we add a new carrier," Hjelm says, "that's done with zero impact on existing systems plus all the systems could care less what airlines we're adding in. The folks doing the 'plumbing' can work separately from those doing the actual customer-facing work."

Orbitz has had to build a fairly sophisticated customer services tool, he added.

"What are the enablers that allowed Orbitz to do what it does?" he asked rhetorically. "For a start, we built it ourselves. Re-architecting existing systems is very hard to do. When you make hardware upgrades, it's a lot easier just to add a commodity PC."

Using JINI

JINI handles all of the interfaces between Orbitz and third-party services, he explained, "managing down the complexity that would otherwise make it very difficult."

Orbitz has internally hosted J2EE inventory systems and, with the exception of one or two database servers, the entire complex runs on Linux "and it works extremely well," as Hjelm says. The advantage of deploying on Linux, he added, is that it is easily scriptable, which is why it is the building block of many commercial server appliances.

Recounting briefly the history of Orbitz's service-oriented architecture, Hjelm said that Java was the first big decision, and JINI the second. "So when SOA became popular, Orbitz had already found it.

"The JINI distributed computing framework focuses on interfaces and capabilities not implementation and location," Hjelm said.

Using EJBs gave transactional capabilities to the system and provided a robust services layer – and they are colocated on a single VM to reduce latency.

"Redundancy, stateless capabilities, it's all just 'here' with this architecture," Hjelm says.

"This all allows the client code to focus on the capabilities, not the implementation. Our developers don't worry about that side of things.

"If you were to take the average person and explain GDS to them, their head would hurt," he added, "whereas we can get developers up to speed on Orbitz fast. We add a new machine, bring it up into the network, and JINI recognizes it and starts to draw on it. It works.

"So the growth in our code base isn't in the services layer, it's in the application layer," he pointed out. That is the key to Orbitz's success, and that in turn is a function of its architectural choices.

What Next?

"From an industry standpoint nothing works better than SOA," Hjelm said. "At Orbitz we aspire to autonomic computing – we're still 3–4 months away from that."

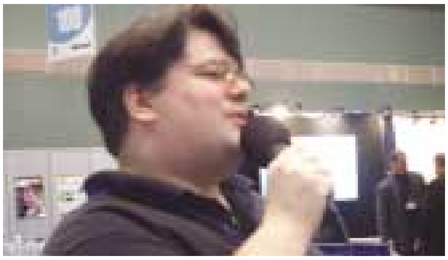
Enhancements going forward will include a guaranteed model for delivery, and building the redundancy and scalability of the database interactions.

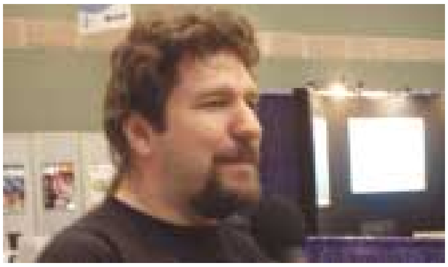
New features on the business side are being implemented constantly, Hjelm said. "Web services allows entry into new markets and facilitates new business models, leverages existing SOA, allows access for non-Java clients and systems, and furthers the goal of flexibility and robustness.

"Web services on top of an SOA is pretty much worthless unless you get SOA right," he observed. "You have to get the SOA right first. Then, like Amazon, we'll open up our transaction engine to anyone who wants to innovate against it."

With 90 or so software developers, the company does all of its hiring through referrals. "It's an open source culture; it's rare we have a project team bigger than 4 or 5 people (usually 2 or 3)."

In his final comments, Hjelm said. "SOA is a key enabler but...the key competitive advantage for Orbitz is speed."





Generation of Integration," by Dave Chappell of Sonic Software introduced me to the Enterprise Service Bus (ESB) and what appears to be a great way to evolve an "accidental architecture" into a more "service-oriented architecture."

2. Macromedia's Flex technology looked pretty exciting. I think they've opened up Flash in a way that the developer community will find appealing. This stands to enrich the whole user experience and make the Internet a vastly more appealing place to visit. I have a sense there are a lot of Web applications out there that will be getting a facelift. :-)
3. Orbitz appears to be one example of an organization that spent time thinking about architecture up front and has reaped rewards for that effort. Without going into detail on reasons why, just let me say that this was refreshing and reassuring to see.
4. "An Experienced Programmer's Guide to C# and the .NET Platform". Wow. Michael Stiefel knows C# and made it look very interesting. He raffled off two copies of his book at the end of his session and I was so bummed to not win one that I went back to the Expo to buy it from the publisher. But the Expo had closed, so I looked for it at Barnes & Noble in the plaza. They didn't have it...yet. So currently it's on my list of books to buy (or convince my employer to buy for me).

Steve N. Bowers, Sr. Systems Analyst
ITS Academic Technologies
The University Of Iowa

SYS-CON Radio

As leading members of the computing industry were speaking at EDGE 2004 East Conference and Expo in Boston, SYS-CON Radio was naturally there, as usual, to talk with the movers and shakers currently shaping the next phase of *i*-technology growth and development.

