



# Web Services

NET J2EE XM

August 2004 Volume 4 Issue 8

Planning for Service Management within a Service-Oriented Computing Infrastructure

A necessary infrastructure for mission–critical deployment pg.22

Six Tips for Moving
Web Services from
the Lab into Action
Flexibility, security,
and manageability
are the watchwords
pg.32

# SOA: Learning from the Past to Disrupt the Future

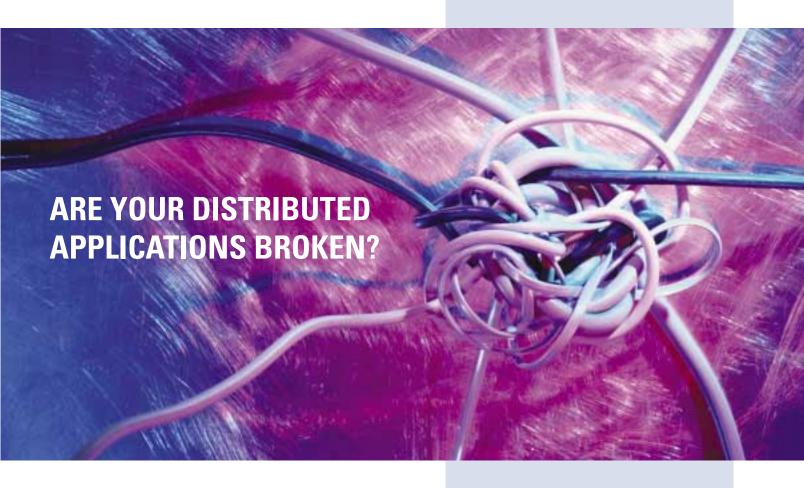
Easier to manage and built to change pg.38



# And Service-Oriented Architecture, Who Will Do the Cooking?

Understanding management trends in the new age of service-oriented architectures

(pg **12** 



# Fix them fast.

The SIFT™ solution framework from Service Integrity™ is the leading monitoring and analysis software for XML and Web services applications. SIFT will allow you to rapidly identify and resolve performance issues in your distributed applications. SIFT installs in minutes and automatically discovers your Web services and provides complete visibility into HTTP and XML data flows — all without a single line of custom coding. SIFT will untangle your problems fast.



# IT Applications of the SIFT™ Software:

- Intelligent Logging to save Time/Expense of Custom Coding
- Deep Visibility into Application Health and Security Status
- Improved Policy and SLA Compliance
- Fully configurable alerting
- Support for Billing and Metering
- And more...

"SIFT has been an invaluable tool and I don't know how we'd live without it.

We've been able to monitor system performance and isolate a few bugs that had gone undetected in the past."

Director of Operations at one of the largest commercial banks in the United States





Visit www.serviceintegrity.com for more information and enter to win a FREE COPY of Securing Web Services with WS-Security, by Dr. Jothy Rosenberg, CTO, Service Integrity, Inc. Fill out our registration form and you will be automatically entered to win!

# WebAppCabaret

http://www.webappcabaret.com/wsj.jsp

J2EE Web Hosting

Quality Web Hosting at a reasonable price...

# <JAVA J2EE HOSTING AND OUTSOURCING>

JAVA Developers: Design and Architect with Struts. Develop with Eclipse Build with Ant. Deploy to WebAppCabaret. It is that easy with a hosting company that understands the Java J2EE standard. Web Services - a buzz word or reality. How many web hosts provide the facilities to deploy and run applications written for Web Services? We Do. At WebAppCabaret, we lead the way in providing comprehensive Java J2EE and PHP/Perl Web Hosting solutions.

Easy Application Deployment is a reality at WebAppCabaret. Each Account's J2EE Application Server is installed with its own instance and default settings so you have complete control. Our Web Control Panel makes it a breeze for JVM restarts. Such is an example of our Advanced Web Hosting Infrastructure and Tools. If you are a consultant, do you have a complex web hosting requirement for your client? Web Host or Reseller, are you looking to provide services without the headaches of managing your own network infrastructure? Look no more.

For JAVA J2EE we offer the latest versions of Tomcat, JBoss, and Jetty Application Servers. For Scripting programming we offer the latest PHP and Perl. We also offer the latest MySql and PostgreSql Databases. Commercial software, including Resin, JRun, and Oracle, is also available for an extra charge. In addition we provide valuable tools such as Bugzilla Bug Tracking, CVS Version Control plus a whole lot more. Does your service provider offer ENCRYPTED ACCESS for Email (POP3S/SMTPS/IMAPS), Shell, and FTP? WE DO. All of this is backed by our Tier 1 Data Center with 100% Network Uptime Guarantee.

Below is a partial price list of our standard hosting plans. (Reseller accounts also available). For more details please log on to http://www.webappcabaret.com/wsj.jsp or give us a call at 1(866)-256-7973.

# \$39/mo Enterprise

Latest JBoss/Tomcat/Jettv Latest JSP/Servlets/EJBs Private JVM Choice of latest JDKs **Dedicated IP Address NGASI Control Panel** PHP and Perl Web Stats 1GB Disk **200MB DB** MySql PostgreSql Dedicated Apache Telnet . SSH . FTP 5 Domains 100 Emails Web Mail . POP . IMAP more...

# \$3000/mo 4Balance

4Balance is our entry level High Availability Load Balancing service comprised of: 1 Database Server 4GB RAM Dual Xeon 2Application Servers 2GB RAM Single Xeon 1 Load Balancer 1GB RAM Single Xeon

# **Options**

The following are some of the add-on options with the associated extra starting at costs:

Resin: \$10/mo

雅 雅·雅·雅 雅

300MB Oracle: \$40/mo

JRun: \$20/mo

ColdFusion: \$30/mo

#### VPN

Need to host a site outside your corporate network with secure access? Our VPN service is the solution for peace of mind.

# \$191/mo Dedicated

Managed Dedicated Server starts at \$191 per month for:

Our J2EE-Ready dedicated servers make deploying complex applications a breeze.

NGASI Control Panel with your own Logo Each dedicated server configured for standalone web application and web hosting (resellers) at no extra charge.

more...

# \$99/mo Reseller

If you cannot afford a Dedicated server for reselling web hosting, for \$99/mo the Shared Reseller plan gives you:

10 Professional Plans Your Web Site NGASI Control Panel with your own Logo 50% Discount No System Admin Easy Account Mgt

more..

# Z JOURNAL

**WSJ: FEATURES** 

# In a Service-Oriented Architecture, Who Will Do the Cooking?

Understanding management trends in the new age of service-oriented architectures By Paul Lipton

# **Planning for Service Management within** a Service-Oriented **Computing Infrastructure**

A necessary infrastructure for mission-critical deployment By Bill Jones & Kevin Minder

**22** 

# **SOA: Learning from the Past** to Disrupt the Future

Easier to manage and built for change By Kerry Champion

**BOOK EXCERPT Web Services** Messaging with JAX-RPC



FROM THE EDITOR

# eBay

7 By Sean Rhody ......

# **INDUSTRY COMMENTARY**

# **From Web Services Management** to Utility Computing

10 By William Vambenepe.....

# WSJ: SECURITY

# **Beyond XML Firewalling**

Security coordination for Web services

18 By Toufic Boubez .....

# WSJ: ANOTHER VIEW

# The Myopia of Web Services Management

What's in your architecture?

26 By Sam Boonin ......

# WSJ: PRODUCT REVIEW

# **Oracle BPEL Process Manager**

Making SOA design easy

28 By Brian Barbash.....

# WSJ: WEB SERVICES@WORK

# **Web Services and Federated** Identity

Standards to better define a space

30 By David S. Linthicum

# WSJ: WEB SERVICES MANAGEMENT

# **Six Tips for Moving Web Services** from the Lab into Action

Flexbility, security, and manageability are the watchwords By Joelle Gropper Kaufman.....



# Banking on a Standard

BY MIKE HAEHN

# **Using XSLT on Bioinformatic** XML data

BY RUSSEL BRUHN PHILIP BURTON GARY THOMPSON

# Look Ma Bell -No Hands!

BY LES WILSON 56

www.wsi2.com August 2004

# How to Make Money with Web Services

By Bob Brauer bob.brauer@strikeiron.com



# A new way to work with Web services - build them and they will come?

The benefits of Web services are well understood – lower integration costs, maximum reusability, faster deployment, more automation, easier to work with new partners, and so on. You can build them but that does not necessarily mean that people will use them or that you can make money on them. Why? And more importantly, how?

# What drives Web services utilization and return on investment?

Here are some key points to consider:

- Accessibility and ease-of-use
- Billing and accounting
- Subscriptions and trials
- Availability, reliability, and security
- Promotions and pricing

# What tools and services do you need to commercialize your Web services?

- Tools to improve understanding and use.
- Commerce capabilities to manage subscriptions, accounting, billing, payments, account management, etc.
- Ability to manage free trials and convert trial users to subscribers.
- Service levels that ensure availability, reliability, and security.
- Knowledge about acceptable pricing structures based on value of the data, process, accessibility, and performance.
- A way to deliver and promote them to the appropriate target audience.

# Creating a new revenue channel

Your Web services provide value and you need to be reimbursed for that value. However, before that can happen, you need a distribution channel with the infrastructure to publish and sell your Web services. This channel must take care of issues such as delivery, account set up, billing and collections, marketing, and customer support, to free you of having to make that investment.



# The answer is Premium Web Services

Premium Web Services provided by StrikeIron and available through the StrikeIron Web Services Business Network™ (WSBizNet™) are the answer.

Whether you are a Web service provider wanting to commercialize your Web services or a company wanting to find a way to manage your Web services without the infrastructure investment, StrikeIron will do the work for you, and you make money in the process.

StrikeIron Premium Web Services are subscription-based Web services that deliver "live" data and functionality over the Internet via XML using a dynamic SOAP interface. Premium Web Services are supported by the commercial services and easier-to-use tools of the StrikeIron WSBizNet to remove barriers to purchase.

# **Premium Web Services benefits**

- An alternate sales channel to reach more customers for minimal cost.
- Web services hosting and delivery services in a secure environment.
- Pricing, billing, collections, and support.
- Reporting for easier maintenance and tracking.

# The StrikeIron Web Services Business Network

The StrikeIron WSBizNet is a subscription-based online network of services and tools for easier and faster utilization of Web services by business users and developers.

The WSBizNet allows anyone, with minimal experience, to quickly find, understand, and utilize Web services for accessing information, integrating with existing applications, and assembling new applications.

# StrikeIron WSBizNet benefits

- Service levels to meet your business requirements.
- Integrated online StrikeIron
   Web Services Analyzer for faster
   understanding of Web services.
- Integrated online Knowledge Base makes it easier to understand and work with your Web services.
- Prime positioning in the WSBizNet.
- Increased visibility in one of the largest Web services business directories.

# Delivering Web Services to the World!

For Free Trials and more about how to make money with the StrikeIron Web Services Business Network and Premium Web Services, please visit our Web site at www.strikeiron.com.

StrikeIron, Inc. +1.919.405.7010 sales@strikeiron.com www.strikeiron.com

©2004 StrikeIron, Inc. All rights reserved. StrikeIron products and services mentioned herein are trademarks or registered rademarks of StrikeIron, Inc. All other brands or product name are the property of their respective holders.



# **Overcoming Unit Testing Obstacles**

Unit testing, as commonly performed, is a difficult and resource-intensive process. First, the developer must analyze the code and determine how to add tests that would best verify the unit. Next, the developer must build a test framework. Even when the framework is well-defined, constructing each framework still requires coding. Finally, the developer must design and implement test cases, ensure that the test performs the necessary setup, and execute the test cases.

When a team begins to implement unit testing, the team members are typically excited by the novelty of the practice and they start writing test cases. However, the novelty soon wears off. In fact, unit testing is often abandoned altogether-especially by teams that do not have a way to automatically generate unit test cases, then expand these test cases to verify specific

Unfortunately, even the best unit testing tools cannot-on their own- ensure that a team performs unit testing. Truly implementing a practice such as unit testing in a team environment involves more than just tools. It also requires the team culture. workflow, and supporting infrastructure needed to embed the practice into the team's development process.

To help development teams make unit testing an enduring part of their development process, Parasoft has defined the unit testing practice as part of the Parasoft AEP Methodology, a comprehensive strategy for preventing errors in a team environment. To learn how this methodology works, visit http://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 x3303 or Email buzz@parasoft.com



# Founded in 1987, Parasoft's clients include IBM, HP, DaimlerChrysler and over 10,000 companies worldwide.

# **Protocol Support**

# **Platforms**

# Web Services NET JZEE XML JOURNAL

#### INTERNATIONAL ADVISORY BOARD

rew Astor, David Chappell, Graham Glass, Tyson Hartmar Paul Lipton, Anne Thomas Manes, Norbert Mikula, 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

Sean Rhody sean@sys-con.com XML EDITOR

Hitesh Seth

#### INDUSTRY EDITOR

Norbert Mikula norbert@sys-con.com
PRODUCT REVIEW EDITOR

Brian Barbash bbarbash@sys-con.con .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

## EXECUTIVE EDITOR

Gail Schultz gail@sys-con.com MANAGING EDITOR

Jennifer Van Winckel iennifer@svs-con.com

EDITOR

# Nancy Valentine nancy@sys-con.com ASSOCIATE EDITOR

Jamie Matusow jamie@sys-con.com

ASSISTANT EDITOR

Torrey Gaver torrey@sys-con com

### **PRODUCTION**

PRODUCTION CONSULTANT

LEAD DESIGNER

Richard Silverberg richards@sys-con.com

ART DIRECTOR
Alex Botero alex@sys-cor

# ASSOCIATE ART DIRECTORS

Louis F. Cuffari louis@sys-con.com Tami Beatty tami @sys-con.com

# **CONTRIBUTORS TO THIS ISSUE**

Brian Barbash, Sam Boonin, Toufic Boubez, Russel Bruhn, Philip Burton, Kerry Champion, Mike Haehn, Mark Hansen, Bill Jones, Joelle Gropper Kaufman, David Linthicum, Paul Lipton, Kevin Minder, Sean Rhody, Gary Thompson, William Vambenepe, Les Wilson

# **EDITORIAL OFFICES**

135 CHESTNUT RIDGE ROAD, MONTVALE, NJ 07645
TELEPHONE: 201 802-3000 FAX: 201 782-9637
WEB SERVICES JOURNAL (ISSN# 1535-5906)
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 ©COPYRIGHT

Copyright © 2004 by SYS-CON Publications, Inc. All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy or any information storage and retrieval system without written permission. For promotional reprints, contact reprint coordinator SYS-COP Publications, Inc., reserves the right to revise, republish, and authorize its readers to use the articles submitted for publication.

All brand and product names used on these pages are trade names, service marks, or trademarks of their

respective companies.

SYS-CON Publications, Inc., is not affiliated with the companies or products covered in Web Services Journal.

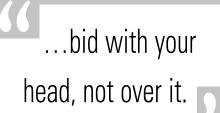


# **eBay**

was recently invited to be a guest speaker at the eBay Developers Conference, where I was part of a panel whose topic was "Delivering the Promise of Web Services." I found it particularly interesting, if slightly worrisome. What struck me most was the differentiation between Web services consumers and Web services authors.

The audience at this presentation asked a large number of questions that helped me understand that consumers and producers are not the same groups of people. eBay developers are particularly interested in consuming Web services (such as the eBay and PayPal APIs that are exposed as Web services), but in general they were not interested in developing and publishing their own Web services. This shaped the discussion in a way that was enlightening for me because it highlighted some of the things that the end users of the services we discuss so much are coping with.

In particular, these consumers were looking for better Web services consumer mechanisms in their languages of choice, which were not the compiled or



near–compiled languages of the developers, who chose to develop services in languages such as Java or C#. Instead, these developers were concerned with the quality of the Web services toolkits in languages such as PHP, Perl, and Python.Once you understand that their primary interest is using the API on their sites, without having to use the eBay interface, you can easily grasp why these open source, thin client-oriented languages occupied a large part of the conversation.

It also tended to underscore a concern I've been feeling for some time now regarding Web services. We, the industry and specification leaders, are ignoring the last mile. The initial set of standards – UDDI, XML, WSDL, and SOAP – are now well entrenched and providing meaningful interoperability between software platforms. The second generation of stan-



WRITTEN BY

SEAN RHODY

dards, which includes things such as WS-Transaction, WS-Security, WS-Reliable Messaging, are starting to result in even greater functionality that can be provided with Web services. But by and large, these standards are clearly aimed at the Web services developer community, and in the end have only so much impact on the Web services consumers.

Now, we can say that Web services is about services themselves, and not user interfaces, and we'd largely be right, but not fully. In the end, a service needs to be used, and even though Web services technologies make it easier to do computer-to-computer conversations, the reality of business computing is that the overwhelming majority of services will be consumed by people. And those people need some sort of visual interface with which to interact (yes, except for IVR).

Now, obviously it is harder to divorce the UI layer from proprietary technology and languages, and this makes it harder to actually come up with a uniform standard for the interface. It's also true that the service is distinct from the interface and for maximum flexibility should be neutral to interface requirements. In fact, trying to come up with a description language for the interface is probably not worthwhile (even if you ignore the fact that HTML is already such a language). Instead, we need to make the intermediate manipulation languages, such as ASP, JSP, PHP, Python, and Perl easier and bulletproof. We also need to give some consideration to common UI tasks such as set iteration. There's nothing in the Web services specifications towards that end, but certainly applications that use Web services will have to make use of a variety of common presentation concepts where an intermediate presentation service language could shape and mold the generic XML message returned from a service invocation into a screen or page layout. Ideally this is again generic, so that varying presentation services can be utilized.

I learned one other thing from eBay – bid with your head, not over it. Now what do I do with a black, chrome-covered parade saddle? Sometimes I amaze myself.

# 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

W W W . W S j 2 . ( 0 M August 2004



**CONFERENCE:** August 2 – 5, 2004 **EXPO:** August 3 – 5, 2004

MOSCONE CENTER • SAN FRANCISCO, CA

# LinuxWorld Conference & Expo

is the #1 marketplace for companies that sell, market or promote open source based products, services, applications and solutions. Experiencing astounding growth, it's the largest gathering of open source professionals in the world.





CORNERSTONE SPONSOR

PLATINUM SPONSORS















# **Keynote Speakers**



Matthew Szulik, Red Hat Chairman, Chief Executive Officer and President Tuesday, August 3

9:15 am - 10:00 am



Nick Donofrio, Senior Vice President, Technology and Manufacturing, IBM Corp. Wednesday, August 4 11:00 am – 11:45 am



Martin Fink, Vice President, Linux Business, Enterprise Storage and Services Business Unit, Hewlett-Packard Company





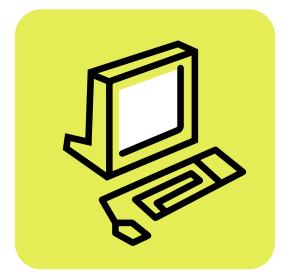
Alfred Chuang, BEA Systems Founder, Chairman and CEO Wednesday, August 4 2:30 pm – 3:15 pm



Michael Steven Rocha, Executive Vice President, Global Support Services & Platform Technologies, Oracle Corporation

Tuesday, August 3 2:45 pm - 3:30 pm





# Attend LinuxWorld and...

- Evaluate the latest Linux and open source products and solutions showcased by the industry's top companies
- Discover real business solutions for real business problems
- Network with peers and gain valuable insight from open source industry leaders
- Participate in LinuxWorld's world-class educational conference and realize Linux as more than an operating system, but as a world of applications
- Learn how Linux can save your company time and money

Register Online With Priority Code: D2501





PRESIDENT AND CEO

Fuat Kircaali fuat@sys-con.com

VP, BUSINESS 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.con

VP, SALES & MARKETING

Miles Silverman miles@sys-con.com

ADVERTISING DIRECTOR

Robyn Forma robyn@sys-con.com

ADVERTISING MANAGER

ASSOCIATE SALES MANAGERS

Kristin Kuhnle kristin@sys-con.com Beth Jones beth@sys-con.com Dorothy Gil dorothy@sys-con.com

SYS-CON EVENTS

PRESIDENT, SYS-CON EVENTS

Grisha Davida grisha@sys-con.com

#### **CUSTOMER RELATIONS/JDJ STORE**

CIRCULATION SERVICE COORDINATORS

Edna Earle Russell edna@sys-con.com Linda Lipton linda@sys-con.com

SYS-CON.COM

VP, INFORMATION SYSTEMS

Robert Diamond robert@sys-con.com

WEB DESIGNERS
Stephen Kilmurray stephen@sys-con.com

Mathew Pollotta mathew@svs-con.com

# ACCOUNTING

FINANCIAL ANALYST

Joan LaRose joan@sys-con.com
ACCOUNTS PAYABLE

ACCOUNTS PAYABLE

ACCOUNTS RECEIVABLE

Shannon Rymsza shannon@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:

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.



# From Web Services Management to Utility Computing

nterprise services are created by combining infrastructure services, applications, and business processes. To be able to adapt quickly to business changes, enterprise IT must evolve from management of individual resources to management of interrelated services. This will be achieved through the development of composable and modular standards that expose the management capabilities of the building blocks of enterprise servic-

es. The Web services platform is an enabler of this transformation: a Web services—based management infrastructure provides a channel that is appropriate for dynamic resource provisioning, allocation, and configuration – often called utility computing.

We can consider this management infrastructure as a four-layered architecture. Starting at the foundation layer, the work on the base Web services infrastructure is far from over. First, until WSDL 2.0 is widely deployed, designers have to compose around the deficiencies of WSDL 1.1, such as the lack of portType inheritance. Second, there is still no standard for referencing Web services. Finally, key specifications such as WSRF (Web Services Resource Framework) and WSN (Web Services Notification), without which people were left to reinvent Web services interfaces to access stateful resources, have only recently reached the standards community. These issues are being resolved and a set of building blocks for accessing resources through an SOA (service-oriented architecture) is shaping up. It is critical that these building blocks be modular and composable to allow incremental adoption and separation of concerns.

Moving from the foundation to the management protocol layer, the OASISWSDM (Web Services Distributed Management) technical committee, through its MUWS (Management Using Web Services) specification, is the key articulation point between the base Web services architecture and utility computing. Both the IT management community and the Grid community rely on MUWS. It defines how to express and exercise manageability capabilities through Web services, putting in place a management channel that is more interoperable and accessible than ever before.

