NOW INCORPORATING XML JOURNAL

WebServices JOURNAL

.NET J2EE XML

January 2005 Volume 5 Issue 1

Deploying Large-Scale Interoperable Web Services Infrastructure

Issues that architects will frequently have to confront
pg. 10

METADATA EVOLUTION MANAGEMENT IN YOUR SOA

Managing the life cycle of your metadata

pg30

An Overview of the Java WSDP 1.5

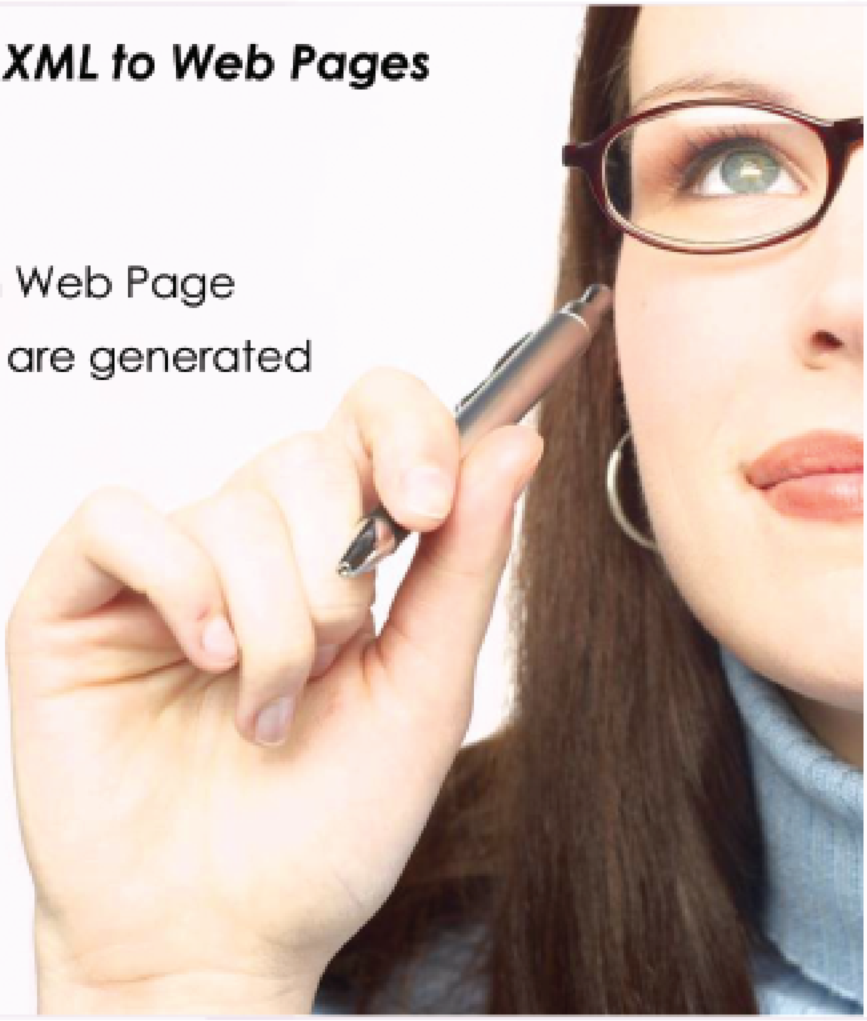
A rich set of utilities for the developer
pg. 20

Consuming Web Services

Why service orientation is a move in the right direction
pg. 24

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

The most cost-effective
route to Enterprise Agility
just became obvious !



JANUARY 2005 Volume 5 Issue 1

WebServices JOURNAL .NET J2EE XML

XML JOURNAL

Inside WSJ

WSJ: Features

JANUARY
BUZZ

SOAPtest®

Combining AEP Practices for Effective Error Prevention

As I mentioned in last month's Buzz, the AEP practices of coding standards and unit testing should be considered two sides of the same coin. Each practice can uncover problems that the other cannot. Consequently, to identify and prevent a wide range of software problems, it's important to perform *both* coding standard analysis and unit testing, not just one or the other.

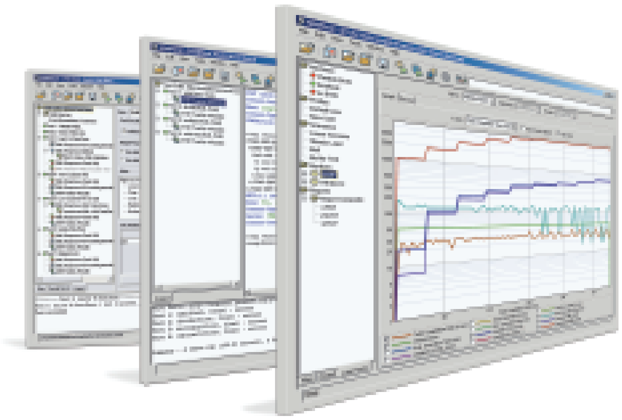
This month, I'd like to elaborate on how combining specific AEP practices is the most effective way to identify and prevent specific errors. For example, let's consider resource or memory leaks. Application-level runtime error detection and/or profiling can identify leaks, but typically not until late in the development process, when correcting the leak is typically time-consuming, difficult, and expensive. With the appropriate technologies, leaks could be detected at the unit level. However, unless the development team is required to close all resources that are opened within a unit, a leak found at the unit level might not indicate a true application-level leak.

One effective solution to this dilemma is to combine coding standard enforcement, unit-level leak detection, and application-level leak detection/profiling. First, implement a coding standard that requires that the same class or function which allocates a resource must also deallocate that resource. By checking compliance with this coding standard, you can ensure that the team's developers write code in a way that prevents leaks. If this coding standard does not make sense in all situations, individual exceptions can be documented and suppressed. You can then use unit-level leak detection to dynamically verify that leaks do not occur within the unit. After all units are developed in this manner, you can use application-level leak detection and/or profiling to verify whether leaks occur at the application level. When the team works in this manner, significantly fewer leaks reach the application level, where they are more time-consuming, difficult, and expensive to fix.

The Parasoft AEP Methodology is constructed to combine multiple practices in this supportive manner. It encourages development teams to determine which types of errors they need to prevent, then combine practices to create a "leak-proof" system for identifying and preventing those types of errors. For details about this methodology, visit **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.



SOAPtest® enables you to deliver better Web Services in less time.

The simple fact is, no other Web service testing product can do what SOAPtest can do. From verifying functionality and performance to ensuring security and interoperability, 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_WSJ 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_WSJ

Email: soaptest_WSJ@parasoft.com Call: 888-305-0041 x1209

PARASOFT®
We make software work.™

Features

- WSDL schema verification and compliance to standards
- Automatic test creation using WSDL and HTTP Traffic
- Data-driven testing through data sources (Excel, CSV, Database Queries, etc)
- Scenario-based testing through XML Data Bank and Test Suite Logic
- Flexible scripting with Java, JavaScript, Python
- WS-I Conformance: Basic Profile 1.0
- WS-Security, SAML, Username Token, X.509, XML Encryption, and XML Signature support
- WS-Security Interop testing emulator
- MIME Attachment support
- Asynchronous Testing: JMS, Parlay (X), SCP, and WS-Addressing support
- Windows Perfmon, SNMP, and JMX monitors
- Detailed Report generation in HTML, XML and Text formats
- Real-Time graphs and charts

Benefits

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

Platforms

Windows 2000/XP
Linux
Solaris

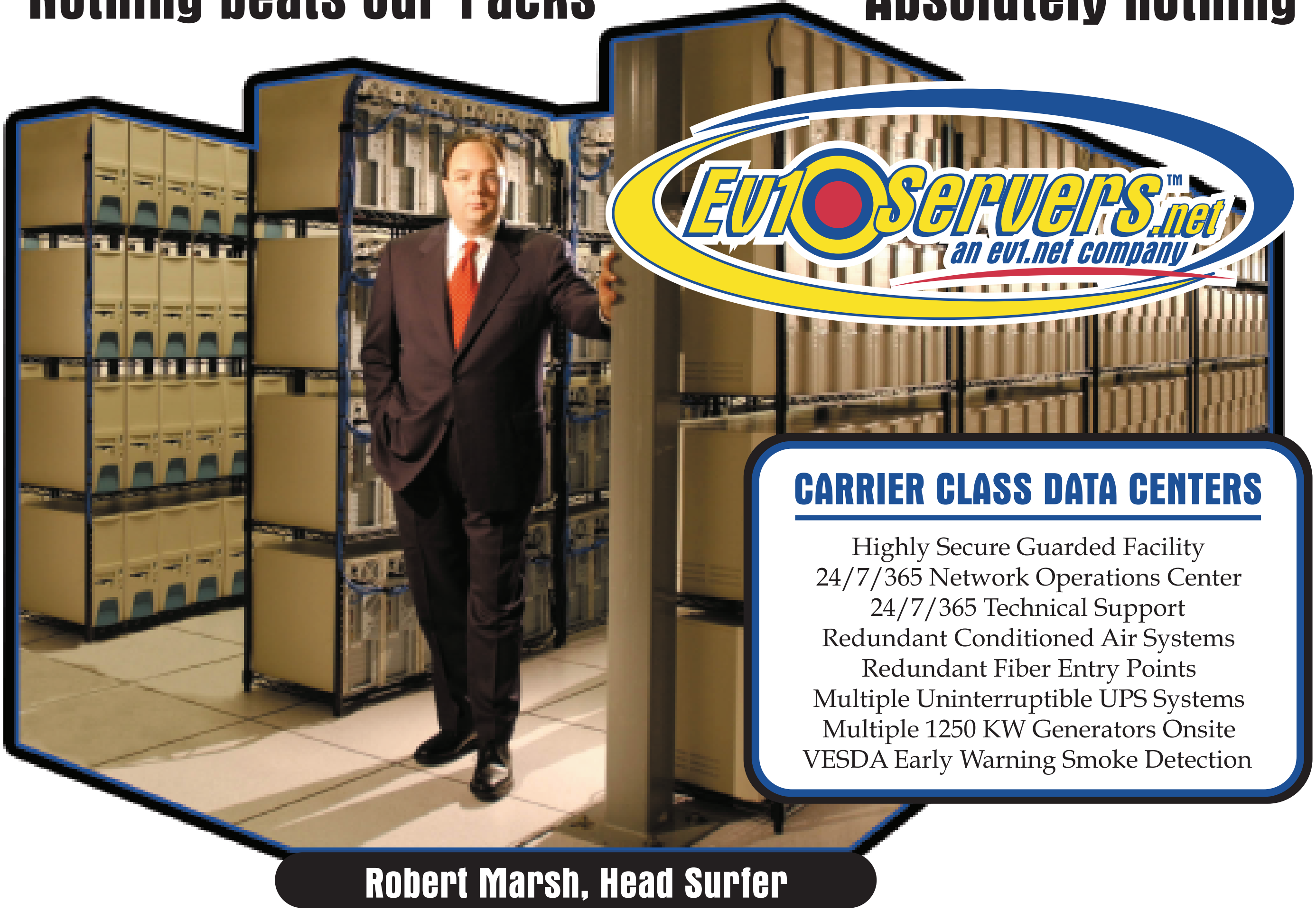
Protocol Support

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

Contact Info:

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

www.parasoft.com


Ev1Servers.net™
an ev1.net company

WebServices JOURNAL
.NET J2EE XML

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

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

EDITORIAL
EDITOR-IN-CHIEF
Sean Rhody sean@sys-con.com
XML EDITOR
Hitesh Seth
INDUSTRY EDITOR
Norbert Mikula norbert@sys-con.com
PRODUCT REVIEW EDITOR
Brian Barbash bbarb-ash@sys-con.com
.NET EDITOR
Dave Rader davidr@fusiontech.com
SECURITY EDITOR
Michael Mosher wsjsecurity@sys-con.com
RESEARCH EDITOR
Bahadir Karuv, Ph.D Bahadir@sys-con.com
TECHNICAL EDITORS
Andrew Astor aastor@webmethods.com
David Chappell chappell@sonicsoftware.com
Anne Thomas Manes anne@manes.net
Mike Sick msick@sys-con.com
Michael Wacey mwacey@csc.com
EXECUTIVE EDITOR
Gail Schultz gail@sys-con.com
EDITOR
Nancy Valentine nancy@sys-con.com
ASSOCIATE EDITOR
Jamie Matusow jamie@sys-con.com
ASSISTANT EDITOR
Natalie Charters natalie@sys-con.com
Seta Papazian seta@sys-con.com
ONLINE EDITOR
Martin Wezdecki martin@sys-con.com
Seta Papazian seta@sys-con.com

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

CONTRIBUTORS TO THIS ISSUE
Bill Appleton, Brian Barbash, Jim Clune, Paul Colton, Jeff Dexter, Jim Gabriel, Mike Lehmann, Sean Rhody, Davanum Srinivas, Deepak Vohra, Sanjiva Weerawarana

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

The Last Mile, Revisited

WRITTEN BY
SEAN RHODY

A couple of issues ago, I railed about the lack of Web services–based technology to provide the last hurdle of any service – representation in a user interface. While many Web services will only be accessed programmatically, still more are destined to find their way, one way or another, onto the screens of computers in front of human beings.