The resulting interviews, with people like CA's Sam Greenblatt, SVP and chief architect for the Linux Technology Group at Computer Associates; IONA's CTO Eric Newcomer; Intel's Senior Architect Alan Boucher; Ascential's Michael Curry; Nexaweb's CTO Coach Wei; and many others, are all still archived and freely available at the conference site: http://sys-con.com/edge 2004.

A LIMITED TIME SAVINGS OFFER FROM SYS-CON MEDIA

SUBSCRIBE TODAY TO MULTIPLE MAGAZINES

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

RECEIVE YOUR DIGITAL EDITION ACCESS CODE INSTANTLY WITH YOUR PAID SUBSCRIPTIONS

CALL TODAY! 888-303-5282

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

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

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

□ **MX Developer's Journal**

U.S. - Two Years (24) Cover: $143	You Pay: $49.99 /	Save: $167 + FREE $198 CD
U.S. - One Year (12) Cover: $72	You Pay: $29.99 /	Save: $60
Can/Mex - Two Years (24) $168	You Pay: $79.99 /	Save: $137 + FREE $198 CD
Can/Mex - One Year (12) $84	You Pay: $49.99 /	Save: $40
Int'l - Two Years (24) $216	You Pay: $89.99 /	Save: $127 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $59.99 /	Save: $30
Digital Edition - One Year (12)	You Pay: $19.99	

□ **Linux World Magazine**

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

□ **Java Developer's Journal**

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

□ **Web Services Journal**

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

□ **.NET Developer's Journal**

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

□ **XML-Journal**

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

□ **WebLogic Developer's Journal**

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

□ **ColdFusion Developer's Journal**

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

□ **Wireless Business & Technology**

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

□ **WebSphere Developer's Journal**

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

□ **PowerBuilder Developer's Journal**

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

*WHILE SUPPILES LAST. OFFER SUBJECT TO CHANGE WITHOUT NOTICE

3-Pack

Pick any 3 of our magazines and save up to $275.00 Pay only $175 for a 1 year subscription plus a FREE CD

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

6-Pack

Pick any 6 of our magazines and save up to $350.00 Pay only $395 for a 1 year subscription plus 2 FREE CDs

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

9-Pack

Pick 9 of our magazines and save up to $400.00 Pay only $495 for a 1 year subscription plus 3 FREE CDs

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

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

SYS-CON MEDIA

Beyond Point to Point

WS-Eventing and content-based routing for integrating multiple applications via Web services

Web services have emerged as an excellent method of integrating pairs of applications. Free and cheap Web services development tools from many different vendors make it easy to expose one application's capabilities to other applications that wish to invoke them. But, given recent trends and innovations in Web service standards for more complex integrations of multiple applications from many parties, integrating applications two-by-two with tightly coupled simple Web services may not be the best approach.

WRITTEN BY
ADAM BLUM

In this article, I'll take a look at a reasonably complex integration scenario performed by integrating with many different applications' Web services. I'll show you how the architecture of the integration is simplified by using XML content–based routing with a publish/subscribe approach. In the remainder of the article we'll assume that we are using a content-based routing service based on the recent WS-Eventing specification from Microsoft, TIBCO, and BEA. The XML document message flow in a WS-Eventing–based approach to the problem is shown. Finally we'll show the code necessary from one of the endpoint applications to generate and process these messages.

One Scenario

A consumer electronics retailer decides to carry a new consumer electronics product. The "product master catalog" might invoke a Web service exposed by the inventory system to create an entry to track the inventory of the product, then call another Web service that performs integration with one supplier of the product to request some product for delivery. A service on the accounts payable system is called to create a new account for the vendor to ensure that the supplier is paid when the invoice is received.

After deciding that they want to carry the product on their direct Web site, the Web site product catalog is updated with the new product's presence, also via a Web service. They then need to invoke Web services to communicate with each of their stores about the impending arrival of the new product, so that the stores can plan stocking and shelf space for it, and to let them know about the initial allocations of the product.

The product marketing group uses the product's category to make Web services calls to the company's cross-marketing software that determines correlated products that are likely to be bought with it. Then they call services on other suppliers to get quotes on those products for consideration of starting the new product stocking process once again.

Figure 1 attempts to show this integration as a diagram. Each edge is a Web service of a system that the "master catalog" application needs to integrate with. Notice that they each have their own names, indicating a different Web service that must be integrated with. There are quite a few independent Web services integration efforts required just around this one business event – the addition of a new product.

A Better Way: XML Content–Based Routing

With the emergence of standards focused on publish-subscribe messaging such as WS-Eventing and WS-Notification, and a coincidental trend toward more coarse-grained, document-style payloads of Web services, I think there is a better way to do this integration and still have the advantages of simplicity, standards support, and ease of development that are the hallmarks of Web services projects.

Publish-subscribe approaches to integration are, of course, nothing new. Before Web services came about, an integration consultant presented with the scenario above might well have architected the integration as publishing the "new product" event out to a messaging bus to a "new product" topic. The other systems that needed to know about new products would subscribe to that topic. Many projects based on software from TIBCO, IBM's MQSeries, and Java Messaging Service implementations from vendors such as Sonic Software and Fiorano Systems were targeted at similar scenarios.