Next is the modeling layer. Information models need to be composed so that a service can be represented based on the services that it is assembled from.



WRITTEN BY
WILLIAM
VAMBENEPE

be they peer or infrastructure services. Since these will be described by different models, the management channel (MUWS) needs to be model-agnostic in order to support a model-centric architecture. For example, CIM (Common Information Model) is a model that focuses on concrete resources. The DMTFWS-CIM subgroup must now open CIM to the Web services platform by developing a standard way to expose CIM-modeled resources through MUWS. Other models provide representations for service security, service-level agreements (SLA), etc. Only by composing

these models will, for example, an auction service SLA be adequately managed as it depends on a combination of the performance of the servers on which the service runs, the application server that hosts it, the other services (authentication, billing, etc.) that it makes use of, and the business process engine that controls the bidding. Once this model-centric architecture is in place, management actions can be policy–driven through explicit constraints.

Finally, at the top layer, the architecture includes a set of common services for utility computing. They are being defined collaboratively by DMTF (Utility Computing working group) and GGF (OGSA working group).

All the pieces are falling into place but much remains to be done to allow comprehensive management of enterprise services in a model-centric way through Web services standards. While it would be easier to develop an end-to-end model specific to one company's offering, standardization allows the integration of the management capabilities of all the components that compose enterprise services. We must keep the pressure on vendors to deliver modular and composable specifications (for format, function, and protocol) that expose management capabilities of infrastructure services, applications, and business processes in such a way that these capabilities can be composed by the next generation of management applications. These applications will use this to synchronize business and IT and to capitalize on change.

# About the Author

William Vambenepe is a senior architect in HP's Advanced Technology Office. As part of his work on management for the adaptive enterprise, William represents HP in several of the Web services standard activities mentioned in the article.

■■■ vbp@hp.com

# Web Services Help

DEBUG
SOAPscope 3.0
TUNE

Try it free at www.mindreef.com



The acclaimed essayist and novelist Nora Ephron once said, "What my mother believed about cooking is that if you worked hard and prospered, someone else would do it for you." Nothing could better capture the spirit of service-oriented architectures (SOAs) than this statement from a person who clearly does not consider cooking a core competency. Translated to human terms, an SOA can help make sure that the right person is doing the cooking at the right time.

he idea of a service-oriented architecture is simple, and much older than any Web services standard. Instead of services being statically bound to each other in some sort of "hard-coded" relationship (which is often the case for many real-world Web service deployments today), in an SOA, service consumers can discover the service providers that they need and use them as required.

Typically, information about these other services is stored in a database or directory, often referred to as a registry by the Web services savvy. To reiterate an often-used analogy, this is similar to how consumers can discover needed services in the phone book.

For business, this architecture can serve as a foundation for distributed systems that are far more flexible and responsive to the organization. This is especially important today. Factors such as globalization, higher customer expectations, and increased regulatory pressure have put pressure



PAUL LIPTON

on IT to be more responsive to the needs of the business, and SOAs are widely held as the key architectural element to making this possible. In short, SOAs are about encapsulating valuable functionality as services made available inside and outside the enterprise, and leveraging those available services to build a better, more flexible, and more useful enterprise IT environment.

# A New Paradigm

Certainly many essential services will be hosted within the enterprise, but others will just as surely be hosted by specialists who will provide their own expertise to other businesses. A number of useful commercial Web services in areas such as sales force automation and customer relationship management will undoubtedly appear. Significant business services are likely to be offered by suppliers, government regulators, customers, and partners as well.

An example of one such service that is already

taking hold is the use of outsourced geographic services exposed as Web services. This is an increasingly popular option for some businesses. Rather than try to master the arcane geographic arts, these businesses are beginning to focus on their own core competencies and are content to leave other tasks to experts. In the area of geographic services, Microsoft's MapPoint Web service, which handles millions of service requests a day, would be a good example.

Customers today are increasingly mobile. Wireless devices, many GPS-enabled, are starting to flood the market. Most of the vendors of the underlying operating systems in these devices are already committed to including Web services stacks in future releases. This would make the devices even more useful in a world where enterprise SOAs will extend to touch the customers directly.

Given these trends, imagine a company like Starbucks being able to direct its customers to its many locations before they are lured into the storefronts of new and old competitors that have "smelled the money in the coffee." Using an outside specialist service to provide dynamically generated, location-dependent walking or driving directions would be very useful indeed to a company whose view on providing directions is centered on the art of brewing, not driving.

In fact, the trend in IT is not dissimilar from trends in the consumer market. In the latter half of the 20th century, many consumers in industrial societies were unlikely to grow their own food. But recently, consumers have left even more of

**12** August 2004 WWW.WSj2.com

# Introducing

# SOAPscope 3.0 Debug

# 4 Ways to Know Your Web Services

Whether you are learning how a Web service works, or troubleshooting a tough problem, you need the help of a "smart" tool. SOAPscope lets you dig deeper, faster.

1. Try It Solve problems by testing your Web service with different inputs without writing any code.

2. See It View WSDL and SOAP to understand what's happening. Capture from any toolkit, and see just the right detail for the task at hand.

3. Diff It Compare a problem message or WSDL with a similar, working one.

4. Check It When the problem's not obvious, rigorous interactive analysis finds inconsistencies, errors, and interoperability problems.

# Look What's ZNEW ? in 3.0

- Integrates with Microsoft<sup>®</sup> Visual Studio<sup>®</sup> .NET and Eclipse
- Graph Message Statistics
- Interactive Message Analysis
- Interoperability Testing System
- SSL Support
- HTTP Authentication Support
- HTTP Compression Support
- Support for multi-byte encoding
- Best usability of any Web Services tool

The most comprehensive Web services diagnostic system available, and still only \$99!

"SOAPScope 3.0 is easily the most encountered since Halo. When does the multi-player version come out?"

# Don Box

XML Messaging Architect, Microsoft and co-inventor of SOAP

addictive piece of software I've

Tes

Try It online at XMETHODS.net or download a trial at Mindreef.com



the "personal business process" of putting food on the table to others. Today there are an increasing number of food delivery options, as well as food markets that offer ready-made food that can be reheated at home. This allows busy consumers to focus on what they feel are their own personal priorities - their "personal core competencies" such as raising children, pursuing careers, or painting pictures, rather than taking valuable personal time to cook. Clearly Nora Ephron knew more about IT trends than she was letting on when she made her prescient comment about cooking!

# If Web Services Are the Trees, then SOAs Are the Forest

In the past, IT organizations focused primarily on the idea of using Web services to enable cheaper, more effective point-to-point integration. Now the idea has taken hold that Web services play only a part (albeit a very important one) in the larger, more strategic stage production called an SOA. How do Web services management/security products need to evolve in order to meet the needs of a service-oriented world? In other words, if Web services are the trees and SOAs are the forest, how do we manage the forest and how do we manage the other types of trees that live in the forest?

One of the great things about Web services standards is that they have been designed to support the idea of an SOA. In particular, the UDDI standard defines a registry for services, the WSDL standard defines a mechanism for service description that can be stored in the registry, and the SOAP standard defines the basic message format for information exchanged between services. However, despite the fact that Web services provide good mechanisms for building an SOA, in practical terms it is likely that not every service in many real-world enterprise SOAs will be a modern SOAP-based Web service.

While enterprises today often look forward to large, enterprise-scale SOAs or even multi-enterprise-scale SOAs, clearly less extensive SOAs have been successfully built and deployed using many other technologies, including CORBA, Java RMI/IIOP, and DCOM. Although Web services are already more widely accepted and supported by most commercial applications and software vendors, it is not likely that all existing services based on these older technologies will be converted to Web services, at least for some time to come.

There are many reasons for this decision to defer or avoid conversion to modern SOAP-based Web services. The cost of conversion to Web services may be considered too high relative to



# Web services encapsulate business logic that does the real work



the benefit accrued, or it is impossible that it will be higher performance or other unique characteristics of the older technology (likely based on more efficient proprietary binary protocols) that will drive the decision to build or extend a heterogeneous enterprise SOA. A heterogeneous enterprise SOA would preserve some of the diversity of different standards and technologies already present within the enterprise while still embracing new Web services standards and technologies whenever possible and practical.

Practicality and cost are, indeed, the cornerstones of this approach. While the long-term goal might still be a technologically purer SOA based entirely on Web services, the short-term practical goal would be tolerance of diverse underlying protocols and technologies as part of the larger SOA. After all, even WSDL and UDDI support far more than just SOAP-based Web services.

# If Web Services Are the Trees. What Are the Roots?

Web service management products can help ensure that Web services are visible and controllable. My recent articles (see Resources) have discussed many aspects of Web services management, such as the need for such products to be deeply integrated with enterprise management solutions in order to provide true root-cause analysis, and to provide the visibility and correlation necessary to understand and correct the real cause of any service disruption.

The principle is simple. Web services encapsulate business logic that does the real work as part of the traditional IT infrastructure - the heavy lifting, if you will. It is this underlying business logic that accesses databases, executes complex logic and business rules, and so on. Such business logic can function only when existing hardware (such as routers and disk drives) and software (such as operating systems and a wide range of diverse servers) function properly.

Web services exist at a logical level above the other logical layers of the IT infrastructure. Logical constructs like SOAP messages, identity, service description, and so on are logically independent, but are also highly dependent physically on lower layers such as TCP/IP. So, at least for

Web services deployed within the enterprise, the Web services management products must do two things. They must function at this higher, logical level of Web services and must also be deeply integrated with the enterprise management solution responsible for the rest of the IT infrastructure. This is necessary in order to understand and correlate events and concerns at all logical and physical levels. Ideally, the two levels of management - Web services and the rest of the IT infrastructure - while independent, would function as one from the point of view of the IT operations staff. In practical terms, it would be wise to recognize the justifiable reluctance of IT operations staff to have multiple management consoles rather, they would work through the same familiar enterprise management console using common terms and nomenclature.

# Managing the SOA Forest – **Active Management and Scale**

While Web services can be deployed as a point-to-point integration solution, enterprise SOAs are, by definition, dynamic and flexible. Such an environment requires an active management approach. To understand what active management is, it is best to begin by understanding what it is not. Active management is not about transformation, routing, load balancing, or other functions often provided since the early days of Web services by various software-based brokers and proxies.

Why are these functions not considered active management? In the cases of transformation and load balancing, there certainly is activity. Aren't these functions important in an SOA? Certainly, they are still important and need to be performed, but active management refers to the ability of a management system to optimize the state of whatever is being managed based on policies and rules defined in the management system.

Think of a system that controls the temperature, oxygen content, and lighting in a large outdoor aquarium. Even in this simple example, the policies that define a healthy environment for fish and plants in the water are clearly defined. The "agents" are the temperature and oxygen sensors in the water as well as the various switch-

4. August 2004 www.wsi2.com es and valves that are regulated by the management system. This is an active management system – actively manipulating the infrastructure of the aquarium based on metrics and policies. More advanced versions of such a system might include predictive analysis features to anticipate fish breeding patterns, and so on. Treating this system as an SOA and extending it to weather services might allow further optimization and responsiveness by leveraging the real-time forecasts provided by these services, but the basic principle of active management of the environment based on observed metrics remains the same.

Going back to functions like transformation, load balancing, and routing, these functions are already being subsumed by a new generation of XML hardware appliances designed and optimized for those purposes. This trend is similar to what happened with TCP/IP. Initially, it might have seemed reasonable to route and perform other functions in simple software brokers, often written quickly to leverage what was then an exciting new protocol. But these days nobody would dream of performing these functions without a box made by Cisco or another hardware vendor.

By the same token, nobody today would mistake the functionality of those boxes for a comprehensive enterprise management solution.

Leading enterprise vendors can continue to rest easy knowing that the capabilities they provide in their own solutions were not made irrelevant by the deployment of hardware-based TCP/IP routing, load balancing, and so forth in these boxes!

In fact, quite the opposite is true since these new devices became yet another point of potential failure that needs to be visible and controllable, just like the rest of the managed IT infrastructure.

Similarly, while simple routing and transformation tasks can probably be done at reasonable speed on the various platforms used to develop and deploy services such as integration servers, application servers and enterprise service buses, a new generation of hardware XML appliances from companies like DataPower, Forum Systems, Reactivity, and Sarvega are already performing these same functions at near wire-speed for XML-based services. The move to hardwarebased solutions for transformation, routing, and load balancing as well as other simple functions is likely to be universal and quick. For many applications, especially large-scale ones, XML parsing and transformation is a computationally expensive process best accomplished with hardware-based acceleration.

The challenge, of course, is to make sure that

these devices, which function at the service level, are themselves fully integrated into the SOA management solution. That is, of course, the best role for these devices - as fully managed gateways just like the TCP/IP devices, often deployed in the enterprise's DMZ, between inner and outer traditional firewalls. Such devices would work as full agents (sometimes called observers) of the SOAlevel management/security solution. They would work with the SOA management/security system to obtain and then act upon identity information and policy - reporting faults and errors, metrics, and more to the management system. The management system would present one common management and security console for such instrumented XML gateways, as well as for agents deployed on the service platforms themselves (see Figure 1).

# Managing the SOA Forest – The Challenge of Heterogeneity

What about heterogeneous SOAs, as discussed above? Their diversity will undoubtedly

increase as heterogeneous enterprise SOAs continue to grow. The growth rate will only accelerate as these SOAs begin to extend beyond the world of back-end systems to include mobile consumer devices. It will be some time before they all have full Web services stacks of their own. Many platform vendors will take an incremental approach – initially coping with platform limitations with various compromises and offering their own unique, idiosyncratic implementations.

If SOAs will often be heterogeneous – a practicality dictated by the need to support multiple protocols and types of services – how can we manage this diverse distributed environment? Web services management products in this new world must become unified SOA management and security solutions that can readily be adapted for many different types of services, not just Web services. This will require a new approach to how management agents are created.

The sheer diversity of some SOAs will mandate that it become easier for customers and their consultants to develop their own agents for

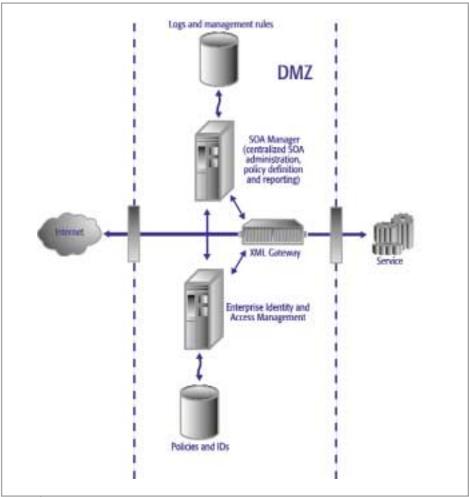


FIGURE 1 SOA management/security system

W W W . W S j 2 . C 0 m August 2004

these specialized platforms and technologies. Some sort of easy-to-use agent development kit will almost certainly be a necessity so that new agents can be created quickly and easily. Management vendors, in order to be serious partners for their customers building heterogeneous SOAs, must also offer comprehensive end-to-end management capability from wireless to Web services and beyond, encompassing the entire SOA and its underlying IT infrastructure.

For example, Computer Associates has released the next generation of CA WSDM, a standards-based SOA management and security solution with suitable improvements for the

to impact overall service performance by overloading the underlying systems.

This remote assessment could be used to obtain important information about the consistency of a service (how much it varies in its performance) over time. Consistency is a particularly important aspect of a service. Suppose that a service typically offers very high performance (say less than one second), but occasionally responds very slowly (say one minute). It might be possible that the average and mean performance measurements would still be fairly high. Nevertheless, every once in a while we might have a tremendously dissatisfied cus-

management substantially, thus accelerating the acceptance of management as a service.

# Beyond the SOA Forest – Universal Manageability

As a wide range of IT resources become virtualized (encapsulated and managed by Web services), the strong possibility exists that we will be able to extend active SOA management principles in both a broader and a deeper fashion. Ultimately, SOA management will be comprehensive – covering how an entire enterprise and its partners, suppliers, regulators, and customers, do business together.

This is really the central promise of the ondemand and Grid computing standards and solutions that are coming together. It would not be possible without management. Management is the cornerstone of any such environment.

So it's not useful to ask who will do the cooking in a service-oriented architecture. Everyone who needs to do the cooking will do so. The more important issue is how we will ensure that any cooking that needs to be done will be performed in a fashion that is truly adequate to our needs. And for that you need management.

# Ultimately, SOA management will be comprehensive

world of heterogeneous SOAs, including a new mechanism to build new types of WSDM agents. But CA also integrated WSDM with it's Unicenter brand, which offers comprehensive IT infrastructure management, wireless management, and so on. To do otherwise would be to leave blind spots and unmanaged areas in the SOA that would be bound to increase risk, perhaps severely. Other leading management vendors are already following suit – many beginning to integrate third-party security and wireless solutions that will likely result in more solutions in the new area of SOA management.

# Managing the SOA Forest – Management as a Service

The trend toward SOAs begs the interesting idea of making SOA management available as a subscriber service. Truly, for at least some applications, this could be an appealing option. Significant information could be obtained about a service remotely without the need to install any agent at all. Since SOAs are message based, metrics about availability could certainly be obtained and analyzed statistically based on messages sent remotely by clients that are monitored by an SOA management system. These messages could be sent periodically over widely spaced intervals so as not

tomer. Over time, that can add up to many angry customers for a service that still, on average, appears to be performing adequately.

This remote assessment of a service's availability and consistency could have some unique benefits. A neutral management authority could be used to provide outside service customers independent assurance that a service being provided is meeting availability and consistency levels. Also, such a service could be used to compare functionally equivalent services against one another – sort of like a rating service. CA has announced such a service, the Web Services Performance Index (see Resources). Other leading vendors have already publicly stated their commitment to the idea of SOAs and to the importance of services in general.

Ultimately, such management services could be used to remotely communicate with deployed agents so that deeper analysis, alert notification, and other functions requiring an agent can be administered in an outsourced fashion. Such solutions might be particularly appealing to small and medium-size companies that do not wish to administer and deploy an SOA management solution. The upcoming OASIS WSDM standard (see Resources) will provide a huge boost to this by providing common management interfaces that will simplify the task of remote service

### Resources

- Lipton, Paul. Web Services Journal, March 2004 (Vol. 4, issue 3). "Snow White's FIRST Web Services (A Cautionary Fable for IT Management)." www.sys-con.com/story/? storyid=43956&DE=1
- Lipton, Paul. Web Services Journal, February 2004 (Vol. 4, issue 2). "Composition and Management of Web Services." www.syscon.com/story/?storyid=43567&DE=1
- OASIS WSDM standard: www.oasisopen.org/committees/tc\_home.php?wg\_abbre v=wsdm
- CAWSDM: www.ca.com/webservices
- CA Web Services Performance Index: www.ca.com/webservices/perfindex.

## About the Author

Paul Lipton is a senior architect and technology strategist in the Office of the CTO at Computer Associates. An architect and developer of enterprise systems for more than 20 years, he has worked closely with key CA customers to solve important business challenges through the creation of mission-critical distributed solutions. He has represented CA in numerous standards organizations and is currently serving on standards committees involved in the definition of new Web services standards for management, orchestration, and choreography. He is a highly sought-after author and conference speaker.

■ ■ paul.lipton@ca.com

16 August 2004 WWW. WS j 2. CO m



# Can you see it?



- 1. Car notifies owner for service.
- 2. Factory informed immediately.
- 3. Supplier ships parts instantly.
- 4. Dealer inventory adjusts in real time.
- 5. Car back on road in no time.

like DB2° and WebSphere° that can connect your business end-to-end. You can build upon your existing applications, creating a responsive environment where partners, vendors and customers can easily work together. And IBM software is based entirely on open standards for maximum efficiency, flexibility, speed and agility. That's ON DEMAND BUSINESS.

See how middleware works. See how it helps businesses succeed. See it at ibm.com/middleware/integrate

# **Beyond XML Firewalling**

# Security coordination for Web services

■ Traditional development produces applications that are closed to wide usage. Custom development is required to open these programs to wide-scale integration. In contrast, Web services applications are by default open to other systems and additional configuration is required to block access.

# The Challenge of Web **Services Security**

A growing share of the Internet marketplace is being turned over to Web services. Studies have shown that by the vear 2006 a full 25% of all network traffic will make use of XMLbased Web services. As with every new technology, the increased convenience comes with new security considerations. The challenges involved in the deployment of Web services must be overcome efficiently to keep business transactions secure and reliable.

The nature of Web services is such that transactions may span multiple systems with various security domains, identity systems, and many transport segments. Confidential data must be kept secure from external threats and authorized access while enabling access by and communication with trusted client applications. XML firewalling provides a portion of this functionality. Like their network-level analog, XML firewalls allow security administrators to define security policies for inbound messages but at an XML application level. However, they don't address the problem of equipping trusted client applications with the information necessary to enable access and communication.

Web services require a security context managed, coordinated, and propagated within a distributed architecture spread across multiple systems involving several trust domains that come together in a just-in-time format. While appearing untenable, these security coordination requirements are not particularly extraordinary. They simply need to be



WRITTEN BY **TOUFIC BOUBEZ** 

approached appropriately to overcome the unique bi-directional challenges of application integration and Web services more specifically.

# Transport-Level Security Is Not Enough

Systems using SSL or VPN technology are used to provide online security. Creating a secure pipeline protects data effectively from one point to the next in

transit. However, the message is vulnerable at each stop from the source to the final destination.

Transport-security implementations are excellent at providing just that, security during the transmission portion of an online transaction. The transactions in a Web services implementation use a spanning architecture to bridge technologies. Often a deployment will make use of multiple applications, traverse diverse transport mechanisms, and cross between "identity" and "trust" domains. In each of these scenarios there is a gap in the security of SSL or VPN solutions.

The security provided by a transport-level implementation ignores the content of the message itself. Using XML there are many ways to attempt to pass an invalid message. The message must undergo XML schema validation, a multipart message integrity check, address normalization, two-way authentication, and other processes in order to ensure that only proper messages are transmitted. This is important because, for example, a properly compromised XML schema can result in a denial-of-service from the provider. Authentication is required to ensure that the

data in any section of the message is from a trusted source. If the data that is sent is invalid or malicious, protecting that data against attack en route is ineffective. Effective security must perform message-level checks to deal with these and similar difficulties.

SSL and VPN implementations have their niche and provide good security for certain applications. Transport-level security protects data during transmission, but providing Web services is much more than just transmitting information. With all the hazards that the open architecture of Web services encompasses, a different approach is necessary to provide good security.