I got a number of comments on the article, the most interesting one being from a reader who thought I was joking – after all, there already was a last mile technology called the World Wide Web. And he wasn't being facetious either, he was serious.

I can see his point – the Web is certainly ubiquitous. However there are a number of issues that accompany it, issues I guess I didn't make plain enough when I was calling for a different technology to be developed.

HTML is not a programming language. It was never meant to be, so there's certainly no blame to be placed on it from that perspective. At the same time, though, instead of developing a programming language neutral-presentation language, and clients that could consume such a language, the industry has come up with various ways to spew out HTML and accept data back from it.

There are problems with HTML that make it a poor choice for some applications. Data is not separate from user interface; it is all part of a message. Effectively, the interface is sent every time a call is made to the Web server. There is no two-way conversational state – in fact, state is not maintained at all by HTML. There's no way to send a message to the interface, so things like real-time feeds have to be accomplished by some other mechanism.

The visual constructs of HTML are not as sophisticated as those available to the native user interfaces (Windows, or KDE, etc.). Multiple documents, grid controls, sophisticated graphics, charts, and other unique controls are all beyond the reach of basic HTML. Anything addressing those needs is custom, and not part of HTML.

Additionally, not everyone wants to expose his or her applications using a Web browser or over the Internet either. Various reasons are cited, but they include security, performance, firewall issues, or a desire for a rich user interface.

What I was pushing for is a new paradigm, one that Microsoft is testing the waters on. I would like a platform-neutral rendering language supported by a platform-specific implementation (it could be a browser, but it could also be the native user interface itself) that would allow interpretation of the platform-neutral presentation language. It would provide the ability to maintain state, allow for strong security, provide all the control depth that the native UI provides (perhaps with certain tweaks to provide for modal operation, for example). It should allow for bidirectional, asynchronous secured conversations utilizing Web services as a mechanism. It would not need to download the presentation with every request – that would be part of the presentation definition as to when new presentation elements need to be requested (i.e., a table would not need to download its format, just the data if it had changed).

None of this is easy. One of the reasons it has been delayed in the realm of Web services is that it requires new technologies, new paradigms, and significantly, new cooperation.

I mentioned that Microsoft is attempting something of this nature, although limited to the Windows platform. The first step is a descriptive language that will provide for interface creation and element binding. They're pursuing this for the next version of Windows. Obviously this will be an XML-based language. Hopefully, the first step towards a truly platform-neutral presentation will be the donation of this technology to the open source community so that platforms beyond Windows can benefit.

This month's focus is on consuming Web services. Currently there is a number of ways to consume Web services. It's my hope that a few years from now, I'll be telling you about the new, platform-neutral way to consume them. Until then, we'll limp along with HTML and proprietary stop gaps. ⓔ

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

WebServices JOURNAL
.NET J2EE XML

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.com
VP, SALES & MARKETING
Miles Silverman miles@sys-con.com
ADVERTISING DIRECTOR
Robyn Forma robyn@sys-con.com
ADVERTISING SALES
Dennis Leavey dennis@sys-con.com
ADVERTISING MANAGER
Megan Mussa megan@sys-con.com
ASSOCIATE SALES MANAGERS
Kristin Kuhnle kristin@sys-con.com
Dorothy Gil dorothy@sys-con.com
Kim Hughes Kim@sys-con.com

SYS-CON EVENTS
PRESIDENT, SYS-CON EVENTS
Grisha Davida grisha@sys-con.com
NATIONAL SALES MANAGER
Jim Hanchrow jimh@sys-con.com

CUSTOMER RELATIONS/JDJ STORE
CIRCULATION SERVICE COORDINATORS
Edna Earle Russell edna@sys-con.com
Linda Lipton linda@sys-con.com
Monique Floyd monique@sys-con.com

SYS-CON.COM
VP, INFORMATION SYSTEMS
Robert Diamond robert@sys-con.com
WEB DESIGNERS
Stephen Kilmurray stephen@sys-con.com
Matthew Pollotta matthew@sys-con.com

ACCOUNTING
FINANCIAL ANALYST
Joan LaRose joan@sys-con.com
ACCOUNTS PAYABLE
Betty White betty@sys-con.com
ACCOUNTS RECEIVABLE
Steve Michelin smichelin@sys-con.com

SUBSCRIPTIONS
SUBSCRIBE@SYS-CON.COM
1-201-802-3012
1-888-303-5282
For subscriptions and requests for bulk orders,
please send your letters to Subscription Department
Cover Price: $6.99/issue
Domestic: $69.99/yr (12 issues)
Canada/Mexico: $89.99/yr
All other countries: $99.99/yr
(U.S. Banks or Money Orders)

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

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

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

SYS-CON MEDIA

Web Services ... the Apache Way

WRITTEN BY
SANJIVA WEERAWARANA
& DAVANUM SRINIVAS

The Apache Web Services Project is an umbrella project consisting of many projects. The scope of projects goes from SOAP implementations to a UDDI implementation to a host of WS-* implementation efforts. The developer community is a vibrant one, coming from a mixture of large companies, small companies, and independent consultants. The community is also global, with participants originating from the US, Europe and Asia.

In this article we briefly introduce some of the activity around the Apache Axis project, the flagship project of the Apache Web Services activities.

Axis

Apache Axis started with a total rewrite of the Apache SOAP implementation, which was the first SOAP 1.1 implementation. Apache Axis now has both Java and C++ incarnations. The Axis/Java product is a very mature product that is widely used by many vendors for SOAP enablement of their systems. In comparison, Axis/C++ is only about a year old but is rapidly gaining maturity, quality, and users. Axis/Java and Axis/C++ share a common base architecture, which at the time of its design (in 2001-02) was industry leading.

There is now a new effort, called Axis2, to redesign and totally reimplement both Axis/Java and Axis/C++ on a new architecture. Evolving from the now standard "handler chain" model that Axis1 pioneered, Axis2 is developing a more flexible pipeline architecture that can be managed and packaged in a more organized manner. This new design acknowledges the maturing of the Web services space – in terms of new protocols such as WS-ReliableMessaging, WS-Security, and WS-Addressing that are built on top of the base SOAP system. At the time Axis1 was designed, while it was fully expected that other protocols such as WS-ReliableMessaging would be built on top of it, there was no proper extension architecture defined to enable clean composition of such layers. Thus, one of the key motivations for Axis2 is to provide a clean and simple environment like Apache Sandesha and Apache WSS4J to layer on top of.

Another driving force for Axis2 is the move away from RPC-oriented Web services towards more document-oriented, message style asynchronous service interactions. The Axis2 project is centered on a new representation for SOAP messages called AXIOM (AXIs Object Model), which is composed of two parts: a complete XML Infoset representation and a SOAP Infoset representation on top of that.

The XML Infoset representation provides a JDOM-like simple API but is built on a deferred model via a StAX-based (Streaming API for XML) pull parsing API. A key feature of AXIOM is that it allows you to stop building the XML tree and to just access the pull stream directly, enabling both maximum flexibility and maximum performance. This approach allows us to support multiple levels of abstraction for consuming and offering Web services: using plain AXIOM, using generated code and statically data-bound data types and so on.

At the time of Axis1's design, RPC-style, synchronous, request-response interactions were the order of the day for Web services. Today service interactions are much more message oriented and exploit many different message exchange patterns. The Axis2 engine architecture is careful to not build in any assumptions of request-response patterns to ensure that it can be used easily to support arbitrary message exchange patterns.

Conclusion

Axis2 is still in its infancy and many of the key decisions are still being hashed out. As we design the Axis2 core there are efforts underway to ensure that projects such as Apache Sandesha can be built on top of it without any difficulty. We welcome all contributions to this key project. We are working towards an informational milestone release soon to encourage the larger community to examine the progress so far and provide feedback on our approach.

References

- *Apache Web Services Project:* http://ws.apache.org/
- *Apache Axis Project:* http://ws.apache.org/axis/
- *Apache Axis2 Project:* http://wiki.apache.org/ws/FrontPage/Axis2
- *Apache Sandesha Project:* http://ws.apache.org/ws-fx/sandesha
- *Apache WSS4J Project:* http://ws.apache.org/ws-fx/wss4j

About the Author

Sanjiva Weerawarana, PhD, a member of the Apache Software Foundation, is chairman and executive director of the Lanka Software Foundation in Sri Lanka and a research staff member for IBM Research.
■■■ sanjiva@apache.org

Davanum Srinivas is vice president of Apache Web Services at the Apache Software Foundation; and a senior architect at Computer Associates.
■■■ dims@apache.org

Is your SOA really agile?

Building, testing and maintaining SOA solutions requires a flexible and collaborative approach to diagnostics. Finger pointing and confusion occur when all parties do not have a complete and common understanding of a problem. Mindreef® SOAPscope® 4.0 connects developers, testers, support personnel, and operations by combining the testing and diagnostic tools for each discipline and making it easy to share complete problem data. SOAPscope is the only end-to-end diagnostic system for Web services.

Developers • Testers • Operations • Support

Mindreef SOAPscope 4.0

Test - Agile testing approach for both developers and testers allows what-if scenarios without coding or scripting. Includes the industry's most complete interoperability suite and the flexibility to really test the edge cases.

Capture - Easily capture and assemble complete problem data into SOAPscope's workspace. Capture SOAP messages off the wire, test cases, or reproduced problems. Capture a snapshot in time of a WSDL including all related resources.

Share - Exported workspaces support sharing among team members preserving complete problem data including messages, WSDLs, and notes. Sharing workspaces is a great way to support web service users.

Solve - Diagnostic suite of analysis, comparison and visibility tools simplifies the understanding of complex XML, yet are powerful enough for the industry's most protocol-savvy experts.

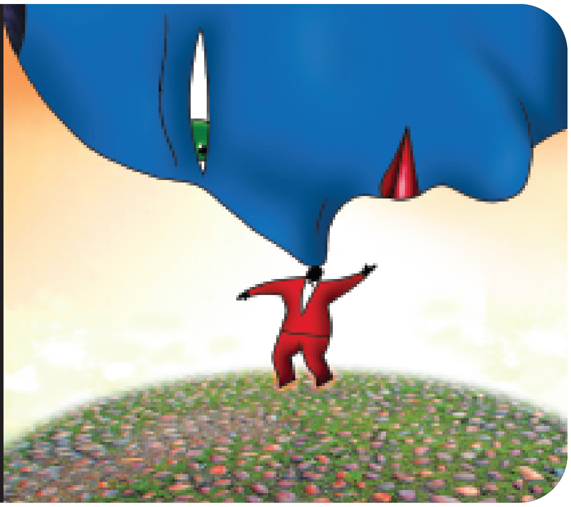
"SOAPscope has been invaluable in testing and diagnosing our enterprise scale Web Services. The new Workspace feature in SOAPscope 4.0 is an excellent way to support customers and collaborate as a team. A must have for every WS-Developer"

Simon Fell
Senior Member of
Technical Staff at
salesforce.com and
author of PocketSOAP

Try SOAPscope **FREE** at www.Mindreef.com/tryout

Mindreef, Inc.
www.Mindreef.com

Deploying Large-Scale Interoperable Web Services Infrastructures



Issues that architects will frequently have to confront

Web services have moved beyond the experimental stage in many organizations and are now becoming the foundation of numerous service-oriented architectures. Thus, architects are concerned about best practices for building, deploying, and maintaining a large-scale interoperable Web services infrastructure.

WRITTEN BY
MIKE LEHMANN

In one sense, Web services applications are like other applications. They represent a code base developed by a team of engineers. This code base needs to go through a methodological development life cycle, followed by testing and quality assurance, before it is finally released.

Frequently, however, Web services are not new applications at all, but rather carefully crafted message-based interface layers on top of existing systems and applications. New applications may be composed of Web services, or the services may be orchestrated into new business processes.

Given this evolutionary approach to application design and deployment, Web services and applications and business processes using Web services have a different set of provisioning and management concerns that enterprise architects must consider. This article provides a high-level assessment of four key areas that need to be considered.

Interoperability

The underlying value proposition of Web services is operational efficiency, provided by consistent, standards-based mechanisms that link together heterogeneous business applications. Such systems typically originate on a variety of legacy, host-based, and modern architectures, each with different Web service–enabling capabilities.