However, there is an opportunity to perform such integration using pure Web services standards and nothing but XML content. In the scenario described here, the master product catalog application generates a NewProduct XML message. The company's accounts payable application and inventory applications are subscribed to all instances of the NewProduct document. However, other subscription rules vary between these applications. There is content in the NewProduct message that indicates that the product will be sold on the company's Web site; hence, the Web site catalog application subscribes only to that content. Other content in the document indicates which stores will carry the product. Each retail store thus subscribes only to the NewProduct documents that are relevant for it.

Suppliers subscribe to NewProduct documents for products that they carry. Note

Now

More

Than

Ever

Before...

JAVA DEVELOPER'S JOURNAL

JDJ

Means

Business

EXCLUSIVE: WHERE IS TECH GOING IN 2004?
JAVA DEVELOPER'S JOURNAL
JDJ
JANUARY 2004
VOLUME 9 ISSUE 1
SPECIAL REPORT:
SCOTT McNEALY
2004 PREDICTIONS
PLUS...
Viewpoint: Deutsch's Fallacies, 10 Years After
Feature: Multiplatform Frames
Data Integration: Which EII Solution Is Right for You?
EDGE 2004 EAST
BOSTON, MA
TAKE A RIDE WITH THE
CEO OF BEA
THE WORLD'S LEADING i-TECHNOLOGY MAGAZINE WWW.JAVADEVELOPERSJOURNAL.COM

Subscribe Now!

$69⁹⁹ 1 YEAR/ 12 ISSUES

www.SYS-CON.com/JDJ

or call

1-888-303-5282

The World's Leading *i*-Technology Magazine Just Got Better! Now with expanded business coverage, *JDJ* is the world's premier independent, vendor-neutral print resource for the ever-expanding international community of IT professionals.

The Most Popular
i-Technology Magazine
in the World !

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

that the decision to use a publish-subscribe paradigm for some parts of the Web service message exchange (such as informing all of these applications about the NewProduct) doesn't imply that everything has to be done that way on a given project or message choreography. For example, suppliers may call back to the master product catalog application by invoking a ProvideQuote Web service operation exposed by that application. Other suppliers of accessory goods can call the ProvideQuote Web service on related products to the original product.

WS-Eventing Message Exchange

WS-Eventing is a specification proposed in January 2004 by Microsoft, TIBCO, and BEA (you can see the spec itself at http://msdn.microsoft.com/webservices/default.aspx?pull=/library/en-us/dnglobspec/html/ws-eventing.asp). It is a general-purpose mechanism for Web services to send messages to register interest in certain events. Those Web services then expect to receive messages about those events. In the terminology of WS-Eventing, "event sinks" create "subscriptions" by sending messages to "event sources." Those event sources then send "notifications," all documents that match the subscription criteria, back to the event sinks.

WS-Eventing is not particularly targeted to content-based routing per se. It can be used to express interest in system events such as servers being down or printers being out of paper. Nevertheless, it has just about all of the messages that we would want for XML content–based routing in a relatively simple specification that makes intelligent use of existing Web services standards. Because of the need for content-based routing for scenarios like the one above, and the historical availability of other non-Web services–based publish-subscribe solutions, WS-Eventing–based, content-based routing systems will probably emerge quickly. In this article I'll show you some code samples against an implementation of WS-Eventing that my company has released for demonstration purposes only.

To make the scenario more concrete I'll show you how a WS-Eventing–based, content-based routing service can make that integration much simpler. Specifically, we'll look at how the publishing master catalog application and various subscribing enterprise applications in the example just described can use WS-Eventing's Subscribe, SubscribeReponse, and notification messages for more efficient and direct integration.

Subscribe Messages

Each of the applications that need to be informed about the addition of new products subscribes to the NewProduct document. For example, the inventory application may be interested in all NewProduct documents. To express that, the application would send the message shown in Listing 1 to the content-based routing service.

This message is sent by the inventory application to the Electronics Retailer content-based router service. The address of the CBR service is http://electronicsretailer.com/cbrservice as expressed in the WS-Addressing To header. Although the WS-Eventing specification implies that the To header is required, SOAP stacks with WS-Addressing support are few and far between. So the WS-Eventing-based CBR service that we use in this code example (an alpha implementation from Systinet) can be connected to via basic SOAP over HTTP and does not require that the wsa:To header be present.

The WS-Addressing ReplyTo header is used to indicate the location for the CBR service to deliver SubscribeResponse messages to. The contents of the WS-Addressing MessageID header sent in the subscription request are placed by the CBR service (or any WS-Eventing–compliant event source) in the subscription response RelatesTo header.