# **XML Firewalls: A Starting Point** for XML Security

Because of the unique hazards, challenges, and audit requirements represented by communicating XML formatted messages, Web services technology vendors have borrowed a page from the Web world and created XML firewalls. XML firewalls are designed to examine the XML content of the incoming traffic, understand the content, and based on that understanding, take an action – for example, routing the message to the appropriate end point or blocking it entirely.

XML firewalls typically work by examining SOAP message headers. The header may have detailed information put there specifically for the firewall to examine, or might have information about the recipients of the message, about security of the overall message, or about the intermediaries through which the message has passed. In addition, XML firewalls can look into the body of the message itself and examine it down to the tag level. It can tell if a message is an authorized one, or if it originated with an authorized recipient. XML firewalls can also provide authentication, decryption, and realtime monitoring and reporting.

Although the firewall takes care of enforcement, it is still up to the firewall's administrators to define rules or policies that describe when to accept messages, when to reject them, and what, if any, other operations to perform. These policies are typically written from the perspective of blocking all access to a protected service unless the messages conform to the policy. This is a key point since although the XML firewall can remove the need to program security logic into the application it is protecting, it does nothing to address applications attempting to access those protected services.

www.wsi2.com **18** August 2004

In other words, any application attempting to access Web services on the other side of the XML firewall must be programmed to ensure that its messages and credentials will conform to the policy in place. If not, they will be blocked. While this approach could conceivably provide excellent protection for the firewall owner, there are significant security or administrative compromises that result.

# The Need for a New Kind of Security

The security provided for Web systems incorporates many of the current security concepts, but these need to be adapted to be appropriate for the environment. Naturally the system needs to be secure, which means dealing with the unique strengths and new weaknesses that are a part of Web services schemes. On top of the need for solid protection there is the demand for adaptability and flexibility to accommodate change and new developments in the technology.

By design, Web services provide a customizable range of security. The message has to contain the security context in order to be readable by the implementation at the receiving end. The security at each hop in a trans-

mission route needs to be coordinated to maintain message privacy while enabling successful delivery. Using the flexibility of XML in a Web services implementation brings with it distinct security threats. The additional extensibility of the language makes it more prone to malicious abuse by a clever attacker. As a result, a good implementation needs to make use of message-level auditing to ensure that the messages that are passed are sound.

Interoperability between standards-compliant implementations is key to an effective Web services deployment. However, as a result of the newness of the technology, the standards are still evolving. Consequently the security implementation needs to be easily adaptable to be able to accommodate the changes required by those maturing industry standards.

In addition to external interoperability, the security technology used must also accommodate the existence of prior applications. A successful solution fits with the existing enterprise security infrastructure with respect to techniques such as single sign—on, PKI identification protocols, or external policy stores, performance monitoring, and load balancing systems.

Another feature of Web services is the justin-time approach to business processes. Security has to be implemented and managed flexibly to facilitate this time-sensitive form of integration. The time-consuming reconfiguration of a hard-coded system can greatly impinge upon response time to a change in trust relationships or standards compliance.

To meet the changing needs of Web services, a good security solution needs to be conveniently manageable, flexible, and extensible. In order to be a useful implementation the policies governing the security protocols need to be customizable to the specific challenges of a particular service. Those policies also need to be coordinated on both sides of the integration to accommodate changing business relationships and the individual security requirements that go along with them. All of this points to the fact that Web services needs a new approach to security, one tailored to the particular needs and strengths of the technology.

# **Beyond XML Firewalling**

Web services transactions can span multiple security and identity domains. This



Offering leading software, services, and hardware vendors an opportunity to speak to over 100,000 purchasing decision makers about their products, the enterprise IT marketplace, and emerging trends critical to developers, programmers, and IT management

# Reach Over 100,000

Enterprise Development Managers & Decision Makers with IT Solutions Guide

Don't Miss Your Opportunity to Be a Part of the Next Issue!

Get Listed as a Top 20 Solutions Provider

For Advertising Details Call 201 802-3021 Today!

Which all administration are a feel broadward a district and retire relief



The World's Leading i-Technology Publisher.

W W W . W S j 2 . ( 0 M August 2004

requires that a Web services security solution contain technology to restrict access to exposed services from external threats and unauthorized access. However, threat protection and access control, while necessary, are not sufficient for Web services security. Since interacting Web services must agree on integration parameters before the first XML message is exchanged, there is also a need for coordination of security parameters. For VPN technology this coordination of security parameters happens automatically through in-band negotiation between VPN clients and VPN Gateways. On the Web, a similar kind of negotiation also happens automatically in the background between the browser and Web server. This way, crypto parameters like key

acceptable by the receiving application; how SOAP messages are sequenced to avoid replay attack; whether message exchanges are to be time stamped and reconciled to provide nonrepudiation; what CA will provide the signing authority for a SAML identity assertion; do certain message fields or parameters need to masked, encrypted, or translated before leaving a security domain for regulatory reasons; are SSO tokens required; what version of WS-S will the applications conform to; is WS-SecureConversation required for the transaction; and many others.

Because of the open-ended, extensible nature of Web services, this list of security options is quite long. Unfortunately, today for coordination of security preferences between systems participating in Web services transactions. Ideally this coordination is dynamic so that changes on one or more systems can be automatically accommodated without developer involvement.

One possible model would comprise an XML firewall plus a client-side technology for distributing security workload to and coordinating security preferences with client systems. Like VPN security, the client-side technology should be either software or hardware depending on provisioning preferences. There should also be a purely developer-oriented option for customers uncomfortable with any client footprint. Besides coordinating policy, the client-side technology should provide other value functions for a Web services transaction like SSO integration, PKI provisioning, and identity bridging. The service-side and client-side components of this architecture could then be coordinated by exchanging what amounts to a policy document outlining their security preferences, terms, and conditions for the transaction. That way changes in policy in one system get automatically negotiated with the others, preserving loose coupling.

In combination with an XML firewall this kind of client component can provide organizations with a security model that can span transactions both inside and outside the traditional corporate security boundary. From a cost and time-to-market perspective, a client-side element that can negotiate on the fly with an XML firewall saves considerable developer effort and time. It also removes the usual error and consistency risk inherent in a programmed model of security provisioning. While this kind of two-way security model is not universally appropriate, it can prove extremely beneficial in many Web service integration scenarios.

# By design, Web services provide a customizable range of security

"

length, cipher type, credential preferences, and protocol versions get negotiated automatically at the beginning of each session without any programmer involvement on the client or server end. There is no analog to this kind of capability for Web services today.

XML firewalls alone don't address the problem of client-side security coordination. They focus on blocking functions for XML messages entering a restricted security zone. But like VPN and the Web before it, systems exposed as Web services that are to exchange data or be integrated must first have their security preferences aligned consistently. This challenge is even more acute for Web services than for VPN or the Web since the number of security permutations available at an XML message level, far outnumbers the security permutations available when negotiating IPSec VPN or SSL Web communications.

Possible considerations include credential choices and how they are to be passed; where a signature needs to be added inside a message body; what part of the message needs to be encrypted; whether PKI certs are required; how they are provisioned; what schemas are

the only way for a sending application, i.e. a Web service, to communicate its security capabilities and expectations to a client application is by having the sending application's developer communicate this outof-band with the receiving application's developer using e-mail or the phone. WSDL is silent on the subject of security. Even when WSDL starts to accommodate security descriptions, it is a static API protocol. Changes in security requirements and expectations can't be automatically updated on the client system without considerable developer effort. So even a small task like upgrading a Web service to a new version of WS-Security becomes a difficult coordination, development, and testing process.

# A Web Services Security Coordination Model

XML firewalls provide a critical element in a broader Web services security strategy. But XML firewalls alone are not sufficient given the "integrated" nature of Web service transactions. There is also a requirement

# About the Author

Toufic Boubez is the CTO of Layer 7 Technologies. Prior to joining Layer 7, Toufic was CTO at Saffron Technologies, where he drove the company's distributed machine learning and knowledge management initiative for the U.S. military and government. Prior to Saffron, he was chief architect of IBM's Web services initiatives. He was also IBM's technical representative to the UDDI Web services consortium with Microsoft and Ariba (www.uddi.org) and a coauthor of the UDDI API spec.

■■ tboubez@layer7-tech.com

**20** August 2004 WWW.WSj2.com



Imagine XML-enabling your existing PowerBuilder and Visual Basic applications

QUICKIn hours, without changing a line of code

CHEAPInexpensively, using your current skills

SAFESafely, preserving all logic and security

# **NOW YOU CAN**

www.active-endpoints.com

# Planning for Service Management Within a Service-Oriented Computing Infrastructure

A necessary infrastructure for mission—critical deployment

■ The more widely service-oriented architectures become incorporated into core business applications and processes, the more critical the ability to easily configure, manage, and monitor the overall infrastructure becomes.

y their very nature, service-oriented architectures, or SOAs, are about enabling heterogenous, componentized, and distributed applications to work together seamlessly. This presents a number of classic service management challenges such as auditing and logging, security, quality of service guarantees, service-level agreements, service life cycle, and service virtualization.

This article examines SOA management features that will become critical as services are widely deployed in mission-critical environments. In addition, I describe the common architectural approaches used to integrate WSM into a service-oriented architecture, as well as emerging standards and their likely impact and utility.

# The Purpose of Management

The more widely Web services are adopted and integrated into the core business of enter-

prises, the more invisible the structure and performance of these business assets becomes. An enterprise cannot wait until it receives customer complaints or observes decreased revenue before it reacts to problems in its Web

service application deployments. In addition, the more granular and distributed the components from which an enterprise's business is derived, the higher the potential risk and cost of any individual component failing or changing. It is management's role to expose and monitor running applications so that problems can be detected as early as possible—and corrected before they become catastrophic.

It's also clear that nearly all Web service deployments require a common set of capabilities (such as security and monitoring) in order for them to have true enterprise-quality reliability. These capabilities are often complex and, therefore, prone to error. Thus, Web service developers should not be forced to re-

implement them with every new Web service; these capabilities should be provided by the underlying infrastructure.

With this in mind, the aims of Web services management are manyfold. Its goals are to make the structure and performance of Web services applications more visible; mitigate the risks associated with SOAs as they're deployed and as they change over time; and provide a common set of capabilities that let Web service developers focus on implementing business logic as opposed to infrastructure.

# **Key Features of SOA Management Monitoring**

The classic role of a management system is to monitor. The system collects information about the health and performance of the system and notifies the operator when something is wrong. Web service applications provide a unique opportunity due to their loosely coupled and open nature. They make it possible to add meaningful instrumentation without modifying the application code by intercepting the Web service request and response messages. This interception can help generate a clear picture of system performance at the application level.

A Web services management system should provide the following monitoring services:

**22** August 2004 WWW.WSj2.com

- Performance monitoring: The ability to record the amount of time required to accomplish various system operations, such as response time. When considering performance monitoring, note the cost or overhead incurred to perform and record the measurement.
- Resource monitoring: The ability to monitor and record the usage of various consumable system resources under load, such as memory use or concurrent request count.
- Availability monitoring: The ability to determine when the application is available for use. Ideally, this monitoring will occur at numerous levels to support root cause diagnosis.
- Fault monitoring: The ability to recognize and notify an operator when an application component has failed during request processing.
- Notification: The ability to alert an operator to a problem that was discovered as a result of monitoring. Notification can be as simple as e-mail or as complex as custom integration with a third-party network management system (NMS).
- Probing: An active management component that initiates synthetic requests to trigger performance and availability monitoring. This lets the system manager discover problems before users encounter or report them.

# Service-Level Agreements

A service-level objective (SLO) is a proposed acceptable range of a single verifiable measurement -such as request processing time - that's important to the consumer. A network consumer might indicate that it's never acceptable for the request processing time to exceed 30 milliseconds. These objectives can be specified in complex ways. For example, an SLO might state that request processing time not exceed 30 milliseconds for requests with less than 100 data elements. A service-level agreement (SLA) is a collection of SLOs agreed upon by a service provider and a consumer. SLAs typically protect the consumer but can also protect the service provider. A provider may want to constrain a consumer's access to a service so it can meet other objectives in the agreement. For example, a news content provider may allow a consumer to poll new stories only once a day. This facet of SLAs is inexorably related to the concept of quality of service.

### Root Cause Analysis

One of the primary goals of an SOA management system is to provide the ability to diagnose and correct problems. It's the role of monitoring to determine that there is or soon will be a problem. Beyond that, the management system should offer tools to narrow down the cause of the problem.

The nature of distributed, collaborative, loosely coupled services can make this difficult. Services interact independently of human intervention and such interactions

service's location changes due to network reorganization or hardware changes.

#### Security

When dealing with security, the service delivery platform and the management infrastructure must cooperate closely. At times there can almost be no distinction made between which part of the system provides the functionality. It's important that the combination of platform and management system should be able to provide the

# ...nearly all Web service deployments require a common set of capabilities....



often occur between separate organizational units or companies. To help diagnose problems, the management system must provide some way to represent the dependencies and interactions between Web services. The ability to discover these relationships through passive monitoring can be a plus when relationships between consumers and providers are established dynamically. The result is a repository of metadata that describes these relationships and can be used to correlate and order collections of failures.

# Virtualization

Virtualization is an umbrella category of a set of capabilities that are primarily concerned with insulating service consumers from change and providing service providers with implementation and deployment flexibility. Chief among these capabilities are transformation and routing. Transformation lets users adjust for minor changes in a ser vices interface so that older interface versions can be maintained in the virtualization layer. When a request is received with the older format, a transformation is applied to make the request conform with the new interface. In this way, interfaces can be moved forward without requiring clients to upgrade immediately. Routing allows messages to be redirected when a destination

required security features. It's also critical that security decisions be made as late as possible in the development deployment cycle and that a great deal of flexibility exists when these decisions are made. To this end, the combined platform and management system should provide the following features:

- Auditing: The ability to record in a secure and robust manner any access to deployed services.
- Authentication: The ability to validate the identity of intended service consumers. The service developer should not be responsible for specifying who these entities are and how their credentials will be provided and verified.
- Authorization: The ability to control which services and operations are available to a given entity based on a user-specified security policy. The policy should be able to consider not just identity but also access time, the consumer's network location, and other criteria associated with the request context.
- Confidentiality: The ability to hide request and response data from unauthorized viewers. For XML-based services, confidentiality should be applied at a fine-grain level, allowing encryption of selected portions of the message body as well as the entire message.

W W W . W S j 2 . C 0 m August 2004

# **Architectural Approaches**

Using Web services as the implementation technology for a service-oriented computing infrastructure offers a number of options in how to introduce management functionality.

One choice to make is whether management functionality should be integrated into the service execution platform or the service network. Management functionality is typically added to the service execution platform by the platform vendor. In this scenario, service interactions are passively monitored by the runtime, and control points are introduced into the infrastructure to facilitate management features. In contrast, integrating management into the service network can be done independently of the platform provider(s). With this approach, a proxy or intermediary is positioned in the network to intercept service requests and responses. Management capabilities are then deployed on the proxy/intermediary, which acts as a monitoring and control point (see Figure 1). Both approaches have their advantages and disadvantages.

A platform-integrated approach will typically incur less of a performance overhead. It can also allow a more comprehensive insight into message processing. Also, since integrated management capabilities are usually offered by the platform vendor, you'll usually see a broader range of control functions as well as greater consistency between the Web services management features and other systems management functions, as depicted in Figure 2. The biggest disadvantage of platform-integrated management is most evident in heterogenous environments where multiple vendors' service execution platforms are used. In these environments, system administrators have to understand multiple management tools to manage the entire network. Administrators typically want to have a consolidated view of the state of the enterprise's systems, but when integration between multiple vendors' service management systems is possible, it's difficult due to the lack of mature standards in this space. Also, some service platforms may not offer intrinsic management capabilities, in which case an add-on product is necessary.

Placing management in the network makes it easy to manage heterogenous net-

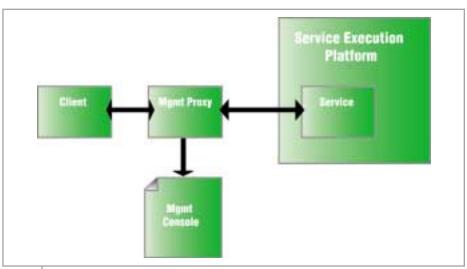


FIGURE 1 Proxy/intermediary management approach

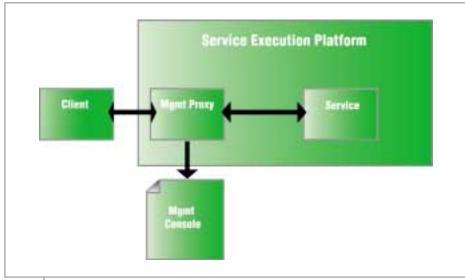


FIGURE 2 Platform-integrated management approach

works. Services from multiple platforms can be routed through the management proxy/intermediary, which also facilitates central management and configuration. The proxy/intermediary model can also be introduced into a service topology as an afterthought without requiring potentially costly upgrades to server infrastructure or applications. The proxy/intermediary approach is also particularly well suited to certain types of applications and environments. Virtualization features are typically associated with an intermediary deployment model, and organizations that broker services for third parties will also typically use this approach. The big disadvantage of the proxy approach is the performance cost of adding an additional network hop.

# Web Services Management Standards

Although there have been a handful of specifications published to address various areas of Web services management, the standards space has been relatively quiet. Currently, the OASIS Web Services Distributed Management Technical Committee (WSDM) is working on two standards that will likely have the most effect on users' perceptions and vendors' products. These specifications are the outgrowth of initial work done by member companies as well as collaboration and interactions with related standards efforts in more specialized forums such as the Globus Grid Forum and the Distributed Management Task Force.

**24** August 2004 WWW.WSj2.com

WSDM will produce two specifications: one focused on Management Using Web Services (MUWS) and the other on Management of Web Services (MOWS). The MOWS specification provides a framework for using Web services to manage general IT resources but does not specify a management information model, leaving this to particular management applications or products. This recognizes that good management models already exist, and allows MUWS to be used in conjunction with existing systems management tools. MOWS, on the other hand, builds on MUWS to provide an information model for Web services, defining the management capabilities and attributes of Web service entities such as ports and endpoints.

The WSDM specifications are currently in draft status, however, leaving developers and architects who need SOA management today to choose from a variety of solutions that offer similar capabilities but are, for the most part, not interoperable. Also, given the velocity at which draft standards become final, and at which final standards are implemented in commercial products,

it will probably be a while before we can count on standards-based SOA management being generally available.

The good news is that at this point, the standards are largely orthogonal to the decisions organizations need to make today. When WSDM is available it will define interoperable interfaces, allowing management functionality to be exposed, queried, and controlled by a broad range of tools. However, the management functionality itself will still be a function of product and platform capabilities, and this is how buying decisions should be made. Assuming the vendor you choose is committed to supporting WSDM as it matures, the migration to standards-based interfaces should be transparent to managed resources such as services and applications.

# Conclusion

SOA implementations require many features to make them robust. Service monitoring, service-level agreements, virtualization, and security are all important parts of most service implementations. Just as application servers provide a multi-user,

transactional infrastructure for applications, Web services management provides the necessary infrastructure to make Web services suitable for mission-critical deployments. Building your SOA architecture with Web services management explicitly addressed will make development time shorter, the services more maintainable, and the system more reliable and robust.

# About the Author

Bill Jones is a senior software development manager at Oracle Corporation with 15 years of industry experience. He currently leads development of Oracle's Web services technology with a focus on Web services management. Bill has a wide range of management and engineering experience including the production of high—performance middleware platforms and large CRM systems.

■ ■ bill.jones@oracle.com

Kevin Minder is a veteran software engineer with 15 years of experience. For the last two years he has been focusing on Web services management. He is currently developing SOA management solutions at Oracle Corporation. His experience ranges from application management to high—performance middleware and embedded systems development.

■ ■ kevin.minder@oracle.com



W W W . W S j 2 . C 0 M August 2004

WSJ: ANOTHER VIEW

# The Myopia of Web **Services Management**

# What's in your architecture?

Rather than taking a myopic Web services management approach to realizing the promise of shared services, enterprise architects should focus on building the architecture that controls chaos and enables sharing and reuse.

eb services, with their standards-based ability to exchange data between disparate, distributed systems, hold great promise for enterprises seeking the holy grail of enterprise computing: a fully integrated infrastructure that inexpensively adapts to business needs at enterprise speed and scale. However, for their full value to be realized, enterprise architects need to view Web services strategically or they will find themselves in a muddle of integration spaghetti.

Web services management (WSM) at first seems to provide an adequate solution for global Web services adoption, but has ended up facilitating a myopic, ad-hoc, project-based approach that encourages a focus on near-term projects instead of on long-term, strategic benefits. You cannot accomplish global interoperability and sharing if you approach it one distinct project at a time. Why? Because you cannot control the chaos inherent in purely project-driven Web services adoption. This only adds diversity and complexity to IT - creating a complex tangle of rigid point-to-point connections, numerous departmental archipelagos, and multiple policy silos - fostering enterprise incompatibility, which runs counter to the notion of SOA.

Look at some of the results of current WSM approaches on software infrastructure:

• Policies, policies everywhere. WSM tools place a policy definition tool on every developer or project manager desktop, which forces policy decisions on each individual short-term project. This places the burden of



**SAM BOONIN** 

mentation to implementation. Over the long term, this approach creates multiple policy silos, leading ultimately to a need to integrate the policy implementations themselves.

infrastructure design on disparate proj-

ect teams as opposed to on central

project-led WSM approach simple

operations staff. For instance, with a

policies for access control, logging, or

routing can be different from imple-

• Agent-based approaches add unnecessary complexity. WSM tools require users to deploy agent or proxy applications onto already-taxed application servers and EAI platforms. Being forced to add an agent or to proxy an application makes participation in the service network more difficult, raising the bar for broad-scale adoption. This approach encourages a myopic focus on the application as opposed to the service network.

• Limited scope = limited scalability. WSM solutions were built to solve relatively small problems residing in a business unit or department, and are not designed to scale across a global enterprise. Trying to scale WSM solutions across the enterprise leads to challenges, especially in unifying policy enforcement for security, routing, message transformations, etc.