Although interoperability best practices for Web services are becoming better understood for lower-level protocols like SOAP and WSDL, issues at the more recently emerged quality-of-service level for security, reliability, and policy are not as well understood. As such, service enabling these applications in a consistent and maximally interoperable fashion is a key concern for enterprise architects.

Performance

Web services are fundamentally a message-based approach to integration and focus on moving XML-based SOAP documents on both public and private networks. Such applications have very different performance characteristics than traditional multitier transactional systems that use binary protocols. Architects must take into account new throughput, latency, network, and client concerns layered on top of existing applications that have frequently been tuned for different usage characteristics.

Quality of Service

Coinciding with the emergence of Web applications was an increase in the number of users for back-end infrastructures. Web services add a new category of client: the programmatic client that has the potential to increase the number of "users" and messages flowing through the environment by another order of magnitude. This new usage model can require new approaches to reliability, availability, and scalability. Furthermore, message-based systems bring new quality-of-service concerns regarding reliable messaging and security of messages coming into the infrastructure.

Manageability

Web services are typically used for application-to-application communication rather than for end user–facing applications. As such, the visibility of a Web services infrastructure to the end user and even the operational staff is less apparent because it is frequently the hidden glue that ties operational systems together. It is critical that architects design with management visibility in mind, taking into account the uniqueness of Web services from a monitoring, diagnostics, service life cycle–management, and service-level agreement perspective.

Let's take a closer look at each of these topics.

Interoperability

A number of key questions regarding interoperability can drive an architect's strategy. Are your Web services internally focused so that you will have control over the clients and the tools that will be used with them? Are your Web services external facing and subject to arbitrary clients and tools? How sophisticated are your users? Are they integration architects using Web

services development tools, or are they end users using Web service–enabled desktop and portal productivity software? These are basic questions, but they direct how you might tackle interoperability.

You can begin by following best interoperability practices for developing Web services.

> “Web services are typically used for application-to-application communication rather than for end user–facing applications”

Development

One way to tackle the issue is by publishing Web services bottom-up (i.e., taking existing applications and simply wrapping programmatic APIs as Web services). However, the top-down approach is more interoperable (i.e., modeling the message using XML Schema and designing the interface first in WSDL so the public contract and message definition works for both the client and server implementation).

The top-down approach is more interoperable for several reasons. For one thing, bottom-up approaches not only tightly couple consumers to existing APIs, but they often pollute the WSDL contract with noninteroperable, language-specific interface and message artifacts. A simple Java example that can be difficult to interoperate within .NET is the frequent usage of collections to move data structures between application tiers. From .NET, a common example is ADO.NET data sets, which are specific to Microsoft's platform. Avoiding language-specific types and starting with interface and message definitions can lead to a much higher likelihood of interoperability.

Third-party testing tools can validate best practices for interoperability. One of the most recognized groups focused on this is the Web Services Interoperability Organization (WS-I), which includes companies such as Oracle, IBM, Microsoft, and others. WS-I has created a set of best practices called WS-I Basic Profile 1.1 that describes how best to craft WSDL and SOAP messages so that Web services conforming to these rules have a maximum chance of achieving interoperability. These rules have been codified into a set of testing tools that can be run against Web service WSDLs and SOAP message exchanges to determine if those practices have been followed.

Testing

Conformance to WS-I does not necessarily guarantee interoperability. Rather, it is an indicator that your Web services are highly likely to be interoperable. Sometimes some older Web services infrastructures may not support the default message types required by WS-I: document/literal and rpc/literal. Or, sometimes some service providers are unable to upgrade their infrastructures to generate WS–I compliant services.

Practically speaking, testing with the actual target Web services clients is the only way to prove real interoperability. This enables architects to validate their own internally developed Web services as well as to validate that their preferred toolkits work with non-WS-I compliant Web services. An analogy can be made to Web application development. Just as many organizations test their Web applications with multiple browsers to ensure HTML compatibility, it is frequently incumbent on the Web services provider to try multiple client environments with their Web services end points.

The degree to which the interoperability testing has to be done depends on how you answer the initial questions regarding usage. Externally available Web services have a higher testing bar associated with them due to the unanticipated nature of clients. Internally available Web services may have a lower testing bar required if the client environment is more tightly managed and homogenous.

A very common practice that has emerged is for Web services providers to offer sample clients in popular programming languages: Java, C#, Visual Basic, Perl, and PHP. Examples of widely used services taking this approach include Amazon, Google, and eBay. This approach may seem to indicate that the promise of the interoperability of Web services has yet to be reached. However, it should be seen simply as a sign of a maturing industry as architects take short-term pragmatic steps toward ensuring interoperability and, as a by-product, usability.

In addition, a Web services provider may make a conscious decision to create a poorly interoperable implementation. If such a situation arises, the designer should provide some workarounds for service consumers.

Workarounds

Just as database architects and middle-tier object modelers often relax design constraints for application-specific reasons, Web services providers may consciously design service interfaces that are not maximally interoperable. Some designers prefer tight coupling to back-end systems for performance reasons. Others really want non-schema-based object models represented in the message exchanges, or moved “over the wire,” for productivity reasons. Sometimes using SOAP over HTTP just does not meet the performance requirements of the target application.

In these cases, it is typically incumbent on the Web services provider to offer recommendations to clients on how to use these services. Common approaches beyond providing sample working clients include the following:

1. Working in a homogenous client/server environment (Web services toolkits invariably are symmetrically interoperable with themselves)
2. Providing custom serializers for proprietary types that can be plugged into third-party toolkits
3. Describing how to use handlers or interceptor architectures provided by most toolkits to transform messages into a usable form at the client or end point
4. Providing code samples of how to parse the raw XML SOAP message

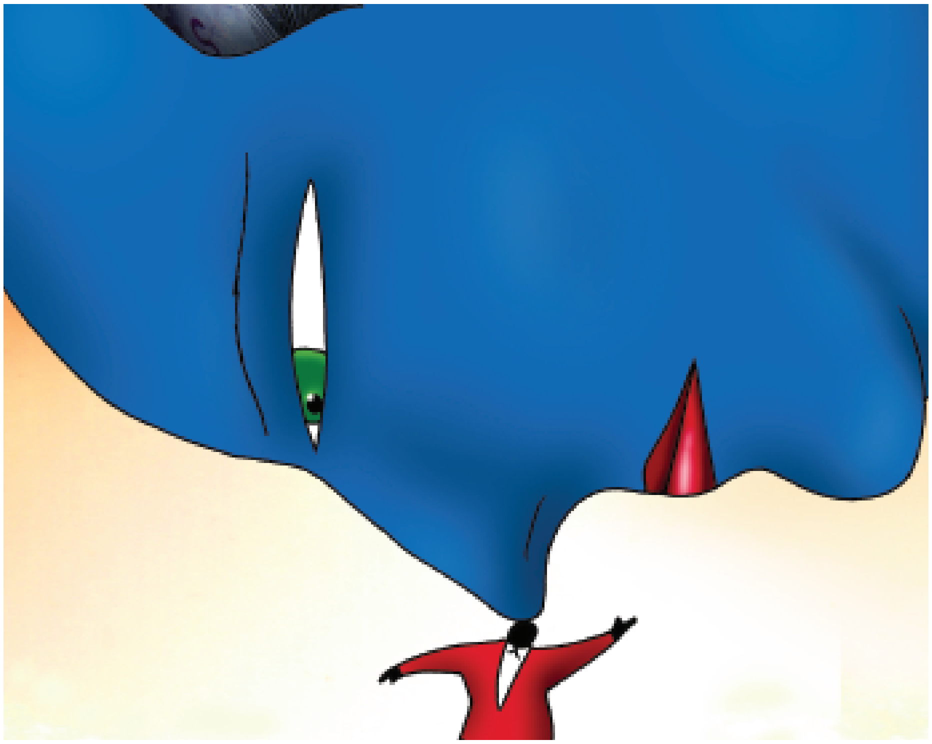
Often Web services toolkit providers, and sometimes platform providers themselves, will have completed specific integration work beyond the standards with other platform providers (e.g., Oracle, Microsoft, and IBM). This is to enable easier integration paths for Web services providers. A simple example of this approach is the widespread use of document-literal wrapped Web services, which is a Microsoft-specific approach to document-literal services modeling RPC calls that nearly all Web services toolkits support.

Beyond interoperability concerns, moving to an XML-based integration stack defined by Web services brings performance characteristics to mind.

Performance

In a Web services application, message sizes typically increase substantially from traditional binary protocols. Additionally, a new layer of marshaling, unmarshaling, parsing, and translating XML messages to and from the underlying protocols is introduced.

Therefore, an important part of any deployment architecture for Web services must include a comprehensive plan to understand the performance characteristics of the service end points and the clients using those service end points. Typically the performance needs to focus on two areas, throughput and latency.



Throughput

Throughput is the number of Web services requests, typically measured in bytes, handled in a given time period. Throughput is measured only on the server side and does not include the time in which it took to send or receive the message.

Latency

Latency is the round-trip time between sending a request and receiving a response. Latency is often subject to issues external to the server, such as network bandwidth and, in a heterogeneous environment, characteristics of the client environments.

The first question is, "What is the expected message size that will be passed through individual Web services end points?" Once the message size is determined, it is often a good practice to start with what might be termed a "null processing" test. The goal is to load up the deployment environment with concurrent requests with zero application processing on the server side to determine what overhead the Web services runtime itself puts on the environment. This allows you to ascertain the overhead of the Web services infrastructure independent of its interaction with the underlying systems.

Going through this exercise can reveal a number of issues within a testing and production environment, including the following.

1. ***Network.*** Often when testing the performance of Web services, network bandwidth can be the bottleneck. Network issues can impact both latency and throughput.
2. ***Client.*** Many vendors will optimize their Web services client to work best with their Web services runtime. However, using the Web services runtime-provided client could result in misleading measurements. Instead, it is a good practice to choose neutral third-party clients to generate load to avoid skewing results.
3. ***Server.*** Frequently, to achieve optimal performance on the server side, it is necessary to consult vendor documentation on how to have the server environment take advantage of the hardware resources available. Some of these settings can be vendor proprietary, and others are common to the runtime chosen. For example, configuration can significantly impact the throughput in J2EE environments that provide adequate memory allocation, parameters, garbage collection settings, and thread pool. Another common approach in J2EE environments (specific to each server) is running multiple Java virtual machines to more optimally take advantage of hardware resources.
4. ***Memory and CPU.*** Some client and runtime environments may be more sensitive to memory and CPU requirements—requiring more or less to generate or process Web services messages. If the client or server is bound by either of these constraints, accurate measurement of throughput may not be possible.
5. ***Message size and complexity.*** It is important to use representative message structures when testing Web services. Clearly the larger and more complex the message, the heavier the XML parsing requirement will be on the Web services runtime environment. Many Web services runtimes have different performance characteristics depending on message size and may have specific tuning capabilities that enable them to process messages differently based on the size of the messages.
6. ***Asynchronous services versus synchronous services.*** Most early Web services infrastructures focused on synchronous request/response implementations and one-way messaging. However, with the recent emergence of Business Process

Execution Language for Web services (BPEL4WS), many organizations are building infrastructures that contain a significant asynchronous component. In asynchronous services, one typically expects to see the ability to handle larger numbers of inbound requests, but mapping this number to a throughput measure can be skewed when comparing synchronous numbers because of the delayed nature of asynchrony.

These are some of the basic variables to keep in mind when considering basic performance testing of a Web services environment. However, sometimes the performance requirements overwhelm the ability of the Web services runtime to deal with SOAP messages. In these cases many architects will investigate messaging alternatives that are aligned with a service-oriented architectural approach.

One popular approach, available on Java platforms and aligned with Web services, is an Apache open source framework called Web Services Invocation Framework (WSIF). Apache WSIF enables developers to describe their underlying application interfaces using WSDL, and yet client invocations use native protocols rather than SOAP over HTTP. Classic examples of this include calling EJBs using native RMI protocols or vendor-specific optimizations such as using WSIF to natively call database-stored procedures.

In addition to interoperability and performance, Web services must be thought about from the classic reliability, availability, and scalability (RAS) characteristics needed in any large-scale deployment infrastructure.

Quality of Service

Web services typically take advantage of the same quality-of-service characteristics such as clustering, reliable messaging, and security available from server vendors for classical multitier applications.

Clustering

For scalability, developers are typically looking for a server environment that enables them to maintain consistent throughput and latency as concurrency of Web services clients varies. Scalable architectures enable the addition of more hardware, including machines, CPUs, and memory, as well as more resources both vertically (to a single machine) and horizontally (adding more machines to a cluster). Moreover, beyond the manual procedures for handling increased demand, modern server environments are self-adjusting, taking advantage of additional hardware resources on demand.