Looking at the elements in the body of the Subscribe request, the NotifyTo element

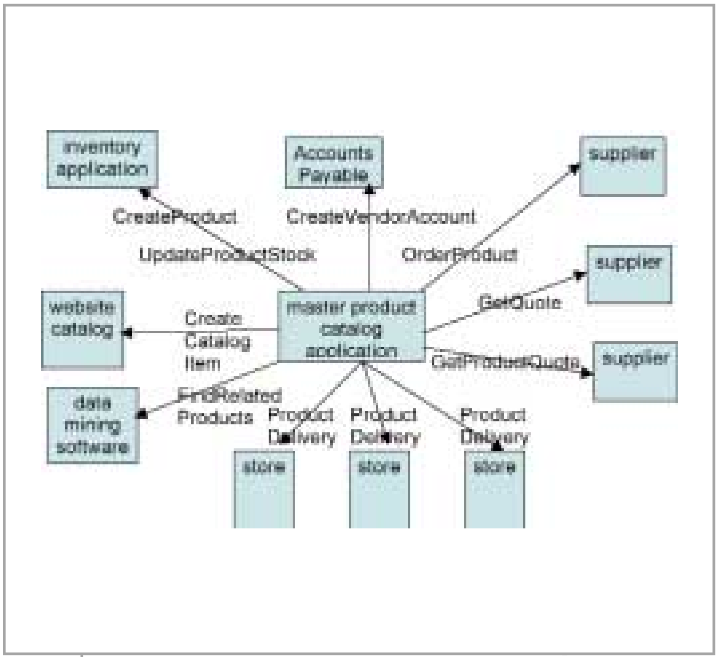


FIGURE 1 New product integration scenario with point-to-point Web services

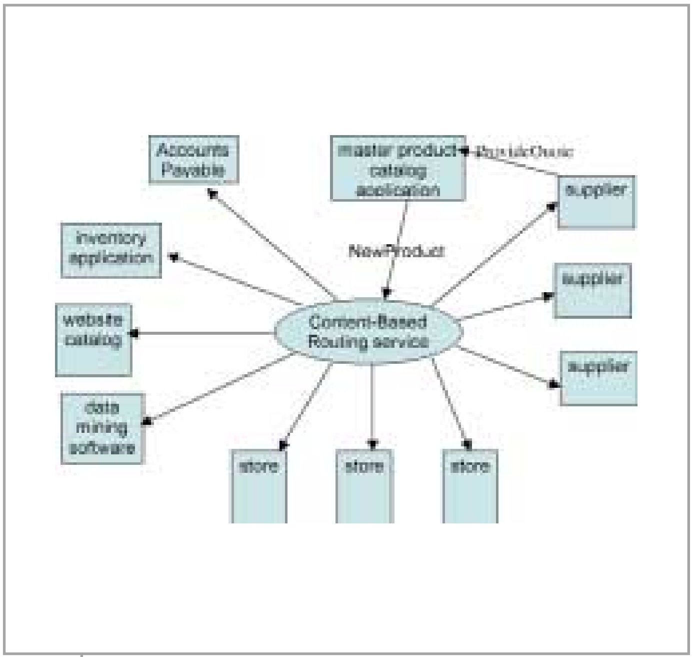


FIGURE 2 Integration via content-based routing

LOOK FOR YOUR FREE...

>Linux >Java >Web Services >.NET >XML >Wireless >Storage >Security

SYS-CON MEDIA

The Premier Resource for Today's Corporate & IT Decision Makers

VOL: 1 ISSUE: 1 SPRING 2004

IT solutions >GUIDE

WWW.SYS-CON.COM/IT

> Delivering Software as Service

> Leveraging Linux/Open Source

> Moving to a Service-Oriented Architecture

> Desktop Software: Migrating from Server to Client

> Using Developer Tools to Drive Cost Out of Software

> Application Integration

> Storage & Security

TECHNOLOGIES YOU NEED NOW!

How to Manage Your Ideas Using Today's *i*-Technologies

Reaching 135,000 Corporate and IT Decision Makers

Coming this **SPRING!**

contains the address where the inventory service wants to receive notifications, or more specifically, the NewProduct documents specified in the Filter element. The address is http://electronicsretailer.com/inventory/newproducts.asp, which presumably will perform the processing necessary on any NewProduct documents received.

> Web services have emerged as an excellent method of integrating pairs of applications

Following the WS-Addressing specification, the contents of ReferenceProperties element will be sent as a header in any notifications that are sent in response to this subscription. In this example, <eri:LocalSubscriptionID>1234</eri:LocalSubscriptionID> will be placed in the headers of any NewProduct documents that are sent to the inventory application. Depending on the application that is handed the document, the identity of the subscription may be useful in the processing of received notifications.

The Filter element allows the subscriber to identify which notifications they want to receive. If not specified, the default dialect for expressing the filter is XPath (www.w3.org/TR/1999/REC-xpath-19991 116). The dialect for the Filter can also be specified as an attribute to another one that the event source supports (e.g., XQuery perhaps). The example above uses the following XPath expression:

```
/s12:Envelope/s12:Body/er:NewProduct
```

to specify that the subscriber wants to receive all NewProduct documents. The expressions can also be used by subscribers to restrict message delivery to only those that match certain criteria on the content of the document. For example the expression below would result in the subscriber receiving all NewProduct documents that have a Price subelement with a value greater than 100.

```
/s12:Envelope/s12:Body/er:NewProduct/er:Pri
ce > 100
```

SubscribeResponse Messages

The content-based routing service will send SubscribeResponse messages to each subscribing application. The SubscribeResponse message that the CBR service sends is shown in Listing 2.