With a properly constructed infrastructure, a planned service-oriented architecture (SOA) can succeed where ad-hoc WSM fails. In contrast to WSM products, there are solutions designed to create service-oriented infrastructures that feature:

- Centralized policy definition, distributed policy enforcement. This federated approach enables service-oriented architectures to appropriately balance two seemingly contradictory needs: the need for central coordination to avoid chaos, and the need to be distributed to maximize flexibility and scalability.
- Distributed networking approach. An SOA is fundamentally a distributed network. The inherent challenges - enabling reliable, consistent, and predictable communication between services deployed across a distributed global enterprise - are distributed networking challenges. Remember it was the IP router (not the server, bus or agent) that acted as the key piece of infrastructure enabling the rapid growth of the first generation of Web computing.
- Scale globally. This class of SOAs can handle millions of messages each day out across thousands of applications and shared services. They interconnect networks of services, from across the enterprise and around the world, all directed centrally and governed locally.

Ultimately, the goal should be an SOA infrastructure that is standards-based; enables sharing; offers loose coupling of applications; and federates control over services, applications, and processes that are, in turn, universal, reusable, flexible, and distributed. In this context, Web services management becomes a set of valuable applications and operational capabilities that leverage that shared infrastructure.

Putting it simply, enterprise architects have global challenges that require global solutions, and thus require a stable, reliable, and scalable infrastructure. If they are to make their architecture more connected to business needs, they will have to move beyond ad-hoc solutions and build a broad, enterprise-wide, coordinated SOA so existing systems can live on, interoperating seamlessly with new services that are brought on line and integrated into the SOA quickly. Ultimately it is the architecture that matters.

# About the Author

Sam Boonin is the vice president of marketing for Blue Titan Software, a provider of service-oriented infrastructure that helps enterprise architects control, share, and scale applications, driving business innovation across the distributed enterprise.

■■ shoonin@bluetitan.com

www.wsi2.com **26** August 2004

# out possibilities!

# **ScanSoft®**

# **UNISYS**

















Silver Sponsors























- >> Learn Industry Leading **Best Practices**
- >> Network with Thousands of Your Peers

**ACT NOW for Biggest Savings** and Free EXPO Pass with priority code NDJ1.

# **Registration information:**

# **Chris Nolan**

1-859-278-2223 or Toll free 1-877-993-9767 chris@amcommexpos.com



September 13-16, 2004 >> EXPO 14-15, 2004 >> New York Marriott Marquis

# **Oracle BPEL Process Manager**

# Making SOA design easy

Service-oriented architecture (SOA), while its underpinnings have been around for years, has recently become one of the most talked about topics in the IT industry. The interest in SOAs is largely due to the emergence of Web services and their ability to expose business applications through well-defined interfaces in a platform-independent manner. While an SOA need not be built using Web services, it certainly presents an attractive option for a variety of environments.

n an SOA, applications are exposed from a process-centric viewpoint. Some examples L include services that handle processing credit card transactions, capturing and executing purchase orders, and viewing and updating inventories. The challenge is collecting these independent services BRIAN BARBASH into a viable business process. Enter the Oracle BPEL Process Manager

(formerly Collaxa BPEL Server). The Collaxa BPEL Server provides an environment to design, develop, deploy, and manage the orchestration of independent applications through the use of Business Process Execution Language (BPEL).

The Oracle BPEL Process Manager is distributed with two main components: the server and the designer. The server handles the runtime execution, management and auditing of business processes. The designer, built on the Eclipse platform, provides the IDE for composing and deploying business processes.

# **Composing Business Processes**

Composing a business in the BPEL Designer is a relatively simple process. At the most basic level, the primary tasks to compose a process include choosing whether to build a synchronous or asynchronous process, establishing Partner



WRITTEN BY

Links with services to be executed, linking the services together using Assign operations, establishing any control flow and special processing requirements, and gathering the necessary results for the endpoint.

For the purposes of this review, I will be creating a simple asynchronous business process that performs the following steps:

- 1. Accept the name, postal address, and email address of a user.
- 2. Look up the current weather forecast for the user by zip code.
- 3. Download information about the address the user has provided.
- 4. Send an e-mail with the collected data to the user's provided e-mail account.

The system will incorporate publicly available Web services to get the weather forecast and address information. The email will be sent from a Java application exposed as a stateless session bean.

As mentioned earlier, the Partner Links to the weather and address services must be established. Using the Add Partner Link wizard in the IDE, WSDL files of desired services may be downloaded into the process model. Developers may enter a URL to a WSDL file or browse a UDDI server to select the appropriate service. If the WSDL file provided does not define a

partnerLinkType, the designer will automatically create a wrapper.

Once the Partner Links are established, the development of the actual flow within the business process may be done. Figure 1 shows the complete business process in the BPEL Designer. An instance of the business process is created when a SOAP message is received by the BPEL Server. The first yellow icon in the figure represents the Assign operation that extracts data from the received SOAP document and places it in the appropriate XML structures for both the Weather and Address services. The Assign operator in the BPEL Designer supports XPath queries, XPath expressions, BPEL XPath extension functions, Collaxa XPath extension functions, and custom XPath extension functions. It should be noted, however, that according to Collaxa's documentation, due to the complicated nature of developing transformation logic in the Assign operator many users of the BPEL Designer offload complex data transformation to Web services or Java applications that utilize XSLT and/or XQuery.

The next group of icons illustrates the parallel processing capabilities of the BPEL server. Since the weather and address services are independent of each other, they may be run simultaneously. After the Assign operation has mapped the SOAP message to the appropriate inputs for each service, they are executed. Any remaining tasks within the business process are put on hold until all elements within a parallel processing group have completed.

# ORACLE

# Company Info

Oracle Corp.

500 Oracle Parkway

Redwood Shores, CA 94065

1-800-ORACLE1

United States Support Sales Phone:

1-800-833-3536

At the time of this writing, Collaxa, Inc., which originated the Collaxa BPEL Server, was a separate firm. It has since been purchased by Oracle, and the Collaxa BPEL Server is now known as the Oracle BPEL Process Manager.

**28** August 2004 www.wsi2.com Once the weather and address services have returned, the e-mail service is executed. Since the e-mail service is exposed in the form of a stateless session Enterprise JavaBean, the Exec function is used. The Exec function is part of the namespace that defines the Collaxa-specific BPEL extensions. This tag allows the developer to write the Java code necessary to execute the e-mail service directly into the BPEL document.

Finally, the last Assign operator places a hard-coded success message in the response payload of the business process, which in turn is sent to the calling process via an asynchronous callback. (*Note:* The result of either of the Web service calls can just as easily be returned to the client).

# **Advanced Business Processes**

In real-world scenarios, business processes typically include complicated logic for handling exceptions and gathering input from users. To that end, Oracle's BPEL Process Manager provides mechanisms to handle both. Notice that in Figure 1 there are two vertical labels - one blue (indicating it has the current focus) and the other gray. These labels and the boxes to which they are attached represent the BPEL Designer concept of Scope. Scope is analogous to a general code block prevalent in most programming languages and specifically related to the try/catch structure in Java. Business process logic may be attached to Scope definitions to handle message faults, perform compensating transactions, execute logic

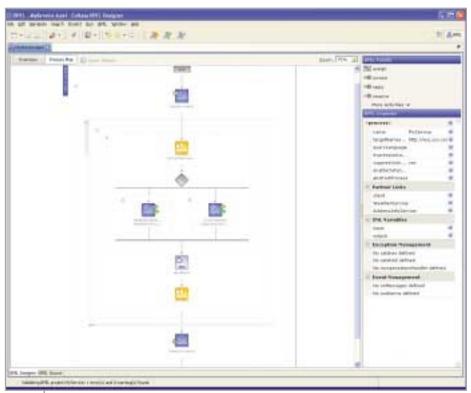


FIGURE 1 BPEL Designer with Business Process Map

The Oracle BPEL Process Manager also contains a User Task element. A User Task is a service that assists with modeling processes that require user input in order to complete. It is accessible externally through a Java API upon which custom interfaces may be developed. Typically in a process that requires user input, an upstream process element creates a User

nal process, the User Task API is called to assign a status of complete (or any other appropriate value). This in turn activates the remaining downstream processes.

# **Summary**

As the development of applications continues to shift toward the process-centric viewpoint, the need to easily piece these components together to form a larger business process increases. Oracle's BPEL Process Manager is a very strong option for meeting this need. The designer provides the tools necessary to create complex business processes while the server delivers excellent management and auditing capabilities. Overall, the Collaxa BPEL Server is a very capable product and should be considered when building service-oriented architectures.

# About the Author

Brian R. Barbash is the product review editor for *Web Services Journal*. He is a senior consultant and technical architect for the Envision Consulting Group, a management consulting company focusing on contracting, pricing, and account management in the pharmaceutical industry.

■■ bbarbash@sys-con.com

# The challenge is collecting these independent services into a viable business process....

when messages are received, and execute logic when a long-running process reaches a predetermined timeout. This allows for the graceful recovery from errors and provides additional monitoring capabilities by responding to important events.

Task and associates a payload of data with it; subsequently suspending downstream process nodes. That task and its associated data may then be accessed through an external application's user interface. Once the user has completed the required exter-

W W W . W S j 2 . C 0 M August 2004

# Web Services and **Federated Identity**

# Standards to better define a space

■ Since the advent of Web services, and other distributed computing standards for that matter,

we've been wrestling with the notion of identity and how to manage it.

ruth-be-told identity management has been put on the back burner as organizations attempt to get their first Web services projects up and running. However, as Web services become more pervasive, this is an issue that is getting more attention.

With the increased interest in identity management so too has risen the need for standards to better define this space. These standards all aim to organization together into a unified whole, allowing for everyone to be known to every-

notion of federated identity management.

LINTHICUM bind identity management systems within an one else, securely. To that point, let's examine the emerging standards, along with the

# Who Are You?

So, why do we need identity management? Web services are not for internal use only anymore, and those who leverage Web services (consumers), or produce Web services (providers), need to be known to each other, else we risk invoking malicious or incorrect behavior, which could cost us dearly. This is clearly the case within trading communities that leverage Web services. Many outside organizations are binding to your services and you to theirs, and the potential for disaster increases, unless you know just who you're dealing with.

Identity is important in the growth of sensitive data and confidential relationships online. Lacking identities, there is no way to provide certain users with access to certain resources.



DAVID S.

Today, we use managed identities, including different user names, passwords, and other identifying attributes. The same person may have links to many organizations, including frequent flyer sites, banking sites, employee benefit sites, etc. Perhaps you have a list of user names and passwords in your drawer today.

The number of identities that we have

creates a challenge. We've all written down user IDs and passwords on sticky notes just to remember them. Moreover, IT organizations find it increasingly difficult to manage the profusion of identity databases, even within their own organizations. The problem becomes more of an issue as we extend our reach outside of the firewall, between organizations. Enter federated identity and a potential solution to this problem.

Federated identity, including supporting standards such as those from OASIS and the Liberty Alliance project, is a defining mechanism that organizations may employ to share identity information between domains. While most understand the value of an identity management system internal to an enterprise, federated identity presents a new set of problems, and an opportunity for solutions.

There are many benefits to employing federated identity solutions, including the ability to perform logging and audit functions centrally, cost reductions associated with password reset, and access to many existing heterogeneous application securely.

# Standards and Identity

In order to support the notion of federated identity you need a loosely coupled architecture that allows for the exchange of identity information in and between entities. Thus, we must all get on the same channel as far as interfaces, messaging, security, and content standards, or we have no hope of solving this problem. There are three contenders:

- Oasis and SAML
- Microsoft, IBM, and the WS-Roadmap
- · Liberty Alliance

# Security Assertion Markup Language (SAML)

SAML is an XML framework for exchanging security information over the Internet and enables disparate security systems to interoperate using a single security mechanism, thus providing federated identity management. SAML resides within a system's security mechanisms to enable exchange of identity and entitlement with other services. It defines the structure of the documents that transport security information among services.

SAML has the following components:

- Assertions and request/response protocols
- Bindings (the SOAP-over-HTTP method of transporting SAML requests and respons-
- Profiles (for embedding and extracting SAML assertions in a framework or proto-
- Security considerations while using SAML (highly recommended reading)
- Conformance guidelines and a test suite
- · Use cases and requirements

SAML provides technology that supports a single sign-on using XML. Using SAML authentication, you can sign-on and receive a SAML authentication assertion as a response to the request. This authentication assertion is simple XML and is transportable using SOAP.

# WS-Roadmap

This is really just a white paper published by IBM and Microsoft outlining a roadmap for building a set of Web services security specifications. WS-Security was the first specification they published.

The WS-Security specification proposes a standard set of SOAP extensions that can be leveraged when building secure Web services to implement confidentiality, or the ability to leverage Web services without having to worry about others getting into your business.

WS-Security is designed as the base for

the construction of a wide variety of security models, which include:

- PKI
- Kerberos
- SSL

Moreover, WS-Security provides support for multiple security tokens, multiple trust domains, multiple signature formats, and multiple encryption technologies. This standard defines three main mechanisms:

- · Security token propagation
- Message integrity
- · Message confidentiality

Each of these technologies does not provide a complete security solution; WS-Security is a building block that can be used in conjunction with other Web services extensions and higher-level application-specific protocols to leverage a wide range of security and encryption technologies. You may use these independently (e.g., to pass a security token) or tightly integrated - for example, signing and encrypting a message and providing a security token hierarchy associated with the keys used for signing and encryption.

The importance of leveraging this standard in the world of application integration is obvious: we seek ways to exchange messages between enterprises with the assurance that those outside the trading partners won't have access to them. The support for multiple security standards is an added value as well, considering the number of organizations that may be involved and the diverse security technologies that may be in place.

# Liberty Alliance

The Liberty Alliance is really a consortium of about 170 companies that built a specification for federated identity management. The idea, at first, was to create a comprehensive federated identity specification. However, last year they also released a new blueprint describing three specifications. You can leverage the specifications together, or separately.

They include:

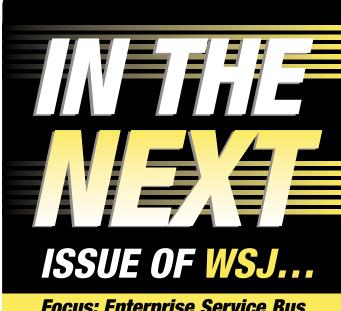
- Identity Federation Framework (ID-FF): Allows single sign-on and account linking between entities with pre-established relationships
- Identity Web Services Framework (ID-WSF): Allows groups of trusted partners to link to other groups, providing control over how their information is shared
- Identity Services Interface Specifications (ID-SIS): Builds a set of interoperable services on top of the ID-WSF specification

As we move forward with service-oriented architectures (SOAs), and learn to extend them beyond the bounds of our firewalls, the need for identity management technology will increase. Security is sometimes an afterthought when building an SOA internally, but those looking to extend their SOA outside of the firewall are seeing the need now.

### About the Author

Dave Linthicum is the CTO of Grand Central Communications (www.grandcentral.com) and has held key technology management roles with a number of organizations including CTO of both Mercator and SAGA Software. David has authored or co-authored ten books, including the groundbreaking and best-selling Enterprise Application Integration released in 1998. His latest book, Next Generation Application Integration, From Simple Information to Web Services was just released.

■ ■ e-mail: linthicum@att.nel



# Focus: Enterprise Service Bus

# SB Integration Patterns

An ESB is a standards-based integration platform that combines messaging, Web services, data transformation, and intelligent routing to reliably connect and coordinate the interaction of significant numbers of diverse applications across extended enterprises with transactional integrity. This article looks at some of the design patterns available for use in creating your ESBs.

# SB: From Design To Implementation

With the large-scale adoption of SOA and Web services reflecting enterprises' ambitions to become on-demand enabled, almost inevitably there arises the issue of an ESB. The highly distributed nature of the services paradigm coupled with a focus on flexible, real-time end-to-end business process management and deployment requires a highly solidified and capable middleware platform solution. Increasingly, the notion of an ESB is commonly perceived as the appropriate solution approach.

# SOA and the Art of Riding the Enterprise Service Bus

The ESB plays a key role in making SOA real. This article will give you an overview of the main characteristics of an ESB and how to use it to realize SOA-oriented component interaction patterns: In particular we will discuss universality, support for service orientation, open APIs and protocols supporting interoperability, and qualities of service.

# Plus...

# Web Services and Secure Voting

Secure voting is in the news. With the dangling chads of the Florida contest still fresh on our minds, we are preparing for another presidential election. This article describes a programming project assigned to a class in applied cryptography at Carnegie Mellon University. Starting from a secure voting protocol, the students were asked to build a secure voting Web service using Sun's JWSDP and Apache's Axis.





www.wsi2.com August 2004

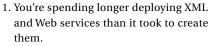
# Six Tips for Moving Web Services from the Lab into Action

# Flexibility, security, and manageability are the watchwords

■ It's relatively easy to build custom Web services. Customers are finding that it's much more difficult, however, to successfully secure and scale them in production. Six tips from Canada's largest loyalty reward program provider, a global financial services company, and Reactivity, Inc. can help you simplify Web services deployment.

# They're Popping Up Everywhere

For one or two XML Web services, why worry about systematizing deployment? The answer is so you don't have to worry about them – or the potential security and availability lapses they can create – later. Here are three indications that your enterprise needs a robust, secure XML Web services deployment system:



- 2. You're going back to modify services to accommodate new standards, meet security requirements, and fix interoperability problems.
- Stealth Web services are popping up around the company and you have no way to efficiently monitor them, consistently enforce security, or even ensure their availability.

If your XML Web services are designed to improve enterprise revenue or processes, you must seriously consider how they are



JOELLE GROPPER KAUFMAN

deployed and maintained. Here are six lessons learned by one of the world's largest financial services firms and Aeroplan, Canada's largest loyalty reward program provider.

# Lesson 1 – Minimize Web Services Interoperability Issues from the Beginning

Even the most specific Web services standards are subject to imple-

mentation interpretation and variability. For example, almost every implementation of Web Services Description Language (WSDL) is unique. This means that associated WSDLs must be easily importable by any client and, once deployed, they must be able to mediate between heterogeneous services without requiring business partners to alter their systems.

In Aeroplan's case, it chose to leverage existing XML services over an MQ Series communication channel that was already integrated with its mainframe. However, rewards partners would not communicate with them using XML over MQ. In fact, Aeroplan's first partner communicated using SOAP over secure HTTP (HTTPS), based on Visual Basic 6.0. Aeroplan needed

an infrastructure that could proxy for its mainframe, create a secure connection, ensure transaction security, and translate from SOAP/HTTPS to XML/MQ and back again (see Figure 1).

As Aeroplan added partners, it encountered different flavors of SOAP and various security standards running on platforms ranging from VB and .NET, to WebSphere, to Axis (see Figure 2).

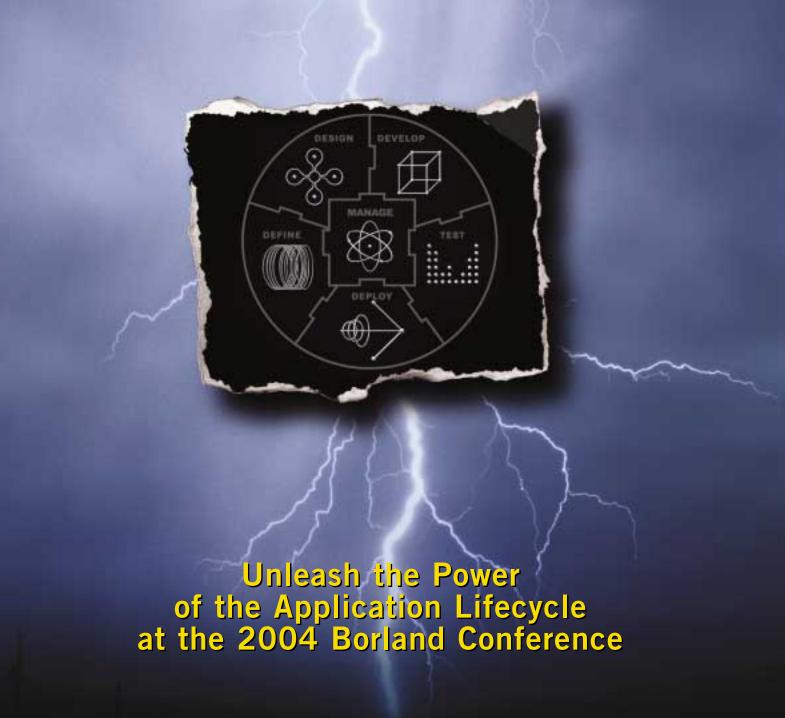
Aeroplan's new deployment infrastructure insulates it from this variability and greatly improves the IT team's productivity when setting up new connections. Plan to accommodate a variety of platforms, standards selections and implementations, and transports; and look for a deployment infrastructure that performs comprehensive mediation as part of its primary purpose – rather than building each XML or Web service to mediate the wide range of standards and transports you'll encounter.

# Lesson 2 – Realize that Standards Are Not Enough for Security

Web services can be reliable, secure, and cost-effective. However, it's important to realize that standards-only based Web services security measures are not enough.

- **Evaluate relevance:** Support the security standards that are most reliable and that deliver the biggest benefits to you, your partners, and customers. For example, ensure that your applications efficiently support one-way and two-way Secure Sockets Layer (SSL) and that support tools minimize opportunities for operator error. Be ready to support WS-Security, because it is the foundation of many high-utility upcoming standards. It's smart to support XML Encryption and XML Digital Signature so that your infrastructure can accommodate partners who just use XML as opposed to SOAP. Finally, consider Security Assertions Markup Language (SAML). Many enterprises are finding many uses for sharing security and authentication information between applications, and your infrastructure should support the current SAML standard and have a roadmap to support the standard in development.
- Look beyond SSL: While SSL secures
  information in transit and is highly interoperable, it requires considerable private
  key/certificate management that the Web
  service must handle. There are no guaran-

**32** August 2004 WWW.WSj2.com



**The Borland Conference** is a premier event for technical education, focusing on all the technologies impacting software development. With more than 200 technical sessions, you will see how you can facilitate teamwork, enhance productivity, improve quality, reduce costs, cut maintenance time, and accelerate business flexibility and success. Learn how the entire development team can create and deploy better software, faster – with the integrated Borland suite of products for the analyst, architect, developer, tester, deployment group, and manager.