Remember, most Web services environments are either stateless or, if they are long-running such as business processes, their state is persisted in back-end databases. Both of these scenarios are supported by the classical cluster architectures available from application server vendors. For Web services running over the HTTP protocol, clustering solutions should span multiple tiers – from front-end caching and HTTP and J2EE servers to back-end databases.

> “A Web services provider may make a conscious decision to create a poorly interoperable implementation”

Reliable Messaging

Unique to the reliability of Web services is the infrastructure needed to guarantee delivery of a message to an end point. It can be relatively easy to get a message to a new service end point. However, when the back-end systems being exposed through Web services interfaces are not available, approaches using asynchronous technologies need to be evaluated.

A common approach for achieving reliable messaging is to receive SOAP messages over HTTP from external business partners for maximum interoperability and then move the SOAP messages over a reliable J2EE infrastructure backbone such as JMS. A simple protocol mediation layer, HTTP to JMS, can add a significant degree of reliability to message propagation within internal architectures.

More recently, with the arrival of reliable messaging standards that are protocol-independent, including WS-Reliability and WS-ReliableMessaging, organizations are looking at new reliability infrastructures. These are typically built into the Web services runtime infrastructures of platforms and ensure that messages arrive exactly once (often referred to as guaranteed message delivery).

The main issues with using the standards-based approach to reliable messaging are the relative immaturity of implementations interoperability concerns and, of course, the unavailability of such technology on older architectures. Although any serious implementation of reliability will gracefully degrade to work with nonreliability-enabled clients, architects who need reliability in their infrastructure often choose variations of the following strategies:

1. Work in a homogeneous environment in which both ends are reliable messaging–enabled from the same vendor.
2. Work with vendor implementations in which bilateral vendor interoperability testing has been done ahead of standards-based interoperability.
3. Offer different levels of reliable messaging. For nonreliable clients, process the messages but offer higher levels of reliable messaging with clients that meet the requirements of items 1 and 2.
4. Design a manual logging and log reconciliation of input and output messages.
5. Develop proprietary agreements between the client and server environments. Approaches here include schemes that rely on message exchange patterns or proprietary mechanisms within message bodies to determine whether messages really did make it to their end point.

Options 1–3 enable reliability to be tactically introduced based on standards. Options 4 and 5 offer solutions independent of standards and interoperability but may set up longer-term upgrade requirements as reliability infrastructures standardize.

Secure Messaging

As with reliable messaging, security of message exchanges has reached the early stages of maturity with the industry-endorsed release of WS-Security in April 2004. WS-Security defines standardized authentication tokens within messages, digital signatures for messages, and message-level encryption for Web services.

This very cleanly separates the security of Web services messaging from the transport protocol layer, providing much more flexibility from more commonly used HTTP protocol security such as SSL/TLS. Much like reliability, the biggest issues for WS-Security are the unevenness of implementations across vendors, interoperability concerns, and availability across older infrastructures.

Approaches for dealing with standards-based message security mirror what architects consider for reliable messaging.

If interoperability is not achievable across WS-Security implementations (e.g., via homogeneous clients and servers or bilateral vendor interoperability), architects will work to the lowest common denominator to achieve secure messaging. Two of the most common approaches are as follows:

1. ***Web-based security.*** Because most Web services run over HTTP, standard Web technologies such as SSL/TLS and basic/digest authentication work equally well. These approaches can be used for authentication, integrity, and encryption of messages "on the wire." Although not Web services–aware, these approaches tend to be supported on both old and new infrastructures, ensuring close to maximum interoperability.
2. ***Passing security tokens inside messages that can be used to verify authentication and message integrity.*** Rather than conforming to WS-Security standards, many organizations engaged in Web services transactions define an encrypted security token in a normal SOAP message for which a key or algorithm for generating or parsing such tokens is provided via an offline secure exchange. Public examples of this include Amazon's public Web services, for which a user key is required before use.

Ultimately, as is obvious from the variety of approaches, how developers tackle message-level quality of service depends on the sophistication of an organization's internal architecture as well as the capabilities of the expected Web services client environment.

Once these issues are addressed, there is a natural tendency to want to establish some sort of governance over those Web services.

Manageability

As a Web services application is deployed, classic management issues begin to appear. These issues can include monitoring and diagnostics, service-level agreements, policy management, centralized auditing and logging, and consolidating around a single identity management infrastructure. What is often done in this arena is to use parallel constructs to address these management concerns in traditional Web and multitier architectures. However, without careful adaptation, organizations can find that these constructs don't fit directly into Web services.

Take auditing and logging, for example. Unlike traditional Web traffic analysis, Web services logging and auditing is typically concerned with confirming how the individual messages or specific content in the messages correspond to what occurred in the back-end business systems. Correlating between these two often distinct tiers is much different than doing simple log analysis typical for Web content.

This simple example touches on the area of monitoring, diagnostics, and root-cause analysis that is critical for large-scale Web services infrastructures. The solutions in this area are mixed with some traditional management frameworks being extended to include this, new vendors emerging in the real time, event-driven business activity monitoring space, and traditional business intelligence tools being extended to be more aware of reporting against message stores.

Similar analysis must be done for service-level agreements (SLA) in the area of quality of service. Take, for example, WS-Security and WS-Reliability/WS-ReliableMessaging. Beyond simply implementing these standards, the longer-term vision is to enable SLAs to be exchanged in an automated fashion using emerging specifications such as WS-Policy. Such an exchange enables clients to programmatically and symmetrically match the quality-of-service capabilities supported by the server. Practically, however, most vendors provide different approaches for doing this today. The most common approach adopted by organizations currently requiring this today is simple, offline, noncomputerized agreements.

A common approach for normalizing the monitoring and diagnostics issues and enabling the centralization of control over Web services infrastructures is the concept of a gateway or intermediary through which all Web services traffic is routed. This central enforcement point provides both a consolidation and separation of management concerns from the back-end infrastructure. It also enables consistent application of quality of service policy as well as a convenient data capture point for analysis of Web services data flow.

The trade-off that architects often have to make with gateway approaches is the centralization of management versus the potential performance overhead of such an approach. Many gateway approaches deal with this performance concern by providing both an intermediary approach and an agent approach that works in conjunction with a centralized monitoring and diagnostics infrastructure.

Conclusion

This article focused on some of the issues that frequently confront architects when they attempt to deploy a large-scale interoperable Web services infrastructure. Although by no means meant to be a comprehensive enumeration of the issues and solutions, I analyzed some of the more widely known Web services concerns in interoperability, performance, quality of service, and management. Hopefully, you have gained an understanding of the key issues for their Web services deployment infrastructure.

About the Author

Mike Lehmann is a senior principal product manager with the Oracle Application Server 10g team at Oracle Corporation. In this role he is focused primarily on building out the Oracle Application Web services infrastructure.

■■■ mike.lehmann@oracle.com

BPEL in a Service-Oriented Architecture

BPEL's benefits for SOA

Service-oriented architectures (SOA) have gained much attention recently as a unifying technical architecture that can be concretely embodied with Web service technologies. SOA is a design model deeply rooted in the concept of encapsulating application logic within services that interact via a common communications framework. A key aspect of the Web service incarnation of SOA is that the Web service is viewed as a fundamental building block of an SOA-based application.

WRITTEN BY
JIM CLUNE

BPEL, originally called BPEL4WS (Business Process Execution Language for Web Services) is a language for describing Web service orchestration in terms of stateful, long-running interactions consisting of synchronous and asynchronous message exchanges. It supplies a notion of abstract processes to describe externally visible behavior as well as executable processes, which can be run either by some interpreter or by compiling them into some executable form. This article focuses on executable BPEL processes and presents one view on the benefits that BPEL brings to an XML Web service embodiment of SOA principles. BPEL plays an important role in SOA by providing a powerful means by which business logic can be articulated and executed at an abstraction level designed to provide the services needed for integration tasks.

Example: Travel Services

In order to illustrate what problems BPEL addresses and how it relates to SOA, we'll consider an example from the travel industry. Imagine a company offering travel services over the Web. The operations might include:

- ***getAvailableHotels***: Takes an input of an airport code and returns a list of hotels near that airport.
- ***getDescription***: Takes an input of a hotel identifier and returns a description.
- ***getRate***: Takes as input a hotel identifier, the type and number of rooms, and the date and provides a quote for the rate.
- ***makeReservations***: Takes as input the hotel identifiers, dates to reserve, and room information. Makes the reservation and returns the confirmation number.
- ***cancelReservation***: Takes a confirmation number and cancels the reservation.

When making a hotel reservation, all of these operations may be called in the process of making a reservation. The messages utilized by these services may be based on vertical industry definitions, such as those provided by the Open-Travel Alliance (OTA; www.opentravel. org). For example, the travel company may write its own WSDLs that rely heavily on the OTA schemas. For simplicity, we will assume that each operation is modeled as a separate service.

Let's consider what happens when we want to create a reusable service that represents some common scenario, such as shopping for a hotel. We will be calling several services, some of which will need to be invoked serially. For example, we need to get the list of hotels near a given airport in order to request detailed descriptions and rates for the hotels. Other services can be invoked in parallel, e.g., once we have the list of hotels, we may want to request descriptions and rates for all of them at once instead of one at a time. Some pieces may require user interaction, such as reviewing the list of options and deciding for which hotel to make the reservation.

We can envision this process as a flowchart consisting of basic activities and structural activities. Examples of basic activities include receiving messages, invoking external services, and assigning values from one message to another. Structural activities may include a sequence that performs nested activities in sequence, and a flow that performs nested activities in parallel. It turns out that this notion of activities is exactly what BPEL provides. In BPEL, the aforementioned activities are receive, invoke, assign, sequence, and flow. Fifteen such activities are defined in BPEL along with non-activity elements such as processes, partner links, variables, and correlations.

Listing 1 shows a brief snippet of how activities might be used in hotel shopping. Details such as data manipulation are omitted for clarity. The point to gain from this is that activities are represented as elements, and subactivities are represented as child elements. Therefore, the structure of XML reflects the structure of the process in an obvious way.

Structured collections of activities such as this can be used to define BPEL processes, which are in turn exposed as Web services; of course, this could also be done in a traditional programming language. So far, the primary benefit we have seen in BPEL is that the notion of interacting with Web services is built into the language. If our focus is almost exclusively on Web service orchestration, it is in some sense more natural to use a language that has these concepts built into it rather than a more general-purpose programming language. Nevertheless, this by itself is not very compelling because there are plenty of libraries that simplify the task of sending and receiving messages with protocols such as SOAP. We will expose more compelling reasons for preferring BPEL in this scenario as we go along.

Everything OLD Is NEW Again!

A WSDL-Centric View of Web Services

Implementing our travel services in a Web service–based SOA implies a decomposition into Web services. A naive implementation of this idea would force the use of SOAP to communicate between all reusable parts of the application, gaining modularity at the expense of efficiency. However, we need not make this trade-off if we are careful about the ontological commitments we make about the notion of a Web service.

The defining technical characteristic of a service from a BPEL standpoint is that it is described in a WSDL. Every message exchange is described in a BPEL process in terms of portTypes and operations defined in the WSDL. BPEL does not assume that services are accessed via SOAP over HTTP, so when appropriate, more efficient bindings may be used. For example, the Apache Web Service Invocation Framework (WSIF) defines WSDL extensions for bindings of local Java invocations, EJBs, JMS, and Java Connection Architecture connectors.

Binding information is not accessed directly in BPEL, but is controlled through deployment configurations. This means that our hotel shopping process can be written in a way that is binding-agnostic, cleanly separating the business logic from lower-level concerns.

Message Exchange Patterns and Stateful Services

Our hotel shopping scenario requires input from the user twice: once to initiate the request with criteria for the city or airport and the dates, and once more to review the offerings and either pick a hotel or cancel the request. Because this input happens twice, the whole service cannot directly be modeled as a simple request/response pair. We have a number of options for how to model this:

1. The process provides three request/response operations: getDescriptionsAndRates(), makeReservation(), and cancel().
2. The process exposes a one-way operation to initiate the hotel from shopping. The customer provides a request/response for picking the hotel from the list of choices or canceling.
3. Each message is modeled as a one-way operation.

Note that for options 1 and 3 the process receives multiple messages. This implies that our process must be a stateful service in the sense that it remembers the salient points from the beginning of the conversation instead of needing to start from scratch when it receives another message. Assuming that we wish to support multiple concurrent requests, we also will need some means of correlating which messages go to which conversation.

Conversation Identification and Message Correlation

One way to perform correlation is to insert explicit conversation identifiers into each message to identify the conversation in which that message is participating. Sometimes that works fine, and there are provisions for message identifiers and conversation identifiers in both JMS and in WS-Addressing. At other times, however, this is unnecessarily restrictive.

For example, someone sending a message might not know what conversation identifier to use, or even whether there is an existing conversation or if a new conversation must be created. BPEL handles these cases by introducing the concept of correlation and allowing processes to define correlations with respect to message contents. In the hotel shopping case, we may have an explicit identifier that is passed around, or we might key in on a collection of application-specific properties such as a tuple, consisting of customer name, date, and city. The flexibility to orchestrate services using both explicit and implicit conversation identifiers is a powerful tool in modeling extended conversations.

Long-Running Processes

Let's return to the options for message-exchange patterns to model our hotel shopping scenario. A key factor in making the decision about how to model this is the expected time involved for each stage of the process. If the user is at the browser and is waiting for a response after putting in the request, the synchronous messaging models may be fine. The rationale might be that maintaining HTTP connections is reasonable for the short duration that the user is waiting. However, a different usage scenario might have the user feedback happen via e-mail with a response time measured in days instead of seconds. In this case, the model that treats each message as a single one-way service seems most appropriate.

Once we open our process up to a longer-running life cycle, a number of interrelated issues comes to the forefront. One is synchronous versus asynchronous messaging. Asynchronous messaging is more appropriate for longer-running processes because it doesn't require connections to be maintained for unreasonable amounts of time or use unreasonable amounts of resources in maintaining connections. A traditional method of achieving this is through an enterprise messaging system, often accessed through some vendor-neutral API such as the Java Messaging Service (JMS). Enterprise-quality message queues from companies such as IBM and TIBCO provide reliable and persistent messaging and are often already in place in larger enterprises. However, the cost and complexity of these systems often serves as a barrier to entry, particularly for smaller enterprises. In recognition of this obstacle, some vendors such as Sonic Software are providing newer enterprise-messaging systems touted as being built from the ground up to support lower cost, as well as being easier to configure and deploy for Internet and firewall-friendly environments.

An alternative approach to asynchronous messaging is SOAP over HTTP with a callback specified through some addressing mechanism such as WS-Addressing. This is appealing from the standpoint of being lightweight and built on Web service standards, but it poses a reliability obstacle that is addressed by JMS and circumvented in synchronous request/response patterns. Standards such as WS-ReliableMessaging and WS-Reliability will address this obstacle, but keep in mind the infrastructure that implements these specifications is still required to achieve reliability. This infrastructure will often be available from the same vendors that are providing the enterprise messaging systems.

Another issue that lurks behind the notion of long-running processes is process persistence. The longer a given process is expected to run, the more critical it is that processes persist across power cycles. For a 500 millisecond transaction, a transaction that fails can simply be retried. When processes span weeks and months, persistence is essential. A major benefit of implementing a long-running process as a BPEL process is that the persistence of the process can be provided by the BPEL engine, greatly simplifying the business logic.

Endpoint Management

In our travel services scenario, one of the items most subject to change is the endpoint of the various services. A desirable feature of SOA is that these endpoints are dynamically and remotely manageable. Several technologies are relevant here. WSDL provides static endpoints, which are useful but limiting. A common practice is to separate the interface WSDL, which excludes the endpoint, and the implementation WSDL, which includes it. This allows for more dynamic endpoint resolution, such as that dynamically established via the ReplyTo field in WS-Addressing or in JMS. There are also cases in which multiple endpoints are valid, but some endpoints are preferable to others for efficiency reasons. For example, if multiple Java services are running in a single JVM, it will be more efficient for them to invoke each other directly in memory rather than via sending messages over socket connections. The key abstraction that BPEL offers in this is the notion of a partner link. Partner links may be specified at deployment time or at runtime. Because partner links are a first-class concept in BPEL and can be manipulated directly in processes, BPEL provides a full solution to endpoint management from the simple static deployment to the dynamic resolution that may depend on multiple factors, including technical considerations as well as business logic.

Transactions

There was one operation in our hotel service that seems innocuous, but hints at something more far-reaching: the cancel operation. The implication of this operation is that sometimes activities can be undone. Transactions are too big a topic to cover in detail here, but I will highlight a distinction between two types of transactions as they relate to BPEL and SOA.

The first type of transaction is an atomic transaction. Readers familiar with relational database management systems will be familiar with this type of transaction and the associated ACID semantics (Atomicity, Consistency, Isolation, Durability). Distributed systems that require ACID semantics normally achieve these properties by way of a two-phase commit protocol. The WS-AtomicTransaction specification defines one way of performing this type of transaction with Web service technologies.

In atomic transactions, data integrity comes at the cost of locking resources for the duration of the transaction. This can be fine for short-lived transactions, but for long-running business transactions, which may even run for months, resource locking is impractical. This motivates the need for another class of transactions that is more appropriate for long-running business activities.

This second class of transactions is sometimes called long-running transactions, or may be called a business activity. A defining characteristic is that instead of tying up resources with a locking-based protocol such as two-phase commit, the approach is to define compensation activities that can undo previous actions. Resources are not locked for the duration of the transaction, and state information can be shared across distributed resources even though the business activity is incomplete. The WS-BusinessActivity specification defines protocols along these lines based on Web service technologies.

WS-AtomicTransaction and WS-Business Activity together supercede the previously proposed WS-Transaction specification. These specifications are not yet widely accepted, and BPEL does not currently depend directly on either of them. However, the distinction between the types of transactions is an important one and BPEL does support this distinction. Some BPEL activities are explicitly specified in terms of ACID semantics (for example, assign activities are atomic, and scopes may be designated as serializable, which means they correspond to a serializable isolation level). BPEL also makes explicit provisions for the long-running transactional model through compensations and compensation handlers. Compensation handling is integrated with a fault-handling mechanism for consistent behavior in the presence of faults.

Benefits of BPEL

We have seen a number of properties common to many business process applications, including long-running processes; a need for reliable, asynchronous communication; heavy usage of Web service technologies; endpoint management; process persistence; and the ability to manage both atomic transactions as well as long-running business transactions. Although any of these properties can be achieved by writing code in a general-purpose programming language, the primary benefit of using BPEL is that it provides abstractions and infrastructure that are particularly suited to this class of applications.

This view of how BPEL derives its benefits can be contrasted with a portrayal that is sometimes found in the media, namely that the benefit of BPEL is that it provides such a high level of abstraction that business analysts can compose and run executable business processes by pointing and clicking in modeling environments. The view presented here is not that BPEL enables analysts to write software, but that the abstractions BPEL provides allow engineers to implement flexible solutions more effectively.

Summary

Service-oriented architectures are emerging as the design model of choice in integrating extended enterprises. BPEL plays an important role in SOA by providing powerful means by which business logic can be articulated and executed at an abstraction level designed to provide the services needed for integration tasks.

About the Author

Jim Clune is a development manager for Parasoft, where he has worked on a number of automated error prevention solutions. His experience with Web services includes the development of testing tools and his most recent project, which is a BPEL runtime engine. Jim holds a BS in engineering from Harvey Mudd College and is working towards a PhD in computer science at the University of California, Los Angeles.

■■■ jim@parasoft.com

Listing 1

```
<sequence>
    <receive operation="shop"
partnerLink="shopper"/>
    <invoke
operation="getAvailableHotels"
partnerLink="registry"/>
    <flow>
        <invoke operation="getRate"
partnerLink="rateProvider"/>
        <invoke operation="getRate"
partnerLink="rateProvider"/>
        <invoke operation="getRate"
partnerLink="rateProvider"/>
        <invoke operation="getRate"
partnerLink="rateProvider"/>
    </flow>
    <reply operation="shop"
partnerLink="shopper"/>
    <pick>
        <onMessage
operation="makeReservation"
partnerLink="shopper">
            <invoke
operation="makeReservation"
partnerLink="hotel"/>
            <reply
operation="makeReservation"
partnerLink="shopper"/>
        </onMessage>
        <onMessage operation="can-
cel" partnerLink="shopper">
            <empty/>
        </onMessage>
    </pick>
</sequence>
```

An Overview of the Java WSDP 1.5

A rich set of utilities for the developer

■ It can be difficult for developers, architects, and managers to keep up with new software packages and releases. This can be especially true with fast moving technologies like Web services. This article provides an overview of the main technologies that comprise the Java Web Services Developer Pack (Java WSDP). For more in-depth knowledge of the WSDP, simply download it and walk through the examples or complete the Java Web Services Tutorial.

WRITTEN BY
MICHAEL SICK

In an effort to standardize XML and Web services–related technologies, Sun Microsystems has developed implementations of popular standards and published them under the umbrella title of the WSDP. The toolkit's stated purpose is to simplify the development, testing, and deployment of secure and interoperable Web services. Version 1.5 is the latest release of the WSDP and contains many updates to existing technologies, new features, and a collection of bug fixes. This article will examine the main technologies provided in the WSDP and review their purpose and status.

Downloading/Installing the Java WSDP 1.5

The WSDP can be downloaded freely from Sun's WSDP site at http://java.sun.com/webservices/downloads/webservicespack.html.

On the server side, the WSDP runs within a container (WebSphere, JBoss, WebLogic, SunONE). Sun provides downloads for WSDP-ready containers from its site, including its own Sun Java System Application Server and Tomcat. Since the licensing is free and open, Tomcat is a good place to start. Sun offers a download of Tomcat for Java WSDP at http://java.sun.com/webservices/containers/tomcat_for_JWSDP_1_5.html.

Tomcat 5.0 for Java WSDP is built on Tomcat 5.0.19. The container supports JSP 2.0 and version 2.4 of the Servlet specification.

Related to the WSDP is the JDBC RowSet Implementations 1.0.1 JWSDP 1.5 Co-Bundle, which provides an implementation of a Web service–enabled rowset for remote access to databases. The Co-Bundle can be found at http://java.sun.com/products/jdbc/download.html#rowsetcobundle1_0.

Installing the WSDP is fairly straightforward once you have a container downloaded. One of the most notable weak points of the WSDP (in the opinion of this author) is that the installation is packaged as an executable (.exe) on Windows and is less transparent than most Java install scripts. A standard Ant-based installation would be helpful in porting the WSDP scripts to currently unsupported platforms.

WSDP 1.5 Technology

Below is a short description of the technologies that make up the WSDP 1.5. For each entry the purpose of the toolkit is summarized, the guiding JSRs listed, and the underlying XML specifications noted where applicable.

Sun Java Streaming XML Parser (SJSXP)

The WSDP includes the Sun Java Streaming XML Parser (SJSXP,) which is an implementation of JSR 173. The Streaming API for XML (StAX) found in JSR 173 provides a stream-based API for reading and writing XML documents. The StAX approach is also known as "pull" parsing because it is left to the programmer to pull the next event from the stream via an iterator-based approach. In cases where the current data/events will affect subsequent parsing steps, stream-based code may be easier to write than a typical SAX-based approach and still avoids the inefficiencies of the in-memory DOM-based approach. While pull parsing is not right for every parsing problem, the SJSXP implementation of StAX provides an additional and powerful tool for Java developers. SJSXP is nonvalidating and W3C XML 1.0 is Namespace 1.0 compliant. (*See* JSR 173*:* www.jcp.org/en/jsr/detail?id=173; *XML Pull Parser:* www.extreme.indiana.edu/xgws/xsoap/xpp/.)

XML Digital Signature Version 1.0 EA2

The XML Digital Signature package in the WSDP provides an implementation of JSR 105. JSR 105 provides a standard way for Java developers to access digital signature services and implements the W3C's XML-Signature Syntax and Processing specification. This specification defines the syntax and rules for creating and representing digital signatures associated with digital content (mainly XML). The specification provides support for enveloped and detached signatures and basic keying and management of digital information.

Using the XML Digital Signature APIs found in the WSDP, developers can sign and validate digital content and represent the signatures in an XML format. The signatures provide a standard way to verify that content originated from a known source and was not altered during transmission. The WSDP contains the core classes for using signatures as well as examples and documents. (*See* JSR 150*:* www.jcp.org/en/jsr/detail?id=105; XML-Signature Syntax and Processing: www.w3.org/TR/2002/REC-xmldsig-core-20020212/.)

JAXB Version 1.0.4 (Java Architecture for XML Binding)

The Java Architecture for XML Binding

WebAppCabaret™

J2EE Web Hosting

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

1.866.256.7973

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/Jetty
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...

Options

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

Resin: $10/mo

300MB Oracle: $40/mo

JRun: $20/mo

ColdFusion: $30/mo

$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...

$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

VPN

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

$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...

Prices, Plans, and Terms subject to change without notice. Please log on to our website for the latest Prices and Terms.

(JAXB) project is the implementation of JSR 33 that provides a standard API for automating the mapping between XML documents and Java objects. Using JAXB, a developer can compile a schema into Java classes that provide support for marshaling, unmarshaling, accessing, updating, and validating. The main goal of JAXB is to relieve the developer of these tasks so that they can be more productive and be exposed to fewer XML-oriented issues. JAXB 1.0.4 provides support for a subset of XML Schema and experimental support for RelaxNG. RelaxNG is a simplified XML schema system sponsored by the Oasis group. JAXB is currently integrated with JAX-RPC for easily transporting objects across the wire. JAXB 2.0 (governed under JSR 222) will focus on further integration with JAX-RPC, broader schema support, and underlying use of StAX for XML handling. (*See* JSR 31: http://jcp.org/en/jsr/detail? id=31; JSR 222: http://jcp.org/en/jsr/ detail?id=222; XML Schema Part 1: www.w3.org/TR/xmls chema-1/; XML Schema Part 2: www.w3.org/TR/xmlschema-2/; RelaxNG: www.oasis-open.org/committees/relax-ng/; UBL: www.oasisopen.org/committees/ tc home.php?wg _abbrev=ubl l.)

JAXP Version 1.2.6_01 (Java API for XML Processing)

The JAXP package (Java API for XML Processing) provides an implementation-neutral way of accessing XML-related tools like SAX2 and DOM Level 2 XML parsers, XSLT processors (via TrAX) and XML utility standards for XBase, XLink, XPath, and XPointer. Developers using JAXP can access implementations of these tools by their interface and keep their code from directly depending on a particular implementation. For example, XML parsers are pluggable and can be switched out to fix bugs, improve speed, or alter the memory profile without a code change or recompilation. The JAXP 1.2 specification is a maintenance release of JAXP 1.1 and adds support for W3C XML Schema. (*See* JAXP: http://java.sun.com/xml/jaxp; JSR 63: www.jcp.org/en/jsr/detail?id=63; JSR 5: www.jcp.org/en/jsr/detail?id=5; SAX2: www.saxproject.org/; DOM Level 2: www.w3.org/TR/DOM-Level-2-Core/.)

JAXR Version 1.0.7 (Java API for XML Registries)

The Java API for XML Registries (JAXR) provides a uniform facade for accessing XML-based registries and was developed under JSR 93. Registries are useful for organizing, relating, and adding metadata to services and resources.

The unifying approach of JAXR is useful because there are several similar and overlapping service registries in the Web services space. The JAXR API provides facilities for establishing connections to, querying, and updating registries. The current JAXR specification includes bindings for UDDI 2.0. Additional registries such as ebXML and eCo Framework can be developed and plugged in in a straightforward manner. (*See* UDDI: http://uddi.org/pubs/ProgrammersAPI-V2.04-Published-20020719.htm; JSR 93: http://jcp.org/en/jsr/detail?id=93; ebXML: www.oasis-open.org/committees/tc home.php?wg abbrev=regrep.)

JAX-RPC Version 1.1.2_01 (Java API for XML-based RPC)

The JAX-RPC project provides a uniform API for building Web services and Web service clients that use remote procedure calls (RPCs) with XML as a transport. JAX-RPC was originally developed under JSR 101 and is currently being developed under JSR 224 for JAX-RPC 2.0. A developer using JAX-RPC can either consume a remote Web service as a client or expose a Web service from the server side. The resulting code is RPC protocol independent. Web service based RPC protocols such as SOAP can be plugged into the JAX-RPC framework as needed. The 2.0 version of JAX–RPC is being increasingly integrated with the 2.0 version of the JAXB specification. (*See* JSR 101: www.jcp.org/ en/jsr/detail?id=101; JSR 224: www.jcp.org/ en/jsr/detail?id=224; SOAP: www.w3.org/TR/soap/.)

JDBC RowSet Implementations: Java Web Services Pack v1.5 Co-Bundle

The WSDP Co-Bundle provides a basic JDBC rowset implementation and several disconnected JDBC rowset implementations that can be used to access and update data in a relational database. Several of the rowset implementations provided are disconnected, that is they are not directly connected to the underlying database. The WebRowSet class is disconnected and can marshal and unmarshal itself to/from XML according to the WebRowSet schema. A developer can expose the WebRowSet via Web services and allow clients easy, powerful, and secure access to a remote database. (*See* Co-Bundle: http://java.sun.com/products/jdbc/download.html#rowsetcobundle 1 0.)

WSDP Recap

The WSDP provides a rich set of utilities for the XML and Web services developer. Developers can use the current implementation of the WSDP to build sophisticated Web service implementations in a standard fashion. As the WSDP expands over time to include additional security tools and updated XML-related standards, developers will continue to find it a valuable resource. For the latest information on the WSDP and Sun's Web service efforts, see http://java.sun.com/webservices.

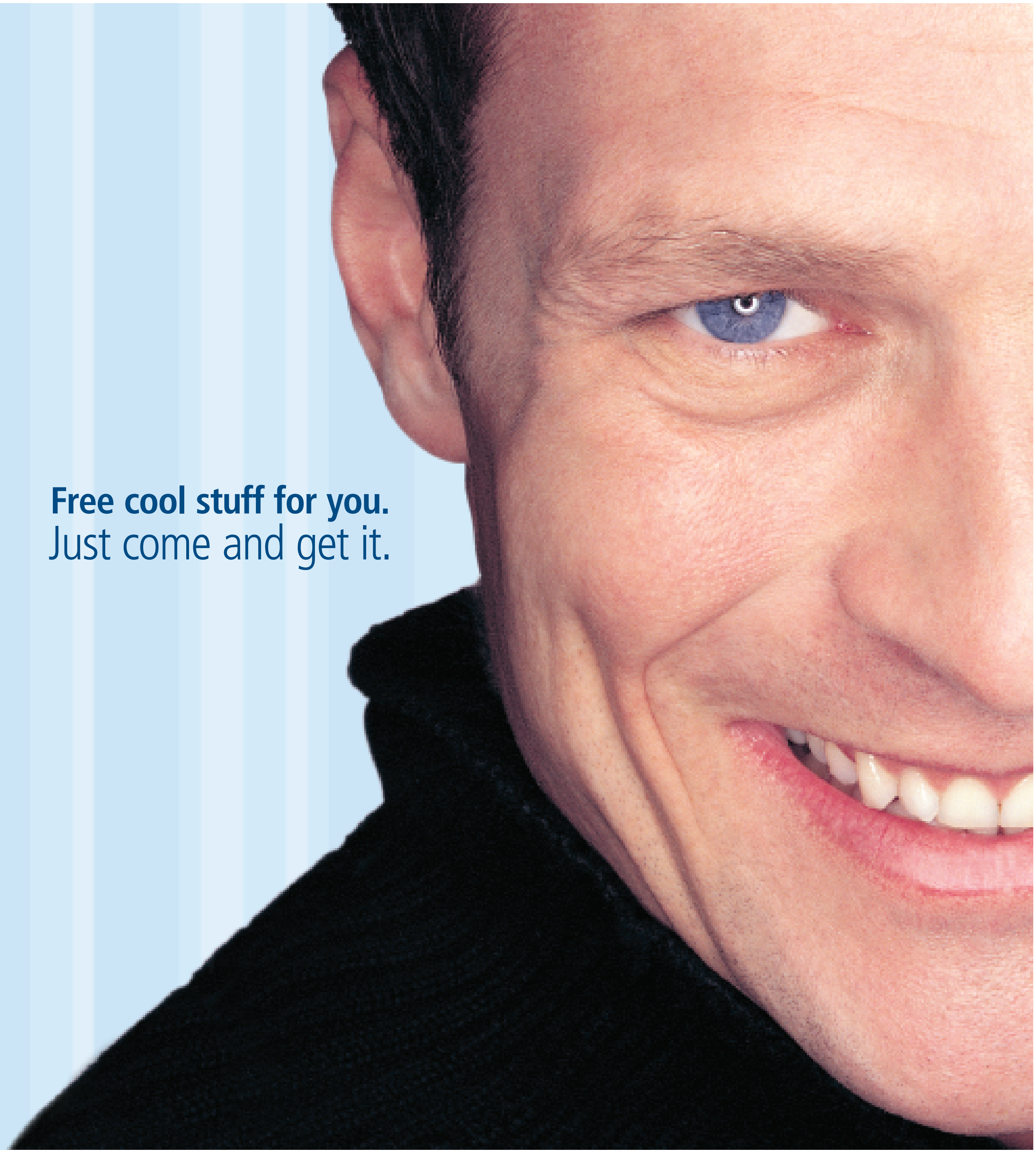
Additional Resources

- *WSDP:* http://java.sun.com/webservices/downloads/webservicespack.html
- *WSDP 1.5 Download:* http://java.sun.com/webservices/downloads/webservicespack.html
- *Java Web Services Tutorial:* http://java.sun.com/webservices/docs/1.5/tutorial/doc/index.html
- *StAX: Java's XML Pull Parser Specification:* www.sys-con.com/story/?storyid=45083&DE=1
- *JSR 173: Streaming API for XML:* www.jcp.org/en/jsr/detail?id=173
- *XML-Signature Syntax and Processing Specification:* www.w3.org/TR/xmldsig-core/
- *JAXP Site:* http://java.sun.com/xml/jaxp/
- *JAXR Site:* http://java.sun.com/xml/jaxr/
- *JAXB Site:* http://java.sun.com/xml/jaxb/
- *WS-I Basic Profile:* www.ws-i.org/Profiles/BasicProfile-1.1-2004-08-24.html

About the Author

Michael Sick is an independent Java architect helping clients solve complex product definition, design, and implementation problems. He has over nine years of experience in the construction of distributed information systems and internet technology. Mike has held many leadership positions including architect and VP of development, and has worked for major companies including Sun Microsystems, Cisco, and Blue Cross of Florida. He is a technical editor for *Web Services Journal* and a member of their editorial board.

■■■ mike@serenesoftware.com


Free cool stuff for you.
Just come and get it.

Consuming Web Services

Why service orientation is a move in the right direction

Until now, Web services have mainly been used to integrate application servers and databases behind corporate firewalls. However, powerful new services are becoming available over the public Internet as well. Companies such as salesforce.com, Grand Central Communications, eBay, Amazon, PayPal, and XMethods have deployed sophisticated service-oriented architectures (SOAs) that are sold either by subscription or as part of their existing product offering.

WRITTEN BY
BILL APPLETON

Salesforce.com has released sforce to provide integration and customization for their customer relationship management (CRM) service. Grand Central has an entire Business Services Network that sells integration as a service. Amazon's SOA focuses mainly on allowing customers to buy listed products, and the eBay service architecture is used by vendors who want to sell their products on the auction site. XMethods provides a directory of third-party services that performs a variety of useful tasks.

As these new services become available more attention will be focused on consuming Web services and combining them like software building blocks into composite applications. Developers and even end users can now create many new applications out of these existing services without the need to create their own SOA or maintain a server environment. This article reviews some popular methods of consuming services, and the technological advantages and disadvantages of the different strategies.

RSS and REST

Some Web browsers like FireFox and other software applications support RSS, which stands for Really Simple Syndication. The RSS format is an XML document that summarizes the content on a Web page, such as news articles or personal blogs. Other applications can use this information to view the content in a different way, for example, with a menu or sidebar full of bookmarks. These XML news feeds can be aggregated, allowing the quick combination of information from many different Web sites.

From a technical perspective, RSS is an example of a REST Web service implementation, which stands for Representational State Transfer. Every time you download an RSS document with an HTTP GET command you receive a summary of the latest news feeds available on the site. By adding parameters to the URL you can specify different types of news or information that are desired.

The REST interface can often be tested right in the browser by constructing the correct URL string and looking at the human-readable XML that is returned. Those little orange XML graphic boxes that you see on Web pages are actually RSS links that can be added to an RSS reader.

Another example of REST is the original Amazon Web services interface. In this case you could use URL parameters to specify a product search string and the downloaded XML document would contain information about the matching inventory items. Since REST commonly uses URL parameters for request arguments, true document exchange is limited, but REST is a simple and straightforward way to consume Web services.

Server-Side Solutions

Most Web services are described with an XML format called WSDL (Web Services Description Language). A WSDL document describes an orchestrated palette of related Simple Object Access Protocol (SOAP) requests and response envelopes. In the olden days SOAP requests looked like remote procedure calls, where you specified some arguments in the request and the service returned a simple answer or a data structure as a response. Nowadays most SOAP transactions look more like document exchanges: you are trading one complex XML document for another with a remote server of some type. The service transactions are often conducted by using HTTP POST running in a Secure Sockets layer (SSL) security environment.

The most popular method of providing WSDL and SOAP-based services is with an application server or other server software, including WebSphere, WebLogic, .NET, Apache Axis, and other J2EE technologies. Some of the powerful SOAs mentioned earlier are built on top of these application servers.

Server-side technologies are also a major method of consuming services (see Figure 1). For example, IBM's Portal Server integrates Web services data from various sources and generates HTML pages for browser display. Essentially, this strategy takes advantage of Web services as a form of data or application integration.

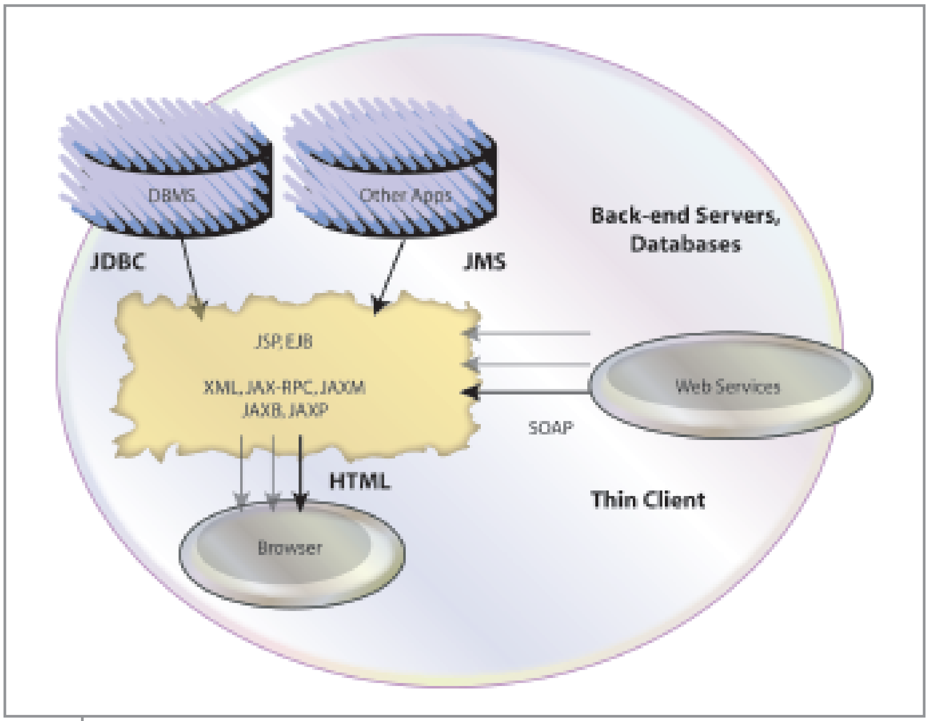


FIGURE 1 Server-side consumption of Web services

A drawback to the server-side consumption of services is that the user experience at the browser is usually limited to the HTML page-publishing model: click, get a page, click, get another page. This is fine for many purposes, but Web service applications often demand multiple data sources combined together in a dynamic fashion in a single, seamless interface. Another issue is that an application server is likely (on average) to make just 1 trip to the database of record for every 20 trips to the client. This native inefficiency is the result of trying to implement user experience and rich media on a server.

Hybrid Client and Server

A number of solutions has arisen in response to the bandwidth inefficiency and user experience problems posed by the traditional thin client model. One approach is the "user interface server" concept that connects a kind of enhanced client to a dedicated server that provides user interface data (see Figure 2). The client is often written in JavaScript, Java, Flash, or .NET, and the server is often some kind of Java, J2EE, or .NET software application.

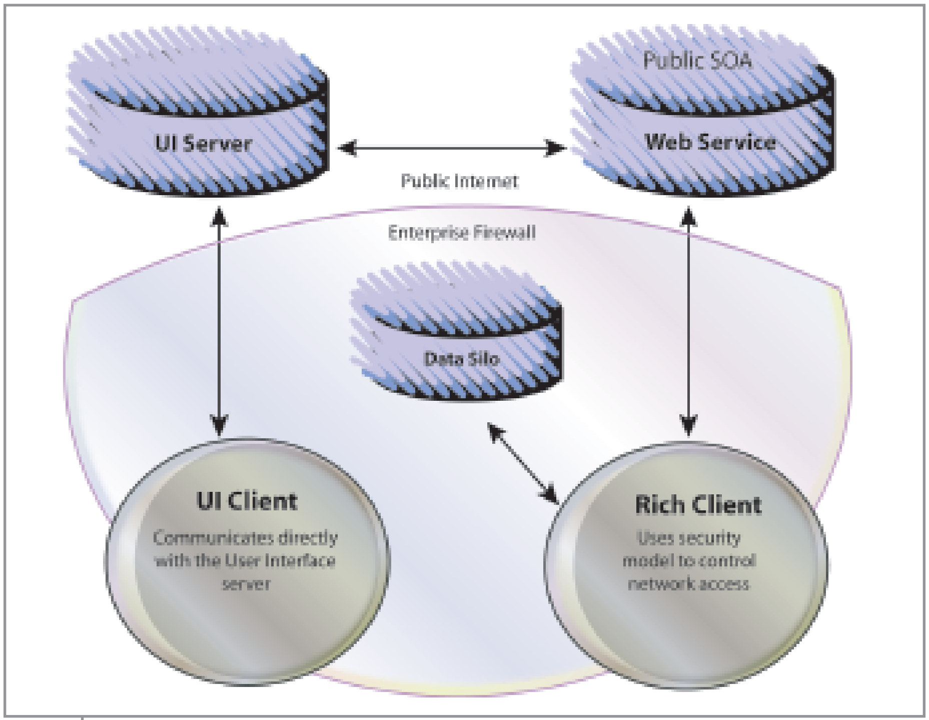

UI Server
Public SOA
Web Service
Public Internet
Enterprise Firewall
Data Silo
UI Client
Communicates directly with the User Interface server
Rich Client
Uses security model to control network access

FIGURE 1 Hybrid client and rich client consuming services

For example, Droplets uses a Java-based client connected to their server, which lowers bandwidth usage and delivers a better user experience. Laszlo Systems provides a server that consumes Web services and other enterprise data sources and generates Flash documents, which are sent to the client. Macromedia's new Flex server operates in a similar manner. The recently announced Avalon product from Microsoft has an XML format for a user interface called XAML that specifies documents that run in the .NET client.

The user interface server concept works nicely for client technologies that do not support or do not want to implement a security model. The client can conduct secure communications exclusively with the remote server, and the remote server can implement connections to other data sources. This allows the client to access remote data without the ability to directly connect to anything besides the dedicated server, which protects private data sources behind the firewall.

A potential disadvantage of the user interface server concept is that all Web service transactions must be conducted twice, once through the client and once again through the remote server. This results in two or three times longer round-trip delays for a given service call compared with a direct connection from the client.

On the open Internet, a single document exchange can usually be conducted in one-half second or less. There is a fixed cost to each Web service call, based on the number of transactions, and a variable cost, related to the amount of data being exchanged. Standards-based technologies like gzip compression can be used to attack the variable cost problem, but the fixed cost appears to be the limiting factor in calling remote services.

One answer here is to combine as much different data as possible into a single service call, or even to combine different services on the server side and send everything in a single package, perhaps with MIME or DIME encoding. A variety of new server products, such as Grand Central Communications' Process Designer, make this type of composite service delivery practical.

Composite Applications

Imagine that you could shatter your favorite HTML pages with a hammer and then combine all the pieces any way you like on a new page. This is the promise of composite applications, which assemble and connect various Web services from different sources together into entirely new applications. A major distinction between these applications is whether they run on the server, where the various services are combined, or on the client, which may also be capable of aggregating Web service information directly.

Essentially, all of the major portal vendors have some kind of composite application story. These systems use Web services or other data sources to compose portlets, Web Services for Remote Portals, or JavaBeans on a single page. This is similar to consuming services on an application server, discussed earlier. One limitation here is that few of these systems are really able to connect services into new applications; they are mainly focused on combining different portlets on the same page.

Composite applications can also be implemented on the client without the need for a dedicated server. These applications are capable of communicating with pure Web services directly at various endpoints and combining this information on the desktop. If the next-generation Internet is a network of intelligent platforms exchanging XML documents, shouldn't the same format be used for client communication?

The difficulty here is to support all of the various different protocols and data formats necessary to perform technical integration with other people's servers. For this reason some desktop clients are often written in Java or .NET to take advantage of the extensive libraries available to support integration activities. However these systems can be massive in size and difficult to install. Viewed differently, this is almost like setting up an application server on your desktop.

Security is another important issue. Client applications potentially have physical access to private data sources behind the firewall, and so these systems must have a strong security model to prevent access to unauthorized networks and yet enable access to authorized data sources. This is precisely why native browser technologies like DHTML and JavaScript only have network access to the originating host. If they could connect to other networks behind the firewall, this data could be pumped out of secure systems into other databases.

Another practical concern is that as more services are combined at the client level, there is an increasing likelihood that one or more of them will experience a service outage. This issue is somewhat easier to control on the server side, where an application server can manage secure connections and quality of service issues.

Examples of "rich client" applications capable of service aggregation on the desktop include DreamFactory, which uses a browser-based MIME type to deliver projects to the desktop; Above All Studio, which runs as a client in the .NET environment; and Digital Harbor, which has a composite application client written in Java.

An extremely useful aspect of rich client applications capable of true service aggregation on the desktop is in "on demand" situations where the customer has purchased a remotely hosted application as a service, but then needs to perform enterprise integration with a local data server or some other Web service. Installation of an additional server in this situation diffuses the original benefits of the on demand software model. But a rich client can connect to the service architecture of the hosted application and to another data source to present composite data views, or to transfer data between the two sources.

Conclusion

There is certainly a wide variety of ways to use Web services – and the diversity will continue to increase for the foreseeable future. For example, while we can easily connect to a service halfway around the world, there is currently no standards-based way to connect to another application on the same computer. This is ironic, because many of the interapplication communication technologies like COM, OpenDoc, and Frontier served as an early conceptual model for the development of XML-RPC, which led to SOAP and the modern Web service standards. Another giant destination for Web service consumption will simply be average software applications such as word processors and spreadsheets. Microsoft has made much progress in this area with its Office Suite.

The rapid development of new methods for consuming Web services is one of the main benefits of adopting a service orientation in the first place. Furthermore, because Web services are standards based you can be sure that the services you develop today can also be used in many different ways and on many different platforms in the future.

About the Author

Bill Appleton is president and chief scientist of Los Gatos, CA–based DreamFactory Software. He is a leading expert on service-oriented architectures, rich media authoring tools, and user interface development environments. Bill has written more than two dozen commercial software publications including the ground-breaking multimedia authoring tool SuperCard and the world wide number one best-selling CD-ROM Titanic: Adventure out of Time. As CTO of Messagebay, his ActiveX control work was downloaded over 75 million times.

■■■ billappleton@dreamfactory.com

eclipse CON™ 2005

2nd Annual Gathering of the Eclipse Community

Plug-in to Eclipse

This is your opportunity to get in-depth technical information from the Eclipse experts, learn the latest tips and techniques for using the tools, network with fellow enthusiasts and experience the breadth and depth of the Eclipse community. Attending EclipseCon™ will expand your knowledge and make Eclipse work better for you.

Tim O'Reilly

Urs Hoelzle

John Wiegand

Erich Gamma

Keynotes: Tim O'Reilly,
CEO of O'Reilly Media

Urs Hoelzle,
VP of Engineering, Google

John Wiegand and Erich Gamma,
Eclipse Platform PMC Leaders

Over 60 Sessions and Tutorials for beginner through advanced participants. Some of the conference sessions and tutorials include: Eclipse Platform · Rich Client Platform (RCP) · Test and Performance Tools · Debugging · J2EE Development · Web Development and Tools · Building Plug-ins · Modeling Framework (EMF) · C/C++ Development · Business Intelligence and Reporting · Embedded Development

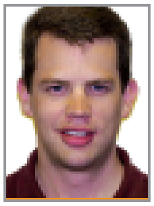
February 28-March 3, 2005
Burlingame, CA

Register at www.eclipsecon.org

Fuego Studio 5.1

A great means to undertake business process modeling

Organizations large and small routinely face challenges in managing business processes. From an administrative standpoint, policies and procedures are put in place to ensure compliance. However, no matter how strictly they are enforced, there are always those who work outside their boundaries.



WRITTEN BY
BRIAN BARBASH

The Fuego BPM solution helps organizations address compliance issues and business process modeling by providing a truly business-centric tool. It also provides the integration capabilities to participate effectively in a service-oriented architecture (SOA) through its native support for Web services, Java, CORBA, and XML technologies, among others.

Process Modeling and Development

Fuego Studio is the development environment for the Fuego BPM solution. Modeling business processes in this environment is very easy. Anyone who has worked with or created a swim-lane diagram in the past will be able to model processes in Fuego Studio. As shown in Figure 1, a very simple business process has been modeled for handling the application process for prospective candidates in a company. Activities are organized into swim-lanes identifying the owner role of each activity. The model illustrated in Figure 1 contains two roles: Candidate and Hiring Manager. Swim-lanes that have no title contain activities processed by the system only. For the purposes of clarity, Fuego Studio provides annotation capabilities that may be used to describe unlabeled swim-lanes.

For the purposes of this article, the business process modeled in Figure 1 is as follows:

1. The job candidate fills out a basic form using the provided Portal Web application requesting their name, e-mail address, postal address, and a text version of their resume.
2. The address of the candidate is validated against an external Web service. If the address is not valid, candidates are notified that they must update their information. If the address is valid, the candidate's data is passed to the Hiring Manager for review.
3. The Hiring Manager reviews the candidate's information using the provided Portal Web application. If the Hiring Manager does not review the candidate's application within one day of receiving the application, a reminder is sent to the Hiring Manager. Once complete, a notification of the Hiring Manager's decision is sent to the candidate via e-mail.
4. If the Hiring Manager decides to hire the candidate, the Process Paperwork action is executed to bring the candidate on board and the process ends. If the Hiring Manager decides not to hire the candidate, only a notification is sent and the process ends.

All orange icons represent activities that require user intervention. All blue icons represent activities that are executed by the Fuego system automatically.

For the business process modeled in Fuego, the next step is to define and develop the supporting code for execution. By allowing process modeling to take place independently of code development, Fuego allows different constituencies to focus on the components of the model that are relevant to their expertise.

Defining and developing the supporting code for process execution involves the following basic steps:

- Establishing the system variables to contain process instance data
- Cataloging external resources and APIs to integration touch points
- Building user interfaces for collecting input from participants
- Writing the necessary code to manage the process

For the purposes of this article, the focus of development will be on cataloging and executing external Web services.

Web Services Integration

Integrating Web services into a business process model in Fuego Studio is very straight forward. As mentioned previously, the candidate's address will be validated against an external Web service. To make this service available, it's added to the catalog (a collection of tools, libraries, and services available to Fuego at design time and runtime).

Adding a Web service to the catalog requires the developer to import a WSDL document using a wizard. The developer identifies the location of the WSDL document and specifies a module name that will contain all services identified in the WSDL. The result is a set of the following objects.

- ***Fuego Component:*** Contains the Service

FUEGO

Company Info

Fuego
2400 Dallas Pkwy
Suite 350
Plano, Texas 75093
Phone: 1 972-801-4200
Fax: 1 972-801-4201
Sales, U.S. & Canada: sales@fuego.com

Names defined in the WSDL document and all supporting data types

- *Fuego External Resource:* Definitions that identify the host of the Web service
- *Fuego External Resource:* Definitions that identify the relative paths on the service host to all services contained within the WSDL document

With the Web service cataloged, invoking it involves writing some basic code into the appropriate Activity. Listing 1 shows the code required to execute the AddressVerificationService within the Check Address Activity.

Logic in Fuego is written in a simple-to-understand language that hides the details of the service invocation. XML documents participating in the Web service call are structured in an object hierarchy, thus making the contents of each document available using standard dot notation. In Listing 1, all variables prefixed with "i_" represent instance variables available to all activities through the life of a given process instance. The variable prefixed with "l_" is a locally scoped variable available only within the Check Address Activity.

The code in Listing 1 is just an example of the capabilities of the processing available in an activity. Additional sophisticated options exist for handling the required logic, routing processes, and interacting with external systems.

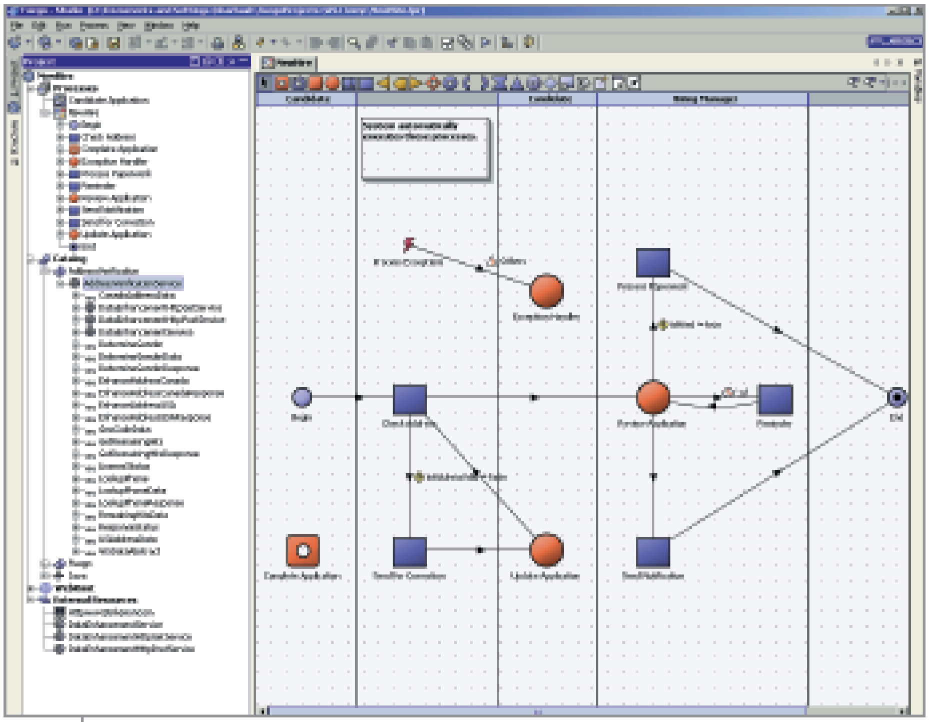


FIGURE 1 Fuego Studio IDE

> "By using the swim-lane diagram as its design interface, it allows the key stakeholders with the process knowledge to work with the tool without requiring a technical skill-set"

Summary

Fuego's BPM solution and the Fuego Studio provide a business-process modeling environment that focuses on business first. By using the swim-lane diagram as its design interface, it allows the key stakeholders with the process knowledge to work with the tool without requiring a technical skill set. Developers may then focus on creating the supporting code and integration touch points to support the true business need. And with its strong integration capabilities, Fuego fits well into an SOA. Overall, Fugeo's BPM solution is a good approach for tackling business-process modeling.

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

Listing 1. Web services invocation

```
// Execute the Web Service
checkddress AddressVerificationService
    using address1 = i_address1,
          address2 = i_address2,
          city = i_city
          state = i_state
          zip = i_zip,
    returning checkAddressResult = l_checkAddressResult

// Check the response
if l_checkAddressResult.responseCode = 0 then
    i_isAddressValid = true
else
    i_isAddressValid = false
end
```

Metadata Evolution Management in Your SOA

Managing the life cycle of your metadata

This article describes the increasing importance of metadata in today's service-oriented application landscape, and the consequent fragility inherent in architectures when faced with change.

WRITTEN BY
JIM GABRIEL

When we reach the point at which metadata drives the development and maintenance of services, evolving business requirements force us to break open and evolve our metadata first, and then to address the services dependent on that metadata. Most development methodologies and environments are not sufficiently equipped to deal with such metadata-driven change. This article recommends a shift in the way we manage evolution in metadata-driven applications.

SOA and Metadata

A service-oriented architecture (SOA) is a metadata-driven architecture. Metadata is crucial to the development life cycle of Web services because the long term maintainability of the SOA is at risk when the business logic expressed in services is not visible to the IT department at a higher level than in the code itself. However, there are many different kinds of metadata, not all of which are visible in application development environments. Figure 1 illustrates the metadata that we care about in a SOA.

The top half of this diamond represents WSDL and "policy" metadata, which is what most developers think of when we talk about metadata in an SOA. This metadata is described in XML, hence the general understanding that services are XML based. WSDL and policy metadata are low in semantic business information and high in technical information – they provide or facilitate the plumbing that allows the services to function.

The WSDL and policy part of this equation is of low strategic value to the business because it is largely generated. It falls out of any one of a number of application development tools that might be used to design and create services, or is handcrafted according to relatively simple requirements. When change is necessary in the business logic of an SOA, developers seldom need to concern themselves with this XML – it is the visible, accessible part of the iceberg, as it were.

The lower half of this diamond describes the payload, or the messages, that the services must process in order for the business process to succeed. Payloads require a very different and altogether more fragile kind of metadata: XML schemas. Strictly speaking, services with document-centric payloads can operate very well without an external description in XML schema – that is, all payloads have an implicit schema, and there is no requirement to express the schema explicitly in XSD. Without comprehensive metadata describing the payloads, however, implementing changes to a business process quickly resembles the process of looking for a needle in a haystack.

Expose the Underlying Models

To repeat an earlier statement: the long-term maintainability of the SOA is at risk when the business logic expressed in services is not visible to the IT department at a higher level than in the code itself. This is an important mantra when you consider that only message-based, document-centric SOAs are likely to be successful and low cost in the long term as these allow us to rise above the point-to-point, RPC-style application integration of early service-based architectures. The latter tend over time toward time-consuming, error-prone, application-specific, and high-risk maintenance phases.

If you accept that the SOA should be message based, and your long-term goal is to achieve optimum efficiency in the development life cycle, a best practice is to externalize the schemas, expose the models, standardize, and federate. (This, by the way, is the advice currently being propagated by the majority of the world's SOA authorities, such as IBM, Sun Microsystems, Gartner, and so on.) Beyond a certain level of complexity, especially with multiple developers and teams collaborating on the development of services, the only safe way to constrain the business processes of an organization is to make the data model explicitly visible to all architects, developers, and project managers as a coherent set of XML schemas, and then to drive all service development on the basis of those schemas.

This article does not discuss how best to externalize schemas, as this will be the subject of an article in next month's issue. Suffice it to say that rather than attempting to extract or derive a schema from a service, our starting point ought to be the integration of the underlying data models, followed by the development of services. The important assumption for this article is that the messages carried by services are described and constrained by the integrated data model, and expressed in XML Schema. This metadata is of very high semantic and strategic value because it describes the business processes of an organization, as opposed to the plumbing.

Advantages

The advantages of externalized schemas for message-based SOA are as follows:

- Enforceable contracts for processing behavior
- Visible specifications for developers
- Public interfaces for new partners in the SOA
- Schema-based access to standard infrastructure such as parsers, transformation engines, and so on
- Insulation for services from changes to schemas
- Support for business analysts when planning changes

Disadvantages

The disadvantages of any metadata-driven application environment are due entirely to the limitations of metadata in general and XML schemas in particular. What are these limitations? In essence, the XML schemas describing payloads are application specific, bespoke metadata that is subject to change, and requires human involvement when it evolves.

Nowadays we must expect schemas to change. Unfortunately, schema families and their associated assets (transformations and so on) present us with horrific redundancy and duplication when we try to evolve them by editing them. In any orchestrated set of Web services used and maintained by multiple development teams – for example, an order-to-invoice trading transaction involving multiple players – the externalized schemas and transformations (probably one of each per service) describe or reference the same data objects over and over again. Modifying any object presents the kind of maintenance nightmare that most of us try very hard to avoid in conventional programming environments.

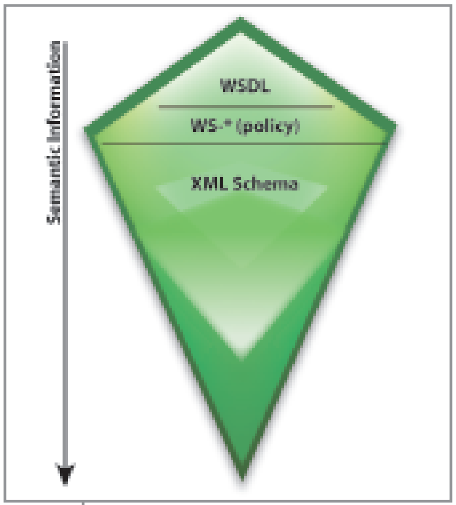


FIGURE 1 Metadata in an SOA

Versioning and Impact Problems

Managing XML infrastructure is different. When developers modify schema-driven applications by modifying the schemas, two problems arise: first, the new versions of the schemas are no longer in sync with the older versions; and second, the lack of a robust, scientific mechanism for identifying where every object has been defined and referenced forces developers into a manual maintenance exercise. This is generally not a problem when there is only one schema and one developer. For multiple schemas and multiple developers (or worse, multiple teams of developers), you have a very serious risk of conflicting modifications and inconsistency.

The sheer proliferation of references to single objects, coupled with the number of places where objects can be reused, increases the workload of maintenance projects in an exponential curve. Typically, the only people who can carry out maintenance work beyond a certain level of complexity are highly paid system experts who become IT bottlenecks due to the level of manual work involved. Such work can be very tedious.

Risk

The high level of risk inherent in such a situation makes the bottom half of the diamond in Figure 1 equivalent to the hidden 9/10 of the iceberg, which as you will recall from the Titanic is the stuff that sinks supposedly unsinkable ships. It is a sobering thought that if you cannot manage the evolution of the schemas governing the payloads in your SOA