According to the WS-Addressing specification, the WS-Addressing To header has the same contents as the WS-Addressing ReplyTo header in the original request. The RelatesTo header contains the contents of the Subscribe request MessageID header. In the body, the ID element contains a unique ID for the subscription generated by the event source, in this case the content-based routing service. The ID element can be used by the subscriber, the inventory application in this case, to maintain the subscription for example, for example, unsubscribing or renewing a subscription.

Notifications

Listing 3 is an example of NewProduct messages as sent by the CBR service to the subscribing inventory application.

Subscription Ends, Renews, and Unsubscribes

Several other messages may be used to optimize the subscription process. If a subscription expires, the CBR service will send a SubscriptionEnd header block in either one of the notification messages (the NewProduct document in this example) or attached to an otherwise empty message.

Although subscriptions will eventually expire based on the requested timeframe of the subscription or the event source's default expiration policy, if an application decides that it is no longer interested in receiving notifications, it should send an Unsubscribe message to the content-based router service or other event source. It should receive an UnsubscribeResponse in reply. If an application knows that its subscription is about to expire then it can send a request to renew the subscription, via a Renew message, and should receive a RenewResponse message in reply.

The CBR service alpha used in this example does not yet support these maintenance messages, so we won't discuss them here. Refer to the WS-Eventing specification for more details.

Code Sample

Listing 4 shows a functional C# program that sends WS-Eventing Subscribe messages to a content-based routing service (it is also downloadable along with the entire C# project at http://adamblum.com/CBRSubscribe. zip). You can simply invoke this program with arguments of:

- The endpoint to send notifications to
- The XPath expression of the documents to subscribe to
- The URL of the content-based router service

After the subscription is established, you can create Web services with any Web service development tool. If they send documents to the content-based routing service address (e.g., http://localhost:6060/cbr) that matches the expressions described in the CBRSubscribe program invocation, those XML documents will be forwarded by the CBR service to the notification endpoint identified in the CBRSubcribe invocation. If you want easy generation of these Web services clients, you may wish to start with an existing WSDL file. Just change the port in the WSDL to use the address of the content-based routing service. Your Web service client will then send XML messages corresponding to your WSDL to the CBR service for possible routing. Refer to the tutorials of your favorite Web services development tool for more information how to create generic Web services client using the WSDL of your choice.

Working More with Content-Based Routing

A functional WS-Eventing–based content-based routing service is posted and downloadable at www.systinet.com/Content BasedRouting. Addresses to a hosted and operational content-based routing service may also be posted there as well. Also, more full-fledged tutorials expanding on this scenario are available there.

Summary

XML content-based routing provides a very efficient way to integrate multiple applications around individual business events. While integration platforms that offer publishing and subscription to topics have enabled this for some time, the advent of WS-Addressing and WS-Eventing brings such efficiencies to the world of Web services. Companies that wish to integrate many applications simultaneously should consider using WS-Eventing–based integration capabilities to perform centralized yet Web services-based integration.

References

- *WS-Addressing by Adam Bosworth et al:* http://msdn.microsoft.com/webservices/default.aspx?pull=/library/en-us/dnglobspec/html/ws-addressing.asp March 2003.
- *WS-Eventing by Cabrera et al*: <http://msdn.microsoft.com/webservices/community/workshops/default.aspx?pull=/library/en-us/dnglobspec/html/ws-eventing.asp> http://msdn.microsoft.com/webservices/community/workshops/default.aspx?pull=/library/en-us/dnglobspec/html/ws-eventing.asp, January 2004.

About the Author

Adam Blum is an evangelist for Systinet Corporation. He spent five years with Microsoft, most recently as a manager in Microsoft's SQL Server product group. Adam has also worked as vice president of Engineering for CommerceOne where he managed development of all XML-related tools and platforms, including MarketSite, the world's first XML-based document integration server.

■■■ adam.blum@systinet.com

Listing 1

```
<s12:Envelope
  xmlns:12’http://www.w3.org/2003/05/soap-envelope’
  xmlns:wsa=’http://schemas.xmlsoap.org/ws/2003/03/addressing’
    xmlns:wse=’http:// schemas.xmlsoap.org/ws/2004/01/event-
ing'
    xmlns:eri=’http://electronicsretailer.com/inventory’>
    <s12:Header>
      <wsa:Action>

http://schemas.xmlsoap.org/ws/2004/01/eventing/Subscribe
      </wsa:Action>

<wsa:To>http://electronicsretailer.com/CBRService</wsa:To>
      <wsa:ReplyTo>
         <wsa:Address>
            http://electronicsretailer.com/inventory
         </wsa:Address>
      </wsa:ReplyTo>
      <wsa:MessageID>
        uuid:d7c5276b-de29-4313-b4d4-b3425b200840
      </wsa:MessageID>
    </s12:Header>
    <s12:Body>
       <wse:Subscribe>
          <wse:NotifyTo>
            <wse:Address>                  http://elec-
tronicsretailer.com/inventory/HandleNewProduct.asp
            </wse:Address>
            <wsa:ReferenceProperties>

<eri:LocalSubscriptionID>1234</eri:LocalSubscriptionID>
            </wsa:ReferenceProperties>
          <wse:NotifyTo>
          <wse:Filter
xmlns:er=’http://electronicsretailer.com/schemas’>
            /s12:Envelope/s12:Body/er:NewProduct
          </wse:Filter>
       </wse:Subscribe>
    </s12:Body>
</s12:Envelope>
```

Listing 2

```
<s12:Envelope
    xmlns:12’http://www.w3.org/2003/05/soap-envelope’
    xmlns:wsa=’http://schemas.xmlsoap.org/ws/2003/03/address-
ing’
    xmlns:wse=’http:// schemas.xmlsoap.org/ws/2004/01/event-
ing'
    xmlns:eri=’http://electronicsretailer.com/inventory’>
   <s12:Header>
      <wsa:Action>

http://schemas.xmlsoap.org/ws/2004/01/eventing/SubscribeRespons
e
      </wsa:Action>

<wsa:To>http://electronicsretailer.com/inventory</wsa:To>
      <wsa:RelatesTo>
        uuid:d7c5276b-de29-4313-b4d4-b3425b200840
      </wsa:RelatesTo>
    </s12:Header>
    <s12:Body>
       <wse:SubscribeResponse>
           <wse:Id>uuid:5005cfe6-c2c6-4296-9c3a-
80b9ad111813</wse:Id>
           <wse:Expires>2004-03-01T00:00:00-000-
00:00</wse:Expires>
       </wse:SubscribeResponse>
    </s12:Body>
```

```
</s12:Envelope>
```

Listing 3

```
<s12:Envelope
    xmlns:12’http://www.w3.org/2003/05/soap-envelope’
    xmlns:wsa=’http://schemas.xmlsoap.org/ws/2003/03/address-
ing’
    xmlns:wse=’http://schemas.xmlsoap.org/ws/2004/01/eventing'
    xmlns:eri=’http://electronicsretailer.com/inventory’
    xmlns:er=’http://electronicsretailer.com/schemas’>
    <s12:Header>
      <wsa:Action>
       http://electronicsretailer.com/schemas/NewProduct
      </wsa:Action>

<wsa:To>http://electronicsretailer.com/inventory</wsa:To>
      <eri:LocalSubscriptionID>1234</eri:LocalSubscriptionID>
    </s12:Header>
    <s12:Body>
       <eri:NewProduct>
          <eri:ProductID>AC-MP471</eri:ProductID>
          <eri:ProductName>
              Acme 128MB Portable MP3 Player
          </eri:ProductName>
          <eri:ProductCategory>MP3
Players</eri:ProductCategory>
          <eri:Price>47.56</eri:Price>
       </eri:NewProduct>
    </s12:Body>
</s12:Envelope>
```

Listing 4: CBRSubscribe.cs

```
//            CBRSubscribe
//
// This is a command line tool that can be used to subscribe
// and endpoint for notifications from a WS-Eventing based
Content-Based Routing service
//
//  Usage: CBRSubscribe <web service for notifications> <fil-
ter expression> [CBR URL]
   //
   // - the first argument is the address of the endpoint where WS-
Eventing notifications
   //    (documents matching subscription criteria) should be sent
//   - the second argument is the filter expression used for
the subscription
//   - the third argument is the URL for the content-based
routing service itself
//
//  For convenience this has been packaged as a standalone
command line tool.
//  However, most likely you would use the Subscribe function
alone in your own C#
//  or other .NET programs.

using System;
namespace CBRSubscribe
{
class MainClass
 {
  [STAThread]
  ///////////////////////////
  /// Main - Acts as simple shell to grab command line argu-
ments
  static void Main(string[] args)
  {
   if (args.Length<3)
   {
    Console.WriteLine("Usage: CBRSubscribe <web service for
notifications> <filter expression> <CBR URL>");
    return;
   }
   bool result;
   result=Subscribe(args[0],args[1],args[2]);
   if (result)
    Console.WriteLine("Endpoint {0} is successfully subscribed
to documents matching {1} with CBR service
{2}!",args[0],args[1],args[2]);
   else
    Console.WriteLine("Subscription failed!");

   return;
  }

  /// Subscribe function that does the actual work
  /// <param name="notifyAddress">
  /// this is the address of the endpoint where notifications
should be sent
  /// </param>
  /// <param name="subscriptionFilter">
  /// the XPath expression used to identify the documents of
interest
  /// </param>
  /// <param name="cbrServiceURL">
  /// The URL where the Content-Based Routing service is
located.
  /// </param>
  /// <returns>boolean to indicate success or
failure</returns>
  static bool Subscribe(string notifyAddress,string
subscriptionFilter,string cbrServiceURL)
  {
   CBRSubscribe.CBRService.Subscribe subscription=new
CBRSubscribe.CBRService.Subscribe();
   subscription.NotifyTo=new
CBRSubscribe.CBRService.EndpointReferenceType();
   subscription.NotifyTo.Address=new
CBRSubscribe.CBRService.AttributedURI();
   subscription.NotifyTo.Address.Value=notifyAddress;
   subscription.Filter=new
CBRSubscribe.CBRService.MessagePredicateAssertion();
   subscription.Filter.Value=subscriptionFilter;

   CBRSubscribe.CBRService.Eventing objEventing=new
CBRSubscribe.CBRService.Eventing(cbrServiceURL);
   CBRSubscribe.CBRService.SubscribeResponse
subscribeResponse=new
CBRSubscribe.CBRService.SubscribeResponse();
   subscribeResponse=objEventing.SubscribeOp(subscription);

   // sanity check the subscription results
   bool result=true;
   if (!subscribeResponse.Id.StartsWith("uuid:"))
    result=false;

   return result;
                }
          }
   }
```

Parasoft Offers AEP Team Starter Kits

(Monrovia, CA) – Parasoft, a provider of Automated Error Prevention (AEP) software, has announced the general availability of AEP Team Starter Kits, which enable development organizations of all sizes to effectively prevent software errors by following the same best practices and error prevention techniques across an entire team. The Starter Kits include Team Configuration Manager (TCM), Group Reporting System (GRS), and SourceScanner bundled with Parasoft automated unit testing and coding standard analysis products. Now, groups of any size can achieve the full benefits of automated coding standards analysis, unit testing, and regression testing for C/C++, Java, and .NET development.

Within the Starter Kit configuration, developers run Parasoft tools on their desktops to apply practices such as unit testing and coding standard enforcement. Architects use an extended version of the same tool to configure and control settings that all team members will use to perform the required practices. www.parasoft.com

When there's no margin for error...
Parasoft Automated Error Prevention (AEP) Products & Sol
Executive Overview
Media Room

webMethods Unveils Framework Solutions for Financial Services

(Fairfax, VA) – webMethods, Inc., a Web services infrastructure company, has announced its Financial Services Frameworks – four tailored solutions that address key financial services business initiatives. Each solution framework consists of a roadmap that highlights the products and services necessary to achieve overall "excellence" in operations, risk management and regulatory compliance, product and services delivery, and customer service initiatives. www.webMethods.com

Avanade Helps Customers with Web Services–Enabled Collaboration

(Mountain View, CA) – Avanade Inc., a technology integrator for Microsoft solutions in the enterprise, has announced that several of its customers are already benefiting from solutions incorporating Microsoft BizTalk Server 2004, a reliable, secure, and scalable platform for enhanced business process and Web services. Avanade applies proven techniques and best practices to speed design and delivery of solutions using BizTalk Server 2004 and other Microsoft technology.

Avanade BizTalk Server 2004 engagements have been driven primarily by two customer needs: strengthening internal business process and exposing business functions to external partners. For these projects, BizTalk Server 2004 acts as an engine of integration. Avanade is also utilizing BizTalk Server 2004 as a significant component of Web services "gateways" that expose systems or processes to external audiences. This approach to customers' Web services initiatives offers greater control over the applications exposed and faster deployment than alternatives such as custom development. www.avanade.com

WS-I Publishes Web Services Security Interoperability Guidelines

(San Francisco) – The Web Services Interoperability Organization (WS-I) has announced the availability of the first Security Scenarios Working Group Draft for public review. Developed by the WS-I Basic Security Profile Working Group, the Security Scenarios document identifies security challenges and threats in building interoperable Web services and countermeasures for these risks.

The Security Scenarios document describes several security challenges, threats, and countermeasures in building interoperable Web services, as well as usage scenarios and solutions.

WS-I is also currently working on the Basic Security Profile, an interoperability profile involving transport security, SOAP messaging security, and other security considerations implicated by the Basic Profile 1.0. The Basic Security Profile is intended to work with other WS-I profiles and will reference existing specifications used to provide security, including the OASIS Web Services Security 1.0 specification, and provide clarifications and guidance designed to promote interoperability of those specifications. www.ws-i.org

Microsoft, Intel Release Web Services Spec

A group of high-tech companies, including Microsoft Corp. and Intel Corp., has released a Web services specification for discovering devices attached to a local area network.

The WS-Discovery protocol provides a means for devices, such as printers, and software running on an application server to announce themselves on a LAN and make themselves available to other systems that want to use their services. The protocol, which would work on a wireless or wired LAN, is meant to work in conjunction with other Web services protocols for system-to-system communications, including WS-Eventing, WS-Addressing, WS-Security, and WS-ReliableMessaging.

WS-Discovery is meant to complement another Web services standard for system-to-system discovery, called Universal Description, Discovery and Integration. UDDI is used in connecting systems across networks, as opposed to just one network.

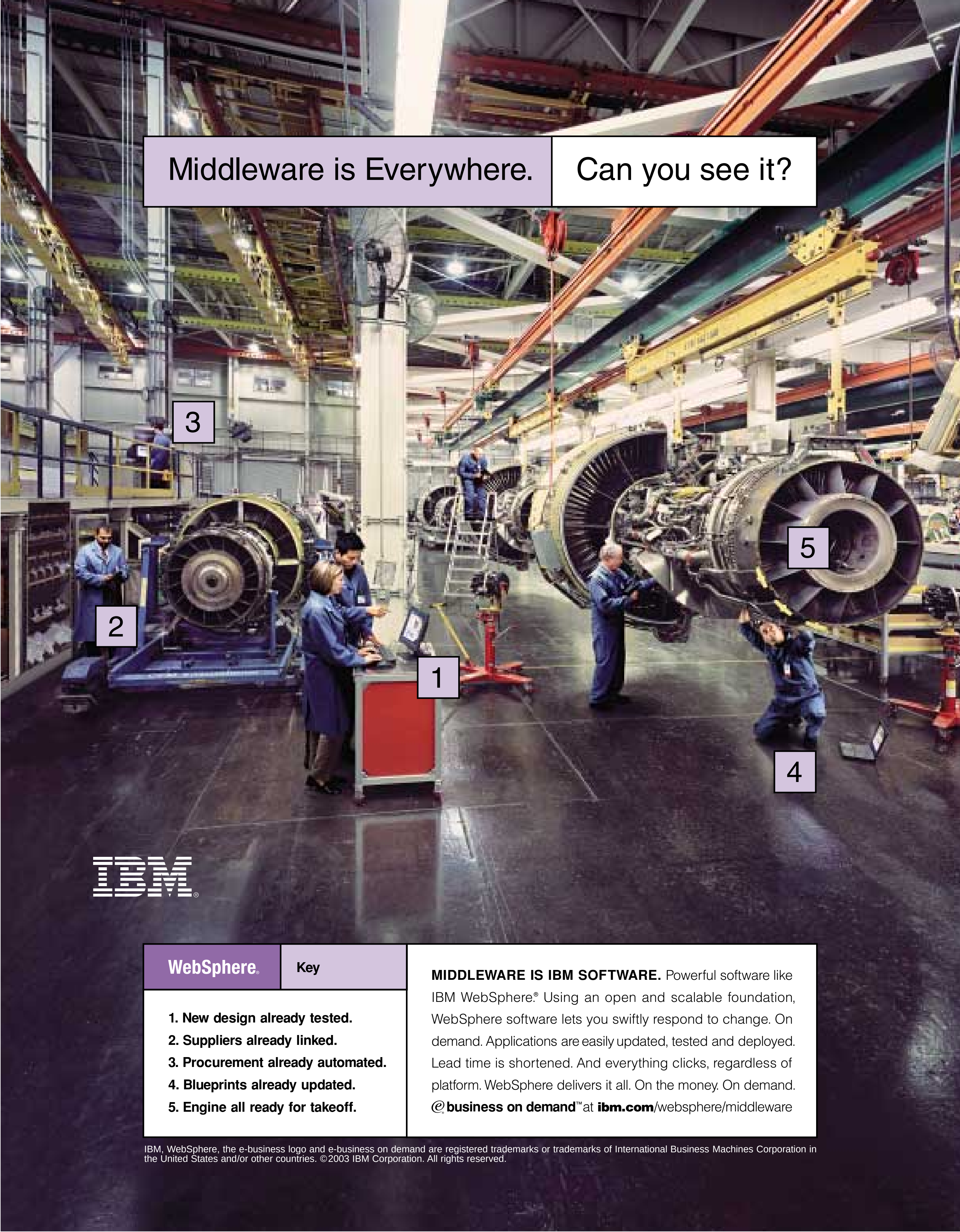
Web Services Help from LISA 2.0 Automated Testing

(Dallas) – iTKO, Inc., an enterprise software development tool provider, has released LISA 2.0 software for Web Services testing.

LISA 2.0 uses an "inline testing" technology to sit alongside and talk to any component that can impact a Web service. It can uncover and report the source of any Web service problems stemming from Web and app servers, network transmission, database storage and retrieval, middleware, other applications, and attached components. www.itko.com

JNetDirect Announces Solution for Translating Relational Data to XML

(Herndon, VA) – JNetDirect Incorporated, a data evolution software company, has announced the release of JSQLMapper, a bidirectional data mapping tool that eliminates the need to construct custom code in order to bring relational data into XML format. The beta release allows for rapid linking of business applications with the creation of relational data entries into a standards-based XML format and a map of those entries into a relational data source. www.jnetdirect.com



IBM

WebSphere. **Key**

1. **New design already tested.**
2. **Suppliers already linked.**
3. **Procurement already automated.**
4. **Blueprints already updated.**
5. **Engine all ready for takeoff.**

MIDDLEWARE IS IBM SOFTWARE. Powerful software like IBM WebSphere.® Using an open and scalable foundation, WebSphere software lets you swiftly respond to change. On demand. Applications are easily updated, tested and deployed. Lead time is shortened. And everything clicks, regardless of platform. WebSphere delivers it all. On the money. On demand.

@business on demand™ at **ibm.com**/websphere/middleware