Special discount of up to 50% on select Borland products • Exhibit hall • Free conference proceedings CD • Product Solution tracks covering all Borland products, including JBuilder, Delphi, Together, StarTeam, CaliberRM, C#Builder, C++BuilderX, Optimizeit ServerTrace, Borland Enterprise Server, Janeva, and InterBase • Interest Area tracks, including ALM, Methods, and Processes; Architecture, Models, and Patterns; Microsoft NET Framework; J2EE; SOA; emerging technologies; and more!

September 11-15, 2004 • San Jose, California

For complete conference details and session information: connect.borland.com/borcon04

Made in Borland® Copyright ® 2004 Borland Software Corporation. All rights reserved. Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. All Borland brand and product names are trademarks or registered trademarks of Borland Software Corporation in the United States and other countries. ◆ 22127.4



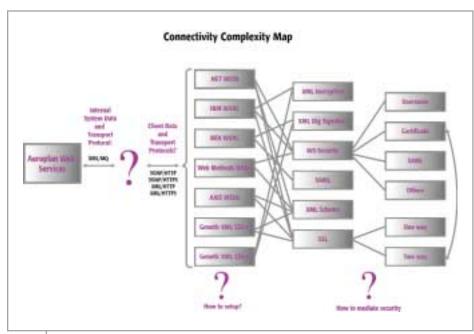


FIGURE 1 | Aeroplan's existing infrastructure

tees of message confidentiality or integrity, no auditable records of the message, and session security can't be enforced. In addition, security threats in the XML payload remain invisible until they are decoded – potentially within the targeted application server where they could wreak havoc.

- Consider new standards: Consider XML digital signatures and field-level XML encryption with WS-Security – newer standards that provide a better way to keep messages confidential and assure their integrity.
- Dedicate a deployment infrastructure: Be sure to include centralized security policy control and distributed enforcement for securing Web services transactions.
- Test, test, test: Rigorously and regularly test standards for interoperability. They continue to evolve and their implementations vary from solution to solution.

# Lesson 3 – Implement Scalable Security and XML Threat Defense from the Start

Plan your defense against XML threats before an attack. Web services can create new vulnerabilities and new threats continue to emerge. Ensure that your architecture minimizes the chances of an attack reaching your services by intercepting and eliminating questionable messages at the network

edge. The main Web services threats fall into three categories, and to beat them, you'll need a deployment infrastructure that is flexible and can scale quickly (see Table 1).

- Trust infrastructure integration: Leverage existing security infrastructure, such as directories, PKI, and Identity Management solutions, to ensure consistent trust relationships and to speed Web services deployment.
- Centralized integration: Evaluate your

- centralized integration options to see if they actually simplify the task. A simple test is to time the amount of effort required to provision a new partner with a two-way SSL connection, including private key operations and storage. A good solution will enable you to provision a simple connection in minutes.
- Built-in scalability: It's possible to successfully provide basic security for a single Web service and write all the code necessary to apply standards using development toolkits. However, when the Web service is connected to a heterogeneous environment, integrates with other standards and policy requirements, and must evolve to meet new threats, it can be overwhelmed by all the security processing overhead required.

# Lesson 4 – Provide Visibility into Transaction Traffic

With visibility into traffic, application behavior, and potential issues, you can quickly uncover root causes of potential problems and defuse them before they disrupt business processes. Look for a secure deployment solution that:

- *Is purpose-driven:* Make sure it is designed to capture traffic information as part of its primary purpose
- Delivers in-depth information: Delivers detailed data about events and actual messages in the form of searchable detailed, configurable, and secure logs.

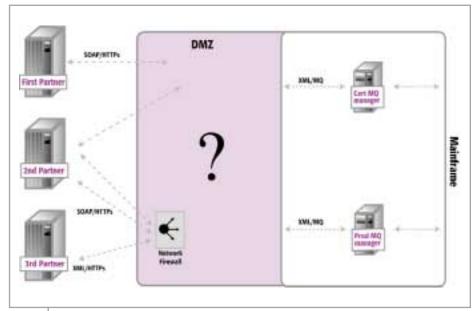


FIGURE 2 | Complexity increases as partners are added



# Secrets of the Web services Masters

Every WSJArticle on One CD!



# — The Complete Works —

CD is edited by WSJEditor-in-Chief Sean Rhody and organized into 40 chapters containing more than 400 exclusive WSJarticles!

All in an easy-to-navigate HTML format! BONUS: Full source code included!

# ORDER AT WWW.SYS-CON.COM/FREECD

\*PLUS \$9.95 SHIPPING AND PROCESSING (U.S. ONLY)

©COPYRIGHT 2004 SYS-CON MEDIA. WHILE SUPPLIES LAST. OFFER SUBJECT TO CHANGE WITHOUT NOTICE. ALL BRAND AND PRODUCT NAMES ARE TRADE NAMES, SERVICE MARKS OR TRADEMARKS OF THEIR RESPECTIVE COMPANIES.

Threat Type	Description	Best Defense
Identity threats	Updated XML versions of traditional identity threats such as authentication attacks and eavesdropping	Intelligent use of standards such as WS-Security and SSL will minimize some of the possibility of these sorts of attacks.
Content-borne threats	Attacks with elements in the actual XML payload such as XML viruses	Centralized validation policies and regularly updated signature files to examine every XML message entering the infrastructure
XML denial of service (XDoS)	New application-level versions of network level DoS attacks	Heuristic, configurable monitoring metrics with variable response to identify questionable traffic behavior.

TABLE 1 Windows implementation summary

This will allow you to easily debug connections, identify potential issues, audit messages, and resolve problems.

 Allows sharing: Information should be sharable between users, developers, and partners to expedite issue resolution.

# Lesson 5 – Make Web Services Easy to Manage and Operate

Successful Web services involve people from business units, developers, and members of the deployment team – as well as technical operations staff. As business needs change, the operations staff must be able to easily and cost-effectively manage them. Here's what a deployment system should provide how:

- Linear scalability: "Rack-and-stack" appliances simplify adding systems without additional management overhead, deployment costs, or personnel.
- Good interface: A good user interface will
  walk operators through processes and
  prevent them from making costly errors
  without requiring them to know XML programming languages on open multiple
  interfaces, such as shells and text logs
- Intuitive role- and function-based access:
   Make sure the interface presents only policies and data that the user can use.
   Ideally, seek a solution that coordinates the efforts of your distributed SOA/Web services team.
- Assured usability: By architects, developers, and administrators. Remember that
  other users, such as business unit managers, administrators, security staff, and

developers must be able to contribute to Web services without taking them offline, disrupting processes, or turning operations staff into data input clerks.

# Lesson 6 – Be Ready to Accommodate and Facilitate Change

Establish a process for change management so you can balance the needs of the business unit requesters with those of the individuals responsible for security, technology infrastructure, systems management, application and network operations, and auditing/record-keeping. For example, a deployment workflow system can make it easier for a broad community of users to provision, review, and approve policies without lengthy meetings. If each user has appropriate delegated authority, operators can deploy partial policies without stopping workflow. Being able to easily coordinate the efforts of everyone involved with XML Web services deployment will benefit your organization through more efficient rollouts and higher service availability.

# The Results of Lessons Learned

The global financial services firm initially built its own infrastructure and found that successful deployments took three to four months and required senior architectural staff to program and debug connections. After implementing several services this way, the firm elected instead to purchase a deployment system. That system reduced deployment time to just a few days and required involvement of only operational staff.

Because of its strategic commitment to XML Web services as a business enabler, Aeroplan chose to immediately implement a robust deployment environment. This was a requirement from deployment of its first service because of its customers' security, interoperability, and productivity requirements. When competitive advantage, consistent security, and return on Web service investment are your goals, it's never too early to consider a secure deployment process and system.

Look for one that:

- Offers flexibility: A solution that enables coarse- and fine-grained security policies allows users in different groups to consistently enforce and intelligently coordinate policies
- Offers manageability: One that can manage policy lifecycles, rationalize policy conflicts, and visually track policy changes greatly reduces management overhead. Since security policy can come from identity and access management systems, PKI management software, and custom policy decisions, intelligent policy coordination is a must-have for an effective secure deployment solution.
- Simplifies operations: Enables application and network operations personnel to use it without significant retraining. A good solution provides efficient tools, such as provisioning certificates, to automate error-prone, repetitive processes.
- Automatically mediates: Enables any-toany platform, protocol, and standards

36 August 2004 WWW. WS j 2. CO m

mediation to save testing and deployment time in heterogeneous environments.

#### About the Author

The director of marketing for Reactivity, Joelle Gropper Kaufman has extensive experience with Web services strategy and products. She previously served as the business manager for Web Services Security Solutions and senior product

manager for RSA Security. Today she is responsible for all marketing within Reactivity, a provider of proven, high-performance secure deployment systems for Web services serving the largest financial institutions and enterprises in the world. To learn more about how Reactivity can make your Web services work for your company, please contact us at www.reactivity.com.

■ ■ info@reactivity.com

#### The Payoff Is Here Now

With a solid deployment infrastructure in place, new SOAs and Web services can be deployed in hours or days. Reactivity, Inc. provides deployment systems that enable customers to rapidly deploy secure Web services connections for business-critical applications. Aeroplan deployed its industry's first real-time rewards system in 38 days - from proof of concept to production - with zero change to its existing systems. The financial services firm successfully manages more than \$12B in assets over the Reactivity secure deployment system. Both companies are generating revenue at a fraction of the cost of their previous proprietary, leased-line connections. They are provisioning new connections, services, and policies in minutes with significant operational and support cost reductions. They gained return on their investment almost immediately (see Table 2).

Metric	Before	After	Change
Time to implement new connection	99 days	< 1 day	- 99%
Time to change a policy	4800 min.	5 min.	- 99.9%
Salary cost per month to deploy & support	\$40,000	\$4000	- 90%
Number of connections	2	10	+ 500%
Incremental additional annual revenue	\$2 M	\$20 M	1000%

TABLE 2 **Results of actual Reactivity customers** 

Through Reactivity, these companies - and others - are already realizing the competitive advantages of SOAs and Web services by deploying them with a secure and scalable system.



International Web Services Conference & Exp

### Announcing Web Services Edge 2005 **Call for Papers**

Submit your proposal by August 15, 2004



The Largest i-Technology Event of the Year!

Minimum

#### Hynes Convention Center, Boston, MA February 15-17, 2005

Join us in delivering the latest, freshest, and most proper Web service solutions at the Fifth Annual With Services Edge 2005 East-International Conference & Expe as we bring together IT professionals, devolupers, policy makers, industry leaders and scaderrics to share information and exchange ideas on technology trends and best practices in secure Web services and related topics including:

- . Transitioning Successfully to SDA
- · Federated Web services
- \* ebXWL
- · Orchestration
- + Discovery
- . The Business Case for SOA
- · Interep & Standards
- . Web Services Management
- . Messaging Bases and SDA
- . Enterprise Service Buses
- . 308As (Service-Driested Business Apps)

- . Delivering RDI with SBA
- . Java Web Services
- . XML Web Services
- . Security
- . Professional Open Source
- · Systems Integration
- · Sarbanes-Oxley
- . Grid Computing
- . Business Process Management
- · Web Services Chareography

#### Interested in exhibiting, sponsoring or advertising?

Becoming a Web Services Edge exhibitor or sponsor offers you a unique apportunity to best position year company's message and visibility to a targeted authence of Web services professionals. Through a wide variety of pre-show, on-site, and post-show marketing programs and opportunities. Web Services Edge 2005 East will serve industry stalwarts as well as enciting start-up companies in reaching the world's most moture internet-based regional market, the US Northeast Region, at Boston's best-known IT business address, the Hyres Convention Center. For Exhibit & Sponsorship Information please contact: Brisha Davida, 201-802-3004, e-mail: grisha@ass-coa.cam

Tuesday, 2/15: Conference & Expo

Wednesday, 2/16: Conference & Expo

Thursday, 2/17: Conference & Expe

Sponsored by:















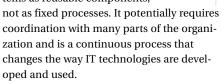
Contact for Conference Information: Lin Goetz, 201-802-3045, lin@sys-con.com

www.sys-con.com/



Service-oriented srchitecture (SOA) is an often-used term in today's IT organizations. Some surveys have shown that half of all Fortune 500 companies are actively pursuing an SOA in some form or another, many instituting it as their fundamental design concept.

OA is not something that you can simply buy off the shelf, but rather is a process and architectural mindset that enables a type of IT structure to be put in place. The SOA technical design approach focuses on organizing business systems as reusable components,



#### Definition

There are many definitions of SOA; however, most definitions center around application functionality that can be discovered and reused through loosely coupled, standardsbased interfaces. One of the more focused definitions comes from CBDI Forum, which states that SOA is:



Policies, practices, frameworks that enable application functionality to be provided and consumed as sets of services published at a granularity relevant to the service consumer that can be invoked, published and discovered, which are abstracted away from the implementation KERRY CHAMPION using a single, standards-based form of interface.

> Other definitions may reference "networkavailable software units" or "business-level services," yet these are essentially different ways of describing the same concept.

SOA environments are typically highly decentralized, with physically fragmented application functionality located throughout an organization (see Figure 1). These services are discoverable and accessible by service consumers. Service interfaces are abstracted from the back-end applications, decoupling interface from implementation details such as application type and operating system. Back-

end services still require context and relevance to the service consumer for them to be reusable. Exposing hundreds and thousands of services does not constitute an SOA until those services are "user-friendly" or "SOA ready" to the calling applications and with the appropriate infrastructure capabilities, such as security, monitoring, interoperability, and standards support attached to them.

#### **An Old Concept Revisited?**

SOA's focus on reusability is not a new concept and has been around for years. Notable attempts at achieving cross-system reusability include CORBA and DCE. They similarly had

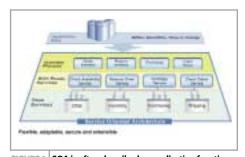


FIGURE 1 | SOA is often described as application functionality that is published and discoverable by calling applications as part of an overall business process. SOA-ready services give context and relevance to base services for the calling applications or business process. This enables better reuse, flexibility, and cost savings.

standardized interfaces and were callable by any application that similarly followed those standards. One might argue that most middleware products are SOAs; however, this implementation approach requires everyone to use the middleware vendor's set of proprietary interfaces as the "standardized interface."

Reasons why these technologies never hit the mainstream included:

- They were difficult to use and required a high level of expertise to coordinate and implement.
- They focused on exposing programminglevel objects, not business-level components.
- They implied an underlying programming model and were difficult to implement on top of a wide variety of legacy systems.
- They did not have a standard that had total industry-wide support.
- The approach tended to take significant time to implement, be very expensive and most importantly was very brittle.

One of the key principles of SOA is loose coupling, which allows the architecture to remain flexible and agile. These past approaches can certainly be described as SOA, but often were so tightly coupled that they were applicable only to the most high-end and expensive of projects.

#### **Along Came Web Services**

The standards that are synonymous with



## The standards that are synonymous with Web services are changing the playing field



Web services are changing the playing field. XML, SOAP, WSDL, and UDDI serve as the cornerstone of Web services and are dramatically lowering the cost of deploying standardized interfaces within and across the organization. Integration technology is now available at the grassroots level, enabling application developers and even casual desktop users to fully leverage value-added services. The fact that there is significant progress on standards and a virtual monopoly of supporters from the vendor and SI community are significant drivers to the success of Web services and SOA. Early success stories from end users and the commitment shown by developers and architects have made Web services and service orientation a foregone conclusion.

#### What Does All This Mean?

Many analysts and thought leaders in the industry believe that SOA is the next big wave to hit IT organizations. What is all the hype about?

A convenient starting point occurred several decades ago (see Table 1), when mainframe systems brought automation and computing power to companies. Complex and core data processing, which were central to the organization, could be offloaded to these expensive machines with large monolithic applications. Mid-range computers allowed many organizations and even individual departments to leverage computing power to suit their needs. The client/server revolution brought computing power to the desktop. This mirrored changes in the organization, where individual employees become knowledge workers and needed the tools and capabilities to do their jobs. The architecture changed to become more two-tiered with server applications providing the heavy lifting and processing while the application logic moved to the desktop. A top-down approach called CASE methodology became a popular methodology for creating applications.

In the past decade, Web-based technologies and thin clients have driven the agenda, putting business logic back at the server while also providing application access to a much broader base, including every worker, partner, and customer relevant to the organization. Around the same time, object orientation was bringing component-based technology to developers building applications. SOA is the next big revolution based on previous standards like SGML, XML, and HTTP. SOA enables application access and reuse throughout the entire organization.

Some have described a massive change in the way IT organizations will operate. Just as moving from mainframe to client/server dramatically changed the IT organization, SOA is predicted to cause a similar disruption. Back in the early 1990s, client/server architecture became highly popular with skills required for database, thick-client applications and client/server network technologies. During

Time Frame	Revolution	Computing Paradigm	Architecture
1970s	Mainframe	Monolithic	Single tier
1980s	Midrange	Departmental	Single tier
Early 1990s	Client/server	Power to the desktop for knowledge users	2-tier architecture (server, application)
Late 1990s	Web	Broad-based portal applications to back-end systems	3-tier architecture (database, server, client)
2000s	SOA	Services-based application infrastructure	Services-based

Many experts deem SOA the next big IT revolution-enabling IT organizations to be better aligned with business requirements and providing the flexibility and agility to respond to change better, faster, and cheaper.

This table provides a review of the past few decades' "next big revolution" in IT.

W W W . W S j 2 . C 0 M July 2004

the Web revolution, experts in application servers and Web-oriented technologies became highly valued as applications became available to a wider audience. In the SOA world, many are predicting that there will be additional specialization in creating and managing components and assembling and reusing them.

#### Why Do We Need SOA?

SOA has many benefits, the most common rationale of which are flexibility and agility. The fact is that SOA enables you to respond to change better, faster, and cheaper. Many organizations have recognized that in order to best align IT with business needs, they need technology systems to match and respond to the needs of the business units. The pace of business and the challenges associated with competition, increasing customer service, mergers and acquisitions, and outsourcing require a flexible architecture that is adaptable to changing needs. This agility must also minimize the constraints of cost, time to market and technology risk (see Figure 2).

IT organizations that deploy SOA gain efficiency by enabling application developers to quickly assemble applications from different available components. This not only saves time and money by reducing duplication, but also encourages a consistency in business process because as services are reused, process, to some extent, is also reused.

SOA also reduces dependence on vendors supplying monolithic solutions. Rather than having to choose one vendor for a fully integrated solution or platform, IT can more easily select

best-of-breed technology to solve their needs. Monolithic application suites are predicted to be much less common given that a services-based architecture breaks up the "stovepipes" and enables companies to reduce their reliance on one vendor and customize applications to their business requirements.

IT organizations are moving to SOA so that they can better address the realities of business today, which is all about being able to handle change. Changes can be long term activities that are proactively pushed by the organization or unexpected changes that require immediate action. An SOA environment is flexible, adaptable, and designed to handle change.

#### What Is Needed?

Unfortunately, you can't simply purchase SOA or deploy SOA by having a few meetings. SOA requires discipline, organization and the right tools to deploy effectively and maintain it going forward.

From an organizational and process standpoint, some best practices include:

- · Creation of a centralized forum for exchanging plans and information on services
- · Policies and incentives that encourage developers to create services, and for business units to reuse services
- · Tracking and regular evaluation to determine usage patterns and to allocate funds appropriately
- · Obtaining skill sets in component management and assembly

In terms of tools and technologies, Web

services standards help considerably with the need to institute standards-based interfaces. Web services, however, does not equal SOA. While Web services solves some important interoperability issues, its networks increase in complexity over time and can result in complicated connection points, redundant infrastructure, and inconsistency (see Figure

Web services provides some of the following benefits:

- Separation of interface and implementation
- · Support for standards-

Margers and A New Business Opportunities Legacy Spring

FIGURE 2 The benefits of change

based interfaces

· Support for mechanisms to dynamically discover and bind to services

However, SOA environments have many requirements above and beyond the use of Web services. These include:

- Separation of application business logic from operational infrastructure functionality.
- The need to address security of new componentized architecture and prevent uncontrolled access to every service.
- Address unreliability of service parts as part of a larger application process
- Service contract between consumer and providers to govern information inter-
- Support for business services relevant to the consumer, aggregated from back-end services

These requirements can all be encapsulated through abstraction, an absolute requirement for all SOA environments. Architects and analysts have varying names for this requirement, including service virtualization, service facades, and service views. Through the use of an SOA-ready abstraction layer, common infrastructure elements can be shared across all services, resulting in reusable services that are simpler to manage and built for change.

#### About the Author

Kerry Champion is CTO and founder of Westbridge Technology, a leading provider of security infrastructure solutions for Web services and service-oriented architecture (SOA). Prior to founding Westbridge Technology, Kerry was senior vice president of product development at Tumbleweed Communications, a leading supplier of e-mail security software and the provider of S/MIME standard reference implementation. Kerry is a frequent writer and speaker on XML Web services.

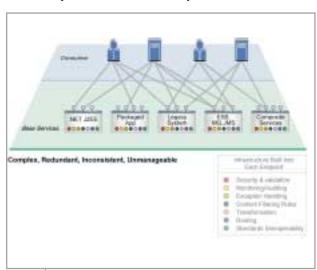


FIGURE 3 Web services is an important part of SOA but can still be highly complex and introduces redundant infrastructure and inconsistencies. Over time, Web services networks increase in complexity and become very costly to manage.

**September 27 – 29, 2004**JW Marriott Desert Ridge Resort and Spa

Phoenix, Arizona

gartner.com/us/ad

And registration information.



#### Four ALL NEW tracks on today's mission-critical AD issues

After three days of high-value learning, you'll come away with:

- State-of-the-art research and guidance from seasoned IT analysts
- A powerful blend of strategic insight and actionable know-how
- Hard-to-find best practices and end-user case studies
- A unbiased assessment of the leading providers and the newest solutions

#### Learn how to:

- Deliver software quality, security and business value
- Smoothly integrate your applications, whether in-house or packaged
- · Run your IT operations and services more effectively
- Decide if outsourcing development makes sense for your enterprise
- Better manage your project portfolio and team productivity

And much more!

#### KEYNOTE SPEAKERS and CONFERENCE CHAIR



Nick Carr

Author of
"Does IT Matter?"



Don Tapscott
World-renowned
technology guru



Mark Allen
Triathlete of the Century



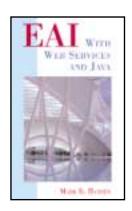
Theresa Lanowitz
Research Director,
Gartner



Dale Vecchio
Research Director,



# Web Services Messaging with JAX-RPC



#### From Chapter 5

■ The last chapter introduced simple SOAP messaging. In this chapter, we move to the next level of sophistication and introduce a true messaging framework that forms the backbone of the EAI infrastructure developed in this book. This messaging framework enables Java applications to send and receive Java objects to and from web services using SOAP as the transport. It also enables asynchronous messaging whereby the sender of a message does not have to block waiting for the receiver to process it. We call our framework the Web Services Messaging Framework (WSMF). It is implemented using J2EE and Web Services standards and serves as the core messaging service used by all the non-JMS EAI applications developed in the remainder of this book.

he WSMF is not a commercial-quality messaging framework, but rather an illustrative example of how to implement Web Services oriented messaging within the J2EE environment. In this chapter, you will encounter the major issues associated with implementing SOAP messaging in Java and some solutions that can be applied to your own EAI projects.

Our WSMF is implemented using the Java API for XML-based RPC (JAX-RPC). As the reader is probably aware, JAX-RPC was created to provide Java developers with a standard for incorporating into Java applications the XML-based Remote Procedure Call (RPC) functionality described in Section 7 of the SOAP 1.1 specification. JAX-RPC was not intended for messaging. The reasons that we have chosen JAX-RPC as the standard for our messaging framework are discussed in Section 1.1.

In addition to JAX-RPC, this chapter introduces two additional standards: JAXB

and WSDL. The Java Architecture for XML Binding (JAXB) provides a standard approach, based on XML Schema, for serializing and deserializing between XML documents and Java objects. The Web Services Description Language (WSDL) is a W3C standard interface definition language for specifying web services APIs using XML. Typically, a web services client reads the WSDL to determine how to structure SOAP messages that can be accepted as valid requests by the service.

This chapter assumes that the reader has a basic familiarity with JAX-RPC and WSDL. If you have never encountered JAXB, that should not be a problem. You can probably pick it up as you read through the material presented here. If you need to brush up on any of these standards, the author suggests Sun's Java Web Services Tutorial (<a href="http://java.sun.com/webservices/docs/1.4/tutorial/doc/index.html">http://java.sun.com/webservices/docs/1.4/tutorial/doc/index.html</a>) for JAX-RPC and JAXB. For WSDL, we suggest

that you read through the specification (<a href="http://www.w3.org/TR/wsdl">http://www.w3.org/TR/wsdl</a>). It is not very long and provides a lot of illustrative examples.

#### 5.1 JAX-RPC for Messaging

With some misgivings, we have implemented the WSMF using JAX-RPC. As you will see in this chapter, JAX-RPC, at least the current version 1.0, has many short-comings as a standard for messaging. The primary design goal for the initial release of JAX-RPC seems to have been enabling RMI-like remote procedure calls (RPCs) using SOAP over HTTP.

Another standard, JAXM, has been developed specifically for XML messaging. However, because JAX-RPC is a required part of the J2EE platform (starting with version 1.4) and the JAXM standard is optional, we believe that a messaging framework based on JAX-RPC is easier to introduce into a corporate Java environment. Any IT department supporting J2EE must support JAX-RPC. Therefore, any code you develop using the techniques illustrated in this chapter should be usable in a corporate Java environment.

As an interesting aside, you may be wondering why there is no true messaging standard for Web Services included in the J2EE standard (at least at the time of this writing). I.e., why is JAXM not included in the J2EE standard? The reason is that many of the vendors driving Java standards (e.g., IBM, BEA, Fujitsu) feel that J2EE already supports enough messaging standards. As IBM wrote in casting their vote against final approval of JSR #67 (JAXM): We are significantly concerned that the introduction of JAX-M as yet another API for calling a service (along with RMI-IIOP, JMS, and the proposed JAX-RPC) is unnecessarily complicating the Java programming model for our developers.

As a result of this decision, we need to work a little harder to implement Web Services messaging within the J2EE framework. This is not all bad, however, because it gives us an opportunity to gain a detailed understanding of the important JAX-RPC standard.

The JAX-RPC 1.0 specification is long and fairly complex. You don't need to understand the entire standard to learn how to use it for messaging. The approach that we suggest is to start working through

42 August 2004 WWW.Wsj2.com

the examples in this chapter and refer back to the JAX-RPC Javadoc API and specification as necessary.

When reviewing JAX-RPC, you should focus on these features, which are used by the WSMF:

- SOAP message support and the SOAP 1.1 binding including SOAP faults (JAX-RPC 1.0 Specification Section 6)
- · WSDL interface definitions of services and their Java representations (JAX-RPC 1.0 Specification Sections 4 and 5)
- · Dynamic invocation of operations on service endpoints (JAX-RPC 1.0 Specification Section 8)
- SOAP Message Handlers (JAX-RPC 1.0 Specification Section 12)

We also develop some workarounds to avoid the JAX-RPC Type Mapping Framework. Version 2.0 of JAX-RPC is going to replace 1.0 type mapping with JAXB, so we don't want to invest much effort incorporating it into our messaging platform. Indeed, as you will see, part of what makes JAX-RPC a bit clumsy as a

messaging framework is its built-in [de]serialization between Java types and XML. Throughout this chapter, we call this the JAX-RPC embedded type mapping. (Be aware that this is our phrase and not standard industry terminology.) Within our WSMF, we "turn off" the embedded type mapping and substitute the Java API for XML Binding (JAXB). JAX-RPC 2.0 promises improved type mapping and support for JAXB, but as of this writing, work on version 2.0 is just starting.

In the following sections, we look at each of these features and discuss how they are used to implement the WSMF. Sections 1.2 and 1.3 illustrate how JAX-RPC support for the SOAP 1.1 binding is leveraged to facilitate communication with Web Services using SOAP over HTTP. We first look at simple one-way messaging and then at request-response. Section Error! Reference source not found, introduces the server side framework for implementing messaging services. Section Error! Reference source not found. shows how WSDL is used to describe messaging services. We also look at how JAX-RPC uses WSDL descriptions to create client side representations of services. Section Error! Reference source not found. discusses how and why we use custom XML/Java bindings (e.g., JAXB) with JAX-RPC. Lastly, Section Error! Reference source not found. ties this all together and demonstrates a sample application that uses the WSMF.

#### 5.2 One-Way Messaging Using JAX-RPC (Client Side)

To start off, we consider the straightforward case of one-way messaging. In this scenario, the sender transmits a message to the receiver, but does not receive a response back. Often times, but not always, this scenario assumes that the receiver processes the mes-

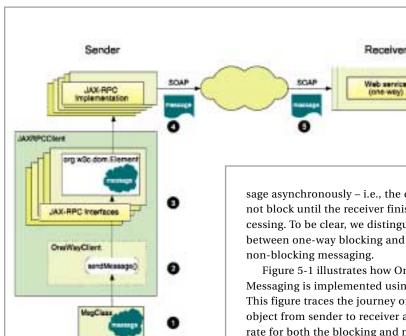


FIGURE 5-1 One-way Messaging using JAX-RPC

sage asynchronously - i.e., the client does not block until the receiver finishes processing. To be clear, we distinguish between one-way blocking and one-way

Figure 5-1 illustrates how One-Way Messaging is implemented using JAX-RPC. This figure traces the journey of a message object from sender to receiver and is accurate for both the blocking and non-blocking scenarios. In either case, something may come back to the sender at a lower protocol level, e.g., a simple HTTP ACK, an empty SOAP message, or a SOAP fault. That

Active-Endpoints	www.activeendpoints.com		21
Borland	http://info.borland.com/conf2004/		33
Gartner	www.gartner.com/us/ad		41
IBM	www.ibm.com/middleware/integrate		17
Information Starage+Security	www.ISSJournal.com	888-303-5282	49
IT Solutions Guide	www.sys-con.com	201-802-3021	19
LinuxWorld Conference & Expo	www.linuxworldexpo.com		8&9
Mindreef	www.mindreef.com		11&13
Open Link Software	www.openlinksw.com/virtuoso/	800-495-6322	Cover IV
Parasoft	www.parasoft.com/soaptest	888-305-0041	6
Prolifics	www.prolifics.com	800-675-5419	Cover III
Service Integrity	www.serviceintegrity.com		Cover II
SpeechTEK 2004	www.speechtek.com	877-993-9767	27
Strikelron	www.strikeiron.com	919-405-7010	5
SYS-CON e-newsletters	www.sys-con.com	888-303-5282	25
SYS-CON Publications	www.sys-con.com/2001/sub.cfm	888-303-5282	45
Web Services Edge East	www.sys-con.com/edge	201-802-3004	37
WebAppCabaret	www.webappcabaret.com/wsj.jsp	866-256-7973	3
WSJ Resource CD	www.sys-con.com/freecd	888-303-5282	35
XML-J Resource CD	www.sys-con.com/freecd	888-303-5282	51

their agents or agencies who are acting on behalf of the advertiser. This index is provided as an additional service to our readers. The publisher does not assume any liability for errors or omissions.

www.wsi2.com Auglust 2004 depends on the implementation details, which we get in to later. What defines one-way messaging is that no response is returned at the message level. Here the term message refers to the contents of the SOAP body and not the SOAP document itself. In Section Error! Reference source not found. we will see that an equivalent definition says that a message service is one-way if and only if it's WSDL port type specifies a one-way operation. As illustrated, one-way messaging involves these steps.

- 1. The message to send exists as an instance of a java class that can be serialized, i.e., marshaled, to XML that the receiver can process. Section Error! Reference source not found. describes how marshalling / unmarshalling is handled within WSMF. Throughout this chapter, we refer to such a class as MsgClass.
- 2. The instance of MsgClass is passed as a parameter to the OneWayClient interface method sendMessage(). One-way messaging clients in the WSMF must implement the com.javector.ws.OneWayClient interface (shown in Example 5-1 below) by extending the class

- com.javector.ws.jaxrpc.JAXRPCClient.
- 3. Within the JAXRPCClient instance, the message is marshaled to an implementation of org.w3c.dom.Element. The JAXRPC interfaces, in particular, the Call.invoke() or Call.invokeOneWay methods, are used to send the message to the receiver.
- 4. The runtime implementation of JAX-RPC (e.g., Apache Axis) on the client now takes our message as an org.w3c.dom.Ele ment and serializes it into a SOAP document that is sent across the wire (e.g. HTTP over TCP/IP) to the receiver. Although it is not spelled out in the JAX-RPC specification, we assume that all JAX-RPC implementations can serialize an org.w3c.dom.Element into the body of a SOAP document.
- 5. The message is received and processed by the web service.

Example 5-1 below shows the OneWayClient interface. Note that the return type of sendMessage() is void, i.e., no response message is returned. Implementations of this interface may be blocking or non-blocking as discussed above. In addition

to the message object, the method's parameters include the WSDL port and operation names. Beyond just the port and operation names, however, JAX-RPC requires the URI and name of the target service in order to instantiate a client side representation of a web service. For this reason, OneWayClient should be implemented by extending JAXRPCClient (discussed in Section Error! Reference source not found.). The JAXRPCClient class maps to a WSDL service and supplies the JAX-RPC run time implementation with this additional information. If you have read a little bit about JAX-RPC at this point, you can see that the OneWayClient interface gets mapped to a JAX-RPC javax.xml. rpc.Call interface.

#### Example 5-1 Interface for One-Way Messaging

```
package com.javector.ws;
 4
     public interface OneWayClient {
     Public Methods
8
9
1.0
     * Sends a message object to a
     service that does not return
11
     * anything. This method can be
     used to call asynchronous
12
     * message services.
1.3
14
     * @param msgObject The message
     to be sent
15
     * @param portName the qualified
     name of the WSDL port that
16
          contains the operation
     for receiving the message
17
     * @param operationName quali
     fied name of the WSDL operation
18
          on the specified port
     that accepts the message
19
     public void sendMessage
20
     (Object msgObject, String
21
     portName, String
     operationName)
22
     throws Exception;
23
24
     com/javector/ws/OneWayClient.java
```

In this section, we have introduced the simple OneWayClient interface which will

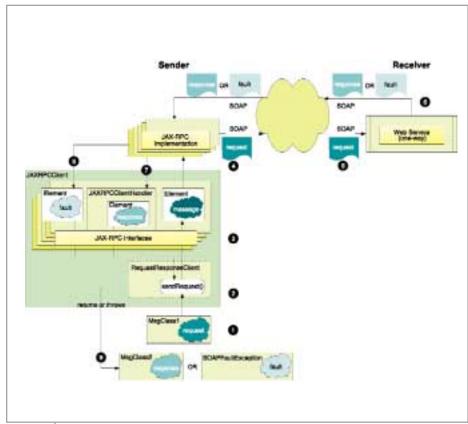


FIGURE 5-2 Request Response Messaging Using JAX-RPC



#### **CALL TODAY!** 888-303-5282

	LinuxWorld	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

#### 🗆 JDJ

U.S. - Two Years (24) Cover: \$144 You Pay: \$99.99 / Save: \$45 + FREE \$198 CD U.S. - One Year (12) Cover: \$72 Can/Mex - Two Years (24) \$168 You Pay: \$69.99 / Save: \$12 You Pay: \$119.99 / Save: \$48 + FREE \$198 CD Can/Mex - One Year (12) \$120 You Pay: \$89.99 / Save: \$40 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 Pav: \$89.99 / Save: \$6 Save: \$46 + FREE \$198 CD Int'l - Two Years (24) \$216 You Pay: \$170 / 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

#### Information Storage + Security Journal

U.S. - Two Years (24) Cover: \$143 You Pay: \$49.99 / Save: \$93 + FREE \$198 CD U.S. - One Year (12) Cover: \$72 You Pay: \$39.99 / Save: \$39 Can/Mex - Two Years (24) \$168 You Pay: \$79.99 / Save: \$88 + FRFF \$198 CD Can/Mex - One Year (12) \$84 You Pay: \$49.99 / Save: \$34 Int'l - Two Years (24) \$216 You Pay: \$89.99 / Save: \$126 + FREE \$198 CD Int'l - One Year (12) \$108 You Pay: \$59.99 / Save: \$48

#### **Wireless Business & Technology**

U.S. - Two Years (12) Cover: \$120 You Pay: \$49.00 / Save: \$71 + FREE \$198 CD U.S. - One Year (6) Cover: \$60 You Pay: \$29.99 / Save: \$30 Can/Mex - Two Years (12) \$120 You Pay: \$69.99 / Save: \$51 + FREE \$198 CD Can/Mex - One Year (6) \$60 You Pay: \$49.99 / Save: \$10 Int'l - Two Years (12) \$120 You Pay: \$99.99 / Save: \$20 + FREE \$198 CD Int'l - One Year (6) \$72 You Pay: \$69.99 /

#### 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 ☐ 9-Pack ☐ 1YR ☐ 2YR U.S. ☐ Can/Mex ☐ Intl.

TO

 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: \$93 + FREE \$198 CD U.S. - One Year (12) Cover: \$72 You Pay: \$39.99 / Save: \$32 Can/Mex - Two Years (24) \$168 You Pay: \$79.99 / Save: \$88 + FREE \$198 CD Can/Mex - One Year (12) \$84 You Pay: \$49.99 / Save: \$34 Save: \$126 + FREE \$198 CD Int'l - Two Years (24) \$216 You Pay: \$89.99 / Int'l - One Year (12) \$108 You Pay: \$59.99 / Save: \$48

#### **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 You Pay: \$159.99 / Save: \$80 + FREE \$198 CD Can/Mex - Two Years (24) \$240 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 You Pay: \$129.99 / Save: \$2 Int'l - One Year (12) \$132

#### ■ WebSphere Journal

U.S. - Two Years (24) Cover: \$216 You Pay: \$129.00 / Save: \$87 + FREE \$198 CD - 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 You Pay: \$189.00 / Save: \$75 Int'l - Two Years (24) \$264 Int'l - One Year (12) \$132 You Pay: \$129.99 / Save: \$2

#### 🗆 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 Save: \$180 + FRFF \$198 CD Can/Mex - Two Years (24) \$360 You Pay: \$179.99 / You Pay: \$169 / Can/Mex - One Year (12) \$180 Save: \$11 Save: \$170 + FREE \$198 CD Int'l - Two Years (24) \$360 You Pay: \$189.99 / Int'l - One Year (12) \$180 You Pay: \$179 / Save: \$1

#### **□** WLDJ

U.S. - Four Years (24) Cover: \$240 You Pay: \$99.99 / Save: \$140 + FREE \$198 CD You Pay: \$49.99 / U.S. - Two Year (12) Cover: \$120 Save: \$70 Save: \$140 + FREE \$198 CD Can/Mex - Four Years (24) \$240 You Pay: \$99.99 / Can/Mex - Two Year (12) \$120 You Pay: \$69.99 / Save: \$50 Int'l - Four Years (24) \$240 You Pay: \$120 / Save: \$120 + FREE \$198 CD Int'l - Two Year (12) \$120 You Pay: \$79.99 / Save: \$40

\*WHILE SUPPILES LAST. OFFER SUBJECT TO CHANGE WITHOUT NOTICE



Pick any 3 of our magazines and save Pay only \$99 for a vear subscription nius a FREE CD

- 2 Year \$179.00
- Canada/Mexico \$189.00
- International \$199.00

Pick any 6 of our magazines and save \$34000 Pav only \$199 for a vear subscription **plus 2 FREE CDs** 

- 2 Year \$379.00
- Canada/Mexico \$399.00
- International \$449.00

Pick 9 of our magazines and save Pay only \$399 for a 1 year subscription plus 3 FREE CDs

- 2 Year \$699.00
- Canada/Mexico \$749.00
- International \$849.00





## Any IT department supporting J2EE must support JAX-RPC



be used throughout the rest of this book for web services messaging. Although JAX-RPC is complex, because we are focused on messaging, we can simplify our client-side API down to a couple of interfaces. The next section discusses the interface for Request-Response messaging.

#### 5.3 Request-response Messaging Using JAX-RPC (Client Side)

Request-Response messaging works the same as One-Way except that the client receives a response message from the web service. As with One-Way, Request-Response messaging can be blocking or non-blocking. Most of us think of Request-Response over SOAP as being implemented as a "SOAP for RPC" application where the request message is carried on the HTTP request and the response message comes back on the corresponding HTTP response. This is the most common scenario, and it implies a blocking client that returns only after the response has been received. However, there are many situations where it is not practical or efficient to block for the response message. These non-blocking scenarios are covered in detail in Chapter 6 - Reliable Messaging. In this chapter, we restrict our consideration of Request-Response to the blocking scenario.

As shown in Figure 5-2 steps 1–5 of the Request-Response messaging are the same as in the One-Way scenario. Steps 6–9 trace the response message which follows a slightly different path depending on whether it is a fault or a response. Where a fault message is deserialized according to the JAX-RPC implementation's embedded type mapping, a response message gets deserialized by a special handler. This handler is discussed in detail in Section Error! Reference source not found. The response steps are as follows:

6. The web service sends either a fault or a

response as XML within the body of a SOAP document. If it is a fault, then the structure of the SOAP document is as defined in Section 4.4 of the SOAP 1.1 specification. Otherwise, the response is the single child element of the SOAP body. When the SOAP document is received by the JAX-RPC implementation, the processing depends on whether it represents a fault or a response.

- 7. If the SOAP document is a response, it is processed by an instance of the class JAXRPCClientHandler, that implements the JAX-RPC javax.xml.rpc.handler.Handler interface. This handler extracts the response message as an instance of org.w3c.dom.Element and hands it to the JAXRPCClient instance to be unmarshalled to a Java object as per the XML binding that has been implemented.
- 8. On the other hand, if the SOAP document is a fault, it is not processed by the handler. Instead, the JAX-RPC implementation extracts the fault and throws it as an instance of SOAPFaultException. This exception, in turn, is thrown by the method sendRequest().
- In this manner, the method sendRequest() either returns an Object containing the response or throws a SOAPFaultException containing the fault.

Example 5-2 below shows the RequestResponseClient interface. It is identical to OneWayClient except that it returns an Object instead of void. Within the WSMF, the RequestResponseClient interface is implemented by JAXRPCClient.

```
3 package com.javector.ws;
4
```

5 public interface

```
RequestResponseClient {
    --Public Methods
8
      /**
9
10
      * Sends a message object
      synchronously and returns the
11
      * response object that it
      receives from the remote
12
      * service.
1.3
      * @param requestObject The
14
      request to be sent
15
      * @param portName the quali
      fied name of the WSDL port
      that
16
            contains the operation
      for receiving the message
17
      * @param operationName
      qualified name of the WSDL
18
           on the specified port
      that accepts the message
      * @return the response
19
      received from the remote
      service
20
21
      public Object sendRequest
22
      (Object requestObj, String
      portName, String
      operationName)
23
      throws Exception;
2.4
25
      com/javector/ws/Request
      ResponseClient.java
```

#### Example 5-2 Interface for Request-Response Messaging

Sections 1.2 and 1.3 have provided an overview of how client side messaging with JAX-RPC is structured in the WSMF. Before getting into the client side details, we use the next section to get a quick overview of how we can write web services in Java.

#### About the Article

This article is excerpted from a chapter within the forth-coming, November 2004 book, *EAI with Web Services and Java*, by Mark D. Hansen, ISBN 0-13-044968-7. This chapter is printed with permission from publisher Prentice Hall PTR, copyright 2005. This book can be pre-ordered at Amazon or BN.com.

■ ■ mark@javavector.com

46 August 2004 WWW. WS j 2. CO m



#### This Month

#### Banking on a Standard

BY MIKE HAEHN

One of the basic challenges of XML developers is formulating best practices and design guides for defining their XML content. This is especially pervasive in industry communities, which is why there has been a growth of community-based XML specifications and standards.

#### Using XSLT on Bioinformatic XML Data

BY RUSSEL BRUHN, PHILIP BURTON, & GARY THOMPSON

For the biologist, the bioinformatic analysis of genes requires the compilation of tables of gene characteristics. To do this, data is often taken manually out of databases in an ad hoc fashion. Different databases (TIGR, MIPS, BLAIR, and NCBI, for example) give different outputs in different formats.

We would like to be able to extract information from the databases in a common, structured file format in a way that allows for easy rearranging and processing of the data.

#### Look Ma Bell - No Hands!

BY LES WILSON

The emerging world without wires has fostered a growing number of small and mobile devices (everything from PDAs to smart phones) capable of accessing data and running applications. The trouble is, while devices are getting smaller, human hands and fingers are not



#### XML-Based Interop, Close Up

n addition to the strategy side of Web services, there is also the protocol-oriented side of things, the XML side. Embracing not only XML itself but also the full range of mainstream XML-based technologies like XPath, XSLT, XML Schema, and SOAP, XML-Journal has been delivering insightful articles to the world of developers and development managers since the year 2000.

It is our privilege to bring *XML-Journal* directly to readers of *Web Services Journal*, and vice versa. Anyone already familiar with the Web services world of SOAP, UDDI, and WSDL will find here articles and features each month that will interest them – about the cutting-edge technologies and latest products that are changing not only our industry,

but the way the world exchanges information. To make it easy for you to find your way around, we have four distinct sections:

#### **Content Management:**

Organization, dissemination, and presentation of information

#### **Data Management:**

Storage, transformation, representation, and general use of structured and unstructured data

#### **Enterprise Solutions:**

Systems and applications that manage mission-critical functions in the enterprise

#### Labs:

Product reviews, book reviews, tutorials, and standards analysis





WRITTEN BY MIKE HAEHN

#### IFX for the retail and commercial banking arenas

ne of the basic challenges of XML developers is formulating best practices and design guides for defining their XML content. This is especially pervasive in industry communities, which is why there has been a growth of community-based XML specifications and standards.

In the financial industry, the Interactive Financial eXchange (IFX) Forum has been working for over seven years to develop a business message specification to satisfy the need for a community vocabulary and messaging specification in the retail and commercial banking arenas. But IFX is much more than just a community vocabulary. IFX provides design rules and a framework to successfully achieve consistency and interoperability, within a bank and between a bank and other entities.

#### **About the IFX Forum**

The IFX Forum is a nonprofit, market-driven organization comprising financial service institutions, service providers, and technology vendors working together to develop interoperable specifications to satisfy the needs of the financial services industry.

The organizational structure consists of the following:

- Steering committee: Acting executive body
- Architecture committee: Oversees IFX framework and architectural rules
- Working group(s): Formed as needed, facilitates messages based upon business requirements

The basic goal of the IFX Forum is to create reusable messages supporting a wide variety of applications that enable interoperability between heterogeneous environments. To achieve these goals, the forum leverages the following:

- Defined design rules: IFX strives for reuse, interoperability, object orientation, and extensibility.
- A structured message framework: Messages are consistently structured and have defined rules on how they are to be processed to allow both the client and server to have predefined and consistent expectations.
- Platform and technology independence: IFX may be imple-

mented on any transport and computing platform.

- Interaction model independence: IFX can be used for realtime and batch transactions.
- Work with other organizations: IFX proactively works with other organizations to ensure, in areas of overlap, there is consistency and interoperability.

#### **How IFX Messages Are Designed**

While the great majority of IFX implementations are developed using XML, IFX has not been designed solely with XML in mind; rather, it is built as a technology-independent business semantic. IFX is then implemented in a technology, primarily XML. This separates the business semantic from the container (XML) and even the transport (such as HTTP or MQ).

By providing these points of isolation, the IFX Forum working groups can focus on their business subject matter the requirements and workflow to satisfy business needs. The output of the working groups is then reviewed by a standing architecture committee, which ensures consistency and functionality before these enhancements are incorporated into the overall specification. The end result of this process provides comprehensive reusable messages and aggregates (complex structures) mutually agreed upon by financial institutions and software vendors alike.

As an example, the Interest Rate Information Aggregate, <IntRateInfo>, describes the necessary information to fully define interest rate information. The additions of usage constraints and descriptions provide for the semantic need and context that some XML vocabularies have not considered. <IntRateInfo> can now be reused in whatever messages need interest rate information, such as an Account Inquiry (see Table 1).

And the XML rendering of this structure may look like the following:

<IntRateInfo>

<Rate>5.25</Rate>

<Desc>Mortgage Rate: Pending Approval</Desc>

</IntRateInfo>

Note: In this example, only two elements are presented

because that may be all that is necessary to satisfy the particular business requirement.

#### A Real-World Example

So what does this all mean exactly? Imagine a service, such as Funds Transfer, which is a common service banks provide. Customers may transfer funds from several different delivery channels, such as the branch teller, call center, Web banking, or even an ATM device. IFX Forum members working together capture the general requirements and the differences in requirements that one particular channel might have from another. All of these requirements are then defined as part of a common Funds Transfer service. Now a developer can provide a single code base for several different delivery channels. This is the embodiment of the goals of service-oriented architectures. This same solution can also be used to satisfy the need for a Web service.

#### Web Services: the Next Step

The latest technology driver to arrive is Web services. Web services promise application-to-application communication. Ensuring this functionality is one of the key drivers for the future of the IFX specification, and formal interoperable definitions will be provided with IFX 2.0. Web services support will further promote the communication between banks and service providers, such as bill pay providers and check ordering providers.

As an example, a bank customer may log onto his or her online banking Web site, which uses IFX, to place an order for new checks. The bank has partnered with a check printer that has exposed an IFX "Check Order" Web service. The customer places an order through the online bank. The request is submitted directly to the check printer. A successful response is then transmitted from the check printer back to the bank. The bank then conveys the successful response to the customer. All of this is transparent to the customer, but the real benefit is the integration and cost savings between the bank and check printer.

#### IFX and Interoperability

We have seen several examples of interoperability with IFX, from the ability to have a single code base work with multiple delivery channels to the upcoming support of Web services enabling better communication with banking partners. The IFX Forum also works to interoperate with other standards bodies. A testament to this is the IST (International Standardization Team) Harmonization team, a joint effort between the IFX Forum, SWIFT (Society for Worldwide Interbank Financial Telecommunication), TWIST (Treasury Workstation Integration Standards Team), and OAGi (Open Applications Group, Inc.) that recently completed a definition of a comprehensive payment kernel to enable straight-through processing (STP) and interoperability of payments between the different standards bodies. This common kernel will reduce manual processing points and further promote standardization across international banking lines.

#### **Participation in the Development Process**

As previously stated, the IFX Forum is a market-driven organization. Forum members bring forward business cases and requirements which in turn grow the specification. Members gain the added benefit of networking with other members, obtaining information in a timely manner, and receiving expert advice. The IFX Forum releases new versions of the specification up to twice a year. Members are well aware of the enhancements long before the official release, which allows them to incorporate



The World's Leading i-Technology Publisher

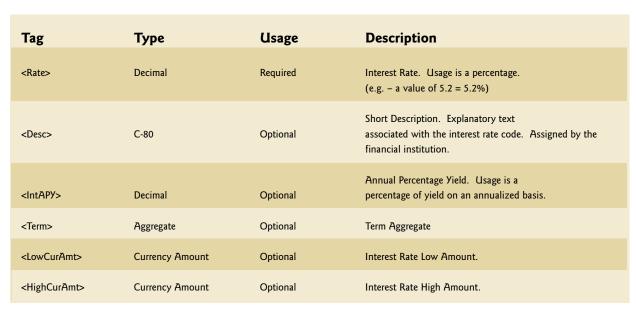


Table I • Interest Rate Information Aggregate

the latest enhancements by the time the latest release is made publicly available. (For more information on the IFX Forum and membership, please visit <a href="https://www.ifxforum.org">www.ifxforum.org</a>.)

#### **Developing Your Own IFX-Based Solution**

So what should you look for in developing an IFX-based solution? The answer truly depends on your business needs. IFX is not a specification that has to be fully implemented; you can pick and choose the messages necessary to support your business services. Thus, in a small-scale environment, such as a bill pay provider, you may only have to implement a handful of messages. There are many implementations of IFX like this, where a service provider or even a financial institution has implemented just a few messages to support a new service. This is an excellent solution for small services such as bill payments, check ordering, or even currency exchanges.

A more interesting example is that of a financial institution interested in leveraging IFX across the enterprise. Here is the opportunity for an institution to bind the benefits of IFX to an SOA. Because of its high reuse and object-oriented design goals, many portions of the specification can be coded once and reused as a service component for multiple services and/or channels. Structures for things like postal addresses, where the business rules are channel independent, can be defined once as a single module and used as "code off the shelf," plugging the module into the many different messages that use it in IFX.

"A well-designed IFX implementation can help reduce integration timelines from months to weeks"

When developing your IFX solution, you should be aware of some risks. Because IFX does not define requirements for infrastructure, implementations between business partners must agree on such things as transport and enveloping. Simply because an entity has an IFX implementation, that does not mean you can merely "plug in" and be up and running. In some cases, because of transport layer differences, adapters may be necessary to perform some minor translation.

Implementers of IFX should also be aware of the risks of deviating greatly from the specification. An "IFX-ish" implementation may not provide you with much return on investment when you are attempting to integrate with a business partner or service provider using IFX. By aligning yourself with IFX and working with the IFX Forum to evolve the specification where necessary, you can achieve interoperability benefits. A well-designed IFX implementation can help reduce integration timelines from months to weeks. And by shedding proprietary solutions, you can now more easily make changes in a heterogeneous environment.

#### Conclusion

IFX provides many design benefits for XML developers in the financial industry. By leveraging IFX for XML related projects, not only do developers gain a huge head start in comparison to defining their own messages, they also have the added benefit of industry agreement on functionality. The benefits of using IFX will continue to grow as financial institutions and software vendors continue to build and support IFX implementations, further reducing proprietary development efforts.

#### AUTHOR BIO

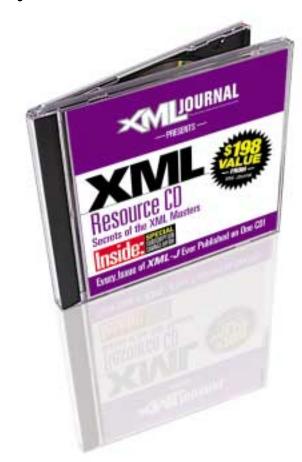
Mike Haehn is an independent consultant working in the financial services industry. He focuses primarily on enterprise XML messaging, Web services, and service-oriented architecture. Mike has been an active participant in the IFX Forum since 2001 and has worked on several IFX-related projects.

MIKE@MIKEHAEHN.COM



### Secrets of the XML Masters

Every XML-J Article on One CD!



#### — The Complete Works —

CD is organized into 33 chapters containing more than 450 exclusive *XML-J* articles!

All in an easy-to-navigate HTML format! BONUS: Full source code included!

#### ORDER AT WWW.SYS-CON.COM/FREECD

\*PLUS \$9.95 SHIPPING AND PROCESSING (U.S. ONLY)

©COPYRIGHT 2004 SYS-CON MEDIA. WHILE SUPPLIES LAST. OFFER SUBJECT TO CHANGE WITHOUT NOTICE. ALL BRAND AND PRODUCT NAMES ARE TRADE NAMES, SERVICE MARKS OR TRADEMARKS OF THEIR RESPECTIVE COMPANIES.



## Using XSLT on

WRITTEN BY RUSSEL BRUHN PHILIP BURTON **GARY THOMPSON** 

## Bioinformatic XML Data

#### Author Bios

ussel Bruhn earned his PhD in electrical engineering from Washington State University in 1997. He is an associate professor and chair of the Department of Information Science at the University of Arkansas in Little Rock. His research interests are in the areas of creating innovative curriculum, computers and education, XML, and applications of XML with SVG graphics

in mathematical physics from the University of Queensland in 1996. He is an assistant professor in the Department of Information Science at the University of Arkansas in Little Rock, Philip is a member of the Institute of Electrical and Electronic Engineering (IEEE) Society, the American Mathematics Society (AMS), and the American Physical Society (APS)

Philip Burton earned his PhD

Gary A. Thompson earned his Ph.D. in plant genetics from the Department of Botany and Plant Pathology at Purdue University in 1989. He is a professor in the Department of Applied Science at the University of Arkansas at Little Rock. Gary is jointly appointed with the University of Arkansas Division of Agriculture and was formerly on the faculty in the Department of Plant Sciences at the University of Arizona. He is a member of the American Society of Plant Biologists. His research interests are in the areas of plant

#### When data changes frequently

or the biologist, the bioinformatic analysis of genes requires the compilation of tables of gene characteristics. To do this, data is often taken manually out of databases in an ad hoc fashion. Different databases (TIGR, MIPS, BLAIR, and NCBI, for example) give different outputs in different formats. We would like to be able to extract information from the databases in a common, structured file format in a way that allows for easy rearranging and processing of the data.

The Extensible Markup Language (XML) is being used increasingly to represent semi-structured data and transmit it over the Internet. XML data is data that is marked up by tags in a manner similar to those in the Hyper-Text Markup Language (HTML). For example, the following code shows one way of using XML to mark up the protein with accession number "BAA-03739.1". It is taken from the National Center for Biological Information (NCBI) Internet Web site. The NCBI output can be expressed in the XML file format. All the main biological databases on the Internet now give the user the option of choosing output as XML.

<GBOualifier>

<GBQualifier\_name>protein\_id</GBQuali fier\_name>

<GBOualifier value>BAA03739.1/GBOual

ifier\_value> </GBQualifier>

HTML markup is used to format and present data; XML is used to organize and structure data. It is the user who defines the choice of elements and attributes, the types of data contained in them, and the way the elements nestle within each other. One of the principal characteristics of XML is the separation of the data itself, which is the XML document, from the formatting instructions in the style sheet. The rules governing the document structure and data types are also kept separate in a document called the schema.

To illustrate the use of XML as an open data format for exchanging and processing biological data, data files describing rice starch synthase genes were downloaded from the NCBI database in an XML format (GBSeqXML). One of the standard XML tools, Extensible Stylesheet Language Transformation (XSLT), was used to process and transform the information required into a table within a text document. XSLT is capable of transforming XML documents from one XML format into another and can also transform XML into HTML for presentation, or into text for documentation.

#### Case Study: Using XSLT on a **Bioinformatic XML Data File**

The NCBI Internet Web site is a huge biological database. The authors were interested in information on the rice soluble starch synthase genes. Rice soluble starch synthase genes are

"nucleotides" and each gene has a unique accession number in the NCBI database. Thus a search was done under the keyword "nucleotide" for accession number "D16202". The output was downloaded in the "GBsSeqXML" format and saved as an XML file on our hard drive. The truncated file, saved as "file1.xml," is shown in Listing 1.

The top line is the XML declaration; it identifies the document as actually being an XML document for an XML processor. The declaration starts with the text "<?xml version="1.0"?>", which signals the parser that an XML declaration follows and that the version number of the XML specification being used in the document is "1.0". The encoding value in the first line identifies the character codes used in the document. Because different languages use different encoding schemes, this declaration allows XML to support different languages. The default encoding scheme is the English language scheme "UTF-8." In this article we use "UTF-16," the 16-bit Unicode scheme or international language scheme. It is important to realize that XML is case sensitive. This is a critical difference from HTML, with which many biologists are familiar, which is not case sensitive.

The second line links the XML document to the schema. A schema can be thought of as a set of rules that establishes the format and structure created for the document. A schema describes precisely the permitted elements and attributes that are available within a given XML document, along with the

molecular biology and

genomics

relationships between the elements. We can think of a schema as a legal contract between the person who created the markup language and the person who will create documents using that language. Each document that conforms to the schema is referred to as an instance of the schema, and within the rules of the schema a wide variation in instance documents is possible. Not all instance documents will contain the same informa-

When it comes to creating schemas, two different approaches can be

- Document type definitions (DTDs)
- XML Schema definitions (XSDs)

The second line in Listing 1 is the document type declaration for a public external DTD. A document type declaration is a line of code that identifies the DTD being used; in this case it is a URL, which indicates its location as "http://www.ncbi.nlm.nih.gov/dtd/ NCBI\_GBSeq.dtd." The big distinction here is that the definition (DTD) actually describes the markup language, whereas the declaration connects the document to the DTD, which may be located on a remote server.

The goal is to make an XML document a valid document. Document validity is extremely important because it guarantees that the data within the document conforms to a standard set of guidelines, as laid out in a schema. Our example in Listing 1 uses a DTD. In a valid XML document, all rules, elements, and attributes match the logical structure and data types defined in the DTD schema. Not all XML documents have to be valid. To validate an XML document the parser must read the DTD, validate the document against it, and report any violations to the XML application. Because this takes time, some XML applications might use XML to code small chunks of data that really don't require the thorough validation options made possible by a schema or DTD.

Even if the XML document is not valid it must be well-formed. A wellformed XML document conforms to the World Wide Web Consortium (W3C) XML specification 1.0. Rules for well-formed XML documents include matching start tags with end tags and setting values for all attributes used. A well-formed XML document contains one or more elements. It has a single root document element. In Listing 1 the root element is "GBSet," all other elements are properly nested under it, and each of the parsed entities is referenced correctly. An XML document is a structure of elements, attributes, and text all nested within the root element. Well-formed XML documents do not require a DTD but valid XML documents do.

#### **XSLT Implementation**

XSLT is used to transform XML documents into other documents, such as HTML or text. XSLT processors parse the input XML document and then process the instructions found in the XSLT stylesheet, using the elements from the input XML document. The familiar markup structure, using the less than "<" and greater than ">" symbols makes its syntax readily identifiable and easier for some people to use than a procedural language. During the processing of the XSLT instructions, which are in the form of XML elements, a structured text output is created. XSLT instructions could also use XML attributes to access and process the content of the elements in the XML input document.

Using XSLT to perform transformations on XML is easier than writing a custom application with a procedural language because the design of XSLT is based on the recognition that these XML documents are all very similar. It should be possible to do the processing using the XSLT declarative language rather than by writing a program from scratch in Java or some other programming language. The required transformation can be expressed as a set of rules. The output we want to generate from particular patterns that occur in the input will define the rules. The language is declarative because one describes the transformation required as a set of transformations, rather than by creating a sequence of procedures in a given order. The process is simpler because XSLT describes the required transformation and is a complete programming language in itself.

Typically, most genetic databases produce data files with a size in the amount of millions of records. It would be faster to parse this data with a procedural language because XML and XSLT require lots of processing power to parse large XML documents. The major disadvantage of a parsing approach is that when data formats change, as they often do, the parsers will not work and the procedural language must be completely rewritten or major modifications must be made. Take, for instance, the recent change where Unigene started appending the version number to their NCBI Gen-Bank Accession Number. Using XML technology, the programmer adds a tag to extract the GenBank Accession Number. The advantage of XML over raw flat files is that no code must be rewritten.

By using XSLT on the downloaded data of rice starch synthase genes, which were saved as XML files, we can construct a table that includes characteristics such as "accession number," "molecular type," "protein number," etc. XSLT can extract any piece of data from the XML file, process it using built-in functions, and format it on the page in any way desired. Listing 2 shows many of these features.

Listing 2 has been truncated for space reasons. Some of the XSLT code used is adapted from XSLT Cookbook by Mangano and further features of the code are described there. An XML document may be visualized as a tree structure of elements, attributes, text, comments, etc. XSLT is a mapping from the source tree into the result tree. Each node that is to be mapped (each branch and leaf of the tree) has a rule associated with it called a "template" that describes how the node is to be transformed. At the top of the tree is an imaginary node called the "document root," denoted by "/". It corresponds to the XML declaration. A node is addressed using the descriptive language XPath. For example, "/GBSet/GBSeq/GBSeq\_primaryaccession" refers to the element "GBSeq\_ primary-accession" which is a child element of "GBSeq" and which is

Prim_acc	Mol_type	Clone	Protein_#	Protein_name	Amino_Acid	Mol_Wt
D16202	mRNA	2533	BAA03739.I	starch synthase	626	79623
D38221	genomic DNA	8900	BAA07396.I	SSSI	626	79713
АУ100471	genomic DNA	8160	AAM49813.1	soluble starch	919	20675
АУ10046	genomic DNA	9586	AAM49811.1	soluble starch	1553	202574
AF432915	genomic DNA	8821	AAL40942.I	starch synthase	1216	159920
AF419099	mRNA	2959	AAL16661.1	soluble starch	810	102838
AJ308110	mRNA	2319	CAC59826.1	starch synthase	637	80765
AF395537	mRNA	2394	AAK81729.1	starch synthase	694	87982
AF383878	mRNA	2903	AAK64284.I	starch synthase	749	96679
AF165890	mRNA	2585	AAD49850.1	starch synthase	626	79713

Table I • Formatted data on rice soluble starch synthase proteins from NCBI Web site

itself a child element of "GBset." This in turn is a child element of the document root "GBSet," which is the topmost element or "root element." In our code the line "document(\$file)/GBSet/GBSeq/GBSeq\_primary-accession" is shortened to "document(\$file)//GBSeq\_primary-accession." The two slashes, "//," are a set of nodes consisting of every GBSeq\_primary-accession element in the tree.

Any of the elements and attributes in the downloaded XML files can be extracted by XSLT code and formatted into a text or HTML file. But XSLT does more than just extract data from the XML source: it processes data into information. For example, using the built-in string function, "stringlength(...)", the number of aminoacids (entry Amino\_Acid, Table 1) can be found. Similarly, the molecular weight (in Daltons) can be calculated using the XSLT arithmetic features.

#### Results

Ten GBSeqXML files on rice synthase proteins from the NCBI Web site were downloaded and saved as XML files. Our XSLT program was written to extract and format the data and present it as Table 1. The first column in Table 1 is the primary accession number, which uniquely identifies each file. Our XSLT looped through the ten files and extracted the data shown in Table 1. Any data in the XML files could have been extracted. The

first five columns were obtained by simple extraction and the last two columns were calculated from the amino acid sequence. The XSLT Altova XMLSPYprocessor was used.

#### **Conclusion**

This article has shown how the declarative language XSLT can be used to extract and format data into a table. No procedural programming is needed to produce output like that shown in Table 1. Only XSLT was used. Since data formats like NCBI change frequencies often, using XML technology is an advantage because the code does not have to be rewritten. Only the necessary declarative tags need to be added to extract the new data, if needed. Using a procedural language to parse the data would most likely require major modifications in the code. The advantage of using a procedural language over a declarative language is that the processing speed is faster.

For our work here, the XML output files were obtained directly by downloading them to the hard drive. XML processing could be combined with Java programming to automate the process. For example, a Java program could have been written to work in conjunction with the gene output from the NCBI Web site to automatically generate the XML output files and process them into the text docu-

ment. This would have required procedural programming skills; what we have done here uses only the built-in declarative features of the XSLT tool.

#### References

- Baxevanis, A. and Ouellette, F. (2001)
   Bioinformatics: A Practical Guide to
   the Analysis of Genes and Proteins,
   John Wiley & Sons.
- Cagle, K. M. Corning, Daimond J., and Duynstee T. (2001) *Professional* XSL, Wrox Press Ltd.
- Castro, E. (2001) XML for the World Wide Web: Visual QuickStart Guide, Peachpit Press.
- Birbeck, M., J. Daimond J. Duckett, and Gudmundsson, O.G. (2001) Professional XML, Wrox Press Ltd.
- Van der Vlist, E. *XML Schema*, (2002) O'Reilly & Associates.
- NCBI Web site: www.ncbi.nlm.nih.gov
- Michael, K. (2001) XSLT Programmer's Reference, Wrox Press Ltd.
- XML Web site: www.w3.org/TR/REC-xml or www.w3.org/XML/Core
- Gardner, J. R. and Rendon, Z.L. (2002) XSLT & XPATH A Guide to XML Transformations, Prentice Hall.
- Mangano, S. (2002) XSLT Cookbook, O'Reilly & Associates.







```
Listing 1
<?xml version="1.0" encoding="UTF-16"?>
<!DOCTYPE GBSet PUBLIC "-//NCBI//NCBI GBSeq/EN"</pre>
"http://www.ncbi.nlm.nih.gov/dtd/NCBI_GBSeq.dtd">
<GBSet>
  <GBSea>
     <GBSeq locus>RICSSS</GBSeq locus>
     <GBSeq_length>2533</GBSeq_length>
      <GBSeq_moltype value="mrna">5</GBSeq_moltype>
      <GBSeq_topology value="linear">1</GBSeq_topology>
      <GBSeq_division>PLN</GBSeq_division>
      <GBSeq_update-date>11-JAN-2003</GBSeq_update-date>
     <GBSeq_create-date>26-JAN-1994</GBSeq_create-date>
     <GBSeq_definition>Oryza sativa mRNA for soluble starch
synthase, complete cds
 </GBSeq_definition>
     <GBSeq primary-accession>D16202</GBSeq primary-acces-
     <GBSeq_accession-version>D16202.1</GBSeq_accession-ver-
sion>
     <GBSeq_other-seqids>
           <GBSeqid>dbj|D16202.1|RICSSS</GBSeqid>
           <GBSeqid>gi|450484</GBSeqid>
      </GBSeq_other-seqids>
       <GBSeq_source>Oryza sativa</GBSeq_source>
        <GBSeq references>
            <GBReference>
               <GBReference_reference>1 (bases 1 to
2533)</GBReference reference>
               <GBReference_authors>
                   <GBAuthor>Baba, T. </GBAuthor>...(truncated
GBReference authors) </GBAuthor>
               </GBReference_authors>
               <GBReference title>Identification, cDNA
cloning, and gene expression of soluble
                 starch synthase in rice (Oryza sativa L.)
immature seeds</GBReference_title>
               <GBReference_journal>Plant Physiol. 103 (2),
565-573 1993)</GBReference_journal>
<GBReference_medline>94302151</GBReference_medline>
<GBReference_pubmed>7518089</GBReference_pubmed>
             </GBReference>
            ......(file truncated due to reasons of
space).....
     <GBOualifier>
       <GBQualifier_name>translation</GBQualifier_name>
        <GBQualifier_value>MATAAGMGIGAACLVAPQVRPGRRLRLQRVR-
RRCVAELSRD
           .....(truncated amino acid sequence).....
        QGTGWAFSPLTIEKNAVGIADGNFDIQGTQVLLGGSNEARHVKRLYMG-
PCRLTV
             </GBQualifier_value>
      </GBOualifier>
        .....(file truncated due to reasons of
        <GBSeq_sequence>CGCGACTCAGCCCACTCCTCTCTCCACCAC-
CACCACCACC
               ....(truncated nucleotide sequence)....
          \tt GTGAAGACTAAATAGTGTTTGGAAGCTGTAGCTACTGCGATGTCAAGT-
GTCAA
         </GBSeq_sequence>
    </GBSeq>
  </GBSet>
 <?xml version="1.0" encoding="UTF-16" ?>
  <xsl:stylesheet version="1.0"</pre>
```

```
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
 <xsl:include href="text.justify.xslt" />
 <xsl:output method="text" encoding="UTF-16" />
 <xsl:strip-space elements="*" />
  <xsl:template match="/">
 Rice Soluble Starch Synthase
  Prim_acc Mol_type Clone_size Protein_# Protein_name
Amino_Acid Mol_Wt
  |-----|-----|------|------|-----|
-----|-----|
  <xsl:call-template name="table-entries">
 <xsl:with-param name="file" select="'file1.xml'" />
  <xsl:with-param name="n" select="1" />
  </xsl:call-template>
  </xsl:template>
   <xsl:template name="table-entries">
  <xsl:param name="file" />
  <xsl:param name="n" />
  <xsl:param name="amino_acid"</pre>
select="document($file)//GBQualifier_value[preceding-sib-
ling::GBQualifier_name/text()='translation']" />
  <xsl:call-template name="text-justify">
  <xsl:with-param name="value"</pre>
select="document($file)//GBSeq_primary-accession" />
  <xsl:with-param name="width" select="11" />
  <xsl:with-param name="align" select="'center'" />
  </xsl:call-template>
  <xsl:text>|</xsl:text>
  <xsl:call-template name="text-justify">
  <xsl:with-param name="value"</pre>
select="document($file)//GBQualifier value[preceding-sib-
ling::GBQualifier_name/text()='mol_type']" />
  <xsl:with-param name="width" select="11" />
  <xsl:with-param name="align" select="'center'" />
  </xsl:call-template>
  <xsl:text>|</xsl:text>
  <xsl:call-template name="text-justify">
  <xsl:with-param name="value" select="string-length(docu-</pre>
ment($file)//GBQualifier_value[preceding-
sibling::GBOualifier name/text()='translation'l)"/>
  <xsl:with-param name="width" select="10" />
  <xsl:with-param name="align" select="'center'" />
  </xsl:call-template>
  <xsl:text>|</xsl:text>
......(truncated to save space)......
   <xsl:call-template name="text-justify">
  <xsl:with-param name="value" select="89*string-</pre>
length(translate($amino_acid,translate($amino_acid,'A',''),'
'))+174*string-
   ....(truncated to save space).....
length(translate($amino_acid,translate($amino_acid,'V',''),'
'))"/>
  <xsl:with-param name="width" select="10" />
  <xsl:with-param name="align" select="'center'" />
  </xsl:call-template>
  <xsl:text>|</xsl:text>
  <xsl:text />
   <xsl:if test="$n<25">
   <xsl:call-template name="table-entries">
  <xsl:with-param name="file" select="concat('file', $n + 1</pre>
 '.xml')" />
  <xsl:with-param name="n" select="$n+1" />
  </xsl:call-template>
  </xsl:if>
  </xsl:template>
  </xsl:stvlesheet>
```







### Look Ma Bell - No Hands!

#### VoiceXML, X+V, and the mobile device

The emerging world without wires has fostered a growing number of small and mobile devices (everything from PDAs to smart phones) capable of accessing data and running applications. The trouble is, while devices are getting smaller, human hands and fingers are not.

To assist users in managing their devices, user interface designers have begun to combine the traditional keyboard-input model with such interactive technologies as voice-directed input. This type of interaction, in which the user has more than one means of accessing data in his or her device, is sometimes called multimodal interaction. It is fast becoming the norm in the world of wireless mobile computing.

If asked, most developers will cite speed and efficiency as the main reasons for developing multimodal interfaces. Parallel input - for example, the ability to both key in commands and voice them - allows users to more quickly access and respond to information delivered by their devices. In fact, multimodal systems don't just enable faster interactions, they also add value to the overall experience of interaction. Multimodal interfaces allow more room for user preference (giving users a choice of how they interact with the system) and reduce the overexertion that can result from single-modality interaction. Being able to switch between modes of interaction can lead to a lower incidence of error (because

users can choose the mode most suited to different activities), as well as easier error recovery. And, finally, multimodal interfaces have the capacity to accommodate a wider range of tasks and environments.

Speech adds tremendous value to small mobile devices, but in tandem, mobility and wireless connectivity are also moving computing into new physical environments. Wireless networks now provide connectivity anywhere and anytime. Connecting mobile devices to the network links mobile computing to back-end data anywhere and anytime. If the need for multimodal interaction extends to the network, then the Internet needs new technologies and standards to enable that functionality. Increasingly, Web developers are seeking ways to turn existing visually oriented Web pages into multimodal ones. And that's where X+V comes in.

The XHTML+Voice profile brings spoken interaction to standard Web content by integrating the mature XHTML and XML-Events technologies with XML vocabularies developed as part of the W3C Speech Interface Framework. The profile includes voice modules that support speech synthesis, speech dialogs, command and control, and speech grammars. Voice handlers can be attached to XHTML elements and respond to specific Document Object Model (DOM) events, thereby reusing the event model familiar to Web developers. Voice interaction features are integrated with XHTML and CSS and can consequently be used directly within XHTML content.

X+V promises to deliver the feature set, flexibility, and ease of use that developers need to write one application that supports visual-only, voice-only, and multimodal interaction. The versatility of the Web and XML is reflected in the fact that X+V nicely integrates VoiceXML into the Web by marrying it with XHTML. X+V brings voice markup to the presentation layer, allowing you to speechenable each component of the application interface.

#### **Combining Voice** and Visual Markup

Visual markup tells a Web browser what you want the user interface to look like and how you want it to behave when the user types, points, or clicks. Similarly, voice markup tells the Web browser what you want it to do when the user speaks to it. For visual markup, the browser uses a graphics engine; for voice markup, the browser uses a speech engine.

While both X+V and SALT use W3C standards for grammar and speech synthesis, only X+V is based entirely on standardized languages. X+V's modular architecture makes it very simple to separate an X+V application into different components. As a result, X+V applications can be coded in parts, with experts in voice programming developing voice elements and experts in visual programming developing visual ones. X+V's modularity also makes it adaptable to standalone voice application development. VoiceXML used in an X+V application can be reused inside a stand-alone

ior technical staff member He has been responsible for a variety of research and development projects relat ed to man-machine interfaces, graphics, network computing, and user-inter-

Author Bio

Les Wilson is an IBM sen-

face technology. Les is cur rently the multimodal architect for IBM's Pervasive Computing VoiceXML application. SALT's reliance on the containing environment makes it very difficult to separate out its coding functions, and also makes the language insufficient to the task of stand-alone application development.

Richness is another factor that differentiates the two languages. Whereas SALT defines three tags - Prompt, Listen, and Bind - as its tag set for speech, X+V is based on the mature and tested VoiceXML standard. Because it uses VoiceXML's Form construct for its speech tag set, X+V includes all the utility of "prompt, listen, and bind," and more.

Just as visual markup specifies the visual interface items, voice markup specifies the voice interface items. Speech-enabling an application interface is a matter of first breaking the visual interface into its basic components (for example, an input field for a time of day and a checkbox for "a.m." or "p.m."), creating snippets of voice markup for each component, and then associating the snippets to the existing visual markup for each component. Consider the following questions:

- What words should the speech engine speak or synthesize?
- What words and phrases should the speech engine listen for?
- · What should the browser to do if the speech engine doesn't recognize a word or phrase?
- · What will be the result of the speech engine recognizing a word or phrase that has been spoken?

#### **Correlating Voice** and Visual Input/Output

Given an application's visual markup plus a collection of voice markup snippets, you have almost everything you need to create the presentation layer of a multimodal Web application. In fact, the only thing you still need is a way to tell the browser which snippets of voice markup go with which visual elements, and (because a speech engine can only have one snippet active at a time) when to activate each snippet of voice markup.

Given that the Web application environment is event-driven, X+V incorporates the DOM eventing framework used in the XML-Events standard. Using this framework, X+V defines the familiar event types from HTML such as "on mouse-over" or "on input focus" to create the correlation between visual and voice markup. Using XML-Events provides X+V with a uniform and standards-based eventing model that enables event integration between XML languages.

#### Separate Files and Reuse

Because all the parts of X+V are XML-compliant, the voice markup can be packaged in two ways: in the same file as the XHTML or in separate files. Separating voice markup from visual markup gives you more flexibility in developing your applications. For example, you can develop the voice markup separately from the visual markup and combine the two

"Most developers will cite speed and efficiency as the main reasons for developing multimodal interfaces"

Another advantage of keeping the files separate is reuse, such as the ability to reuse snippets of VoiceXML in numerous XHTML pages. In the example of a flight-reservation application, when users make the reservation they will be asked if they want a one-way, round-trip, or multi-leg reservation. For each answer, the system will call up a different form. While the three forms differ with regard to the type of trip desired, each one has the same departure city. If you have separated the voice snippet for the departure city you can reuse it in each of the three different XHTML forms, or containers.

The final advantage of keeping the VoiceXML separate from the XHTML is that it allows the snippets of VoiceXML to be reused in containers other than XHTML. In this case, X+V can utilize the VoiceXML notion of documents and forms, wherein a VoiceXML document contains one or more forms. You already know that VoiceXML forms can be linked to XHTML to create multimodal applications. But such forms can also be stitched together in a VoiceXML document (or container) to create voiceonly applications. The end result is that you can (by reuse) create a single application that simultaneously supports multimodal browsers, GUI-only browsers, and voice-only systems such as IVRs.

#### Conclusion

X+V is the latest addition to the XML family of technologies for user interface development. Whereas XHTML is for developing visual interfaces, and VoiceXML focuses entirely on voice-based development, X+V is a hybrid, dedicated to developing multimodal application interfaces. X+V is particularly well suited to wireless development, where developers are faced with small visual interfaces and increasing user demand for voice input and output.

X+V's foundation in existing XML standards lends it tremendous strength and versatility. Interfaces developed using X+V are portable to a wide range of applications and development environments, can be easily developed in teams, and are highly scalable over time. Developers working with X+V can access the numerous resources that come with a welldeveloped standard such as XML. X+V also takes developers out of the loop of learning a new development language such as SALT, or adapting to the constraints of a more visually oriented development environment. Perhaps best of all, X+V does not require training invoice user interfaces or linguistics to operate; a basic knowledge of XML and related standards is sufficient to get started.

LESW@US.IBM.COM

WSJ: **NEWS** 

#### Strikelron Launches the Strikelron **Web Services Business Network**

(Research Triangle Park, NC) - StrikeIron,



Inc., has released an online solution that makes it easier and faster

to work with Web services-the StrikeIron Web Services Business Network. This "software-as-aservice" is an online Web services network that provides an integrated set of services and graphical tools for working with and publishing Web services. New integrated, graphical visualization capabilities combined with dynamic invocation technology essentially liberates the user from the technical intricacies of Web services. The results are an easier and faster understanding of a Web service with no additional software or time consuming programming or compiling.

The WSBizNet is currently available at www.strikeiron.com. Standard services are free and include searching the Business Directory and use of the online Analyzer, Knowledge Base, and Performance Graphing. Pricing is subscription based for premium Web services, monitoring, private directories, hosting, and premium products. www.strikeiron.com

#### **Sana Security and Service Integrity** Team to Offer Intrusion Prevention

(San Francisco) - Sana Security, Inc., a



provider of host-based intrusion prevention software (HIPS), has

announced a strategic alliance with Service Integrity, Inc., a developer of XML/Web services monitoring, analysis, and visibility software. Interoperability between the products creates a best-of-breed solution that can provide large organizations with real-time application security, forensic analysis and regulatory compliance for their Web services infrastructures.

The joint offering will ensure substantially greater levels of security and availability for Windows-based Internet Information Services (IIS) servers from unknown attacks, as well as delivering forensic and compliance reporting capabilities for identifying the source of potential and attempted security breaches.

Applicable to a wide range of customers,



infrastructure and security scenarios, the alliance combines

Sana's Primary Response HIPS product with Service Integrity's SIFT software, offering value to large organizations in financial services, health care, and government. The monitoring and security solution is designed to help enterprises prevent intrusions, protect sensitive data and provide complete visibility into underlying XML-based application, server, and network interactions related to security events. www.serviceintegrity.com www.sanasecurity.com

#### OpenLink Releases **Universal Server Platform**

(Burlington, MA) - OpenLink Software, Inc.,



a leader in the development and deployment of secure, high-per-

formance database connectivity middleware, has announced the availability of Virtuoso 3.5, the company's universal server platform for service-oriented and event-driven applications.

Designed for the emerging event-driven and service-oriented executable Web, Virtuoso 3.5 comprises several traditionally distinct server features in a single server solution that includes a Web services platform, object-relational database (SQL and XML), replication server, Web application server, and more. It provides an adaptable architecture that allows organizations to cost-effectively transition to real-time enterprises, leveraging the newest technologies on top of existing or legacy infrastructure. www.openlinksw.com/virtuoso

#### **Dralasoft Workflow** Selected By Centrata

(Westminster, CO) - Dralasoft, Inc., an innovator of Java technology for business process



management (BPM), has announced that Centrata, Inc., a

provider of IT service delivery management solutions, has chosen Dralasoft Workflow BPM software to enhance the workflow functions of its flagship Centrata Service Delivery Management Suite. Dralasoft Workflow, which will be private-labeled in select modules of the suite, will provide increased efficiency and ensure best-practices compliance for Centrata's enterprise customers, nearly all of whom are in the Fortune 500.

Centrata Service Delivery Management (SDM) Suite helps large enterprise IT operations groups transform from reactive, ad hoc "job shops" to streamlined, service-oriented product houses. It does this by delivering a full service delivery management suite based on an industry-leading service catalog. Centrata's service catalog captures policies, standards and other critical characteristics including pricing and service levels of the services being offered.

Dralasoft Workflow reduces the complexity and cost of integrating business processes into various mission-critical business functions. The solution is comprised of three modules:

Dralasoft Workflow Engine, the runtime component; Dralasoft Workflow Studio, a drag-anddrop design interface for workflow development; and Dralasoft Workflow Manager, the application's monitoring and analysis tool.

Centrata Govern and Centrata Fulfill, the two Centrata SDM modules into which Dralasoft Workflow will be added, both perform key operational tasks of the SDM suite. Centrata Govern is an application module providing end-to-end definition, management and enforcement of IT service and project standards, processes and service level agreements. Centrata Fulfill provides optimized fulfillment of IT requests based on business criticality and delivered via best-practice and industry standard workflows.

The new workflow functionality will be backwards-compatible with existing Centrata SDM deployments by porting existing workflow definitions to the Dralasoft format, and by abstracting out the invocation of Dralasoft Workflow Engine APIs for runtime applications. www.dralasoft.com

#### **Active Endpoints Announces Beta** Program for ActiveWebflow

(Shelton, CT) - Active Endpoints,

Inc., an independent provider of services-oriented integration solu-

tions, has announced public beta availability of ActiveWebflow, a comprehensive suite of process management products based on the BPEL (Business Process Execution Language) specification.

Leveraging the growing acceptance of Web services as the leading enterprise integration architecture, ActiveWebflow users have an open, standards-based solution for automating process flows across systems and enterprises. ActiveWebflow Professional allows users to rapidly compose new services-oriented processes, generate standard BPEL process definitions, and execute those processes using the ActiveBPEL engine (or any BPEL 1.1 compliant engine).

ActiveWebflow is the first in a series of BPEL related products that Active Endpoints plans to release this year. The ActiveWebflow products will collectively deliver enterprise-class process management capabilities as part of the ActiveIntegration suite, a comprehensive family of services-oriented integration solutions. The beta version of ActiveWebflow Professional is available for immediate download at www.active-endpoints.com/awdownload.

**58** August 2004 www.wsi2.com

## Open Up!

Portals-Get what you want easily



#### **Ask the Experts**

Prolifics is 1 of only 3 WebSphere Service Providers retained by IBM. Open up with portals! With 26 years of industry expertise, Prolifics' certified WebSphere portal experts deliver a personalized, secure, unified view to your enterprise assets.

Our portal solutions build agility into your business — enabling you to reach new markets, achieve customer satisfaction beyond expectation, and ease collaboration within your enterprise. Our teams of specialists work together with your business experts from gathering the business requirements, translating them to a technical specification, to delivering the final production application.

#### **Portal Executive Assessment**

Prolifics experts help you articulate the value of portals to your business sponsors and perform a quick business review identifying portal benefits as it applies to your organization.

#### **Portal Workshops and Proof of Concepts**

Prolifics demonstrates valuable portal features and translates the breadth of your portal objectives into a step-by-step roadmap to meet your current and future needs.

#### Portal Jump Start

Prolifics' consultants work with you from the start to install, configure and jump start your team on using WebSphere Portal Software.

#### **Portal Development and Mentoring**

Prolifics' innovative approach to building portals drives down your overall cost and transforms the way you conduct business. We mentor and arm your team with the skills to manage, maintain and extend the portal environment.

#### **Prolifics**

Winner of 2003 IBM BUSINESS PARTNER LEADERSHIP AWARD

116 John Street, New York, New York 10038 800.675.5419 solutions4websphere@prolifics.com www.prolifics.com







**Virtuoso, from OpenLink Software,** is a cross platform Universal Server for the Enterprise.

It's a single server product offering for Object-Relational Data Management (SQL and XML), Enterprise Data Integration, Secure Web Services Composition, and Business Process Management.

Virtuoso achieves all of this without locking you into a specific operating system, programming language, or backend database engine.

Standards compliance and cross platform support make Virtuoso the obvious choice for cost-effective manifestation of the Real-Time Enterprise vision.

Empower your organization today – download a FREE evaluation copy from our Website or call us at 1 800 495 6322.

#### **Database Support**

Oracle, SQL Server, DB2, Sybase, Informix, CA-Ingres, Progress, MySQL, PostgreSQL, Virtuoso, and other ODBC or JDBC accessible databases.

#### Standards Support

SOL, ODBC, JDBC, XML, XSL-T, XQuery, XPath, XML Schema, HTTP, WebDAV, SOAP, WSDL, WS-Security, and UDDI, NNTP, POP3, SMTP, and SyncML.

#### Platform Support

Windows, UNIX (all major flavors), Linux, Free BSD, and Mac OS X

#### Runtime Environment Hosting

Microsoft .NET, Mono, Java, Perl, Python, PHP, and others.

#### Programming Languages

Any .NET bound language (C#, VB.NET, others), Java, C/C++, and others.

#### Web Application Runtime Hosting

ASP.NET, Java Server Pages, PHP, Perl, and Python.

www.openlinksw.com/virtuoso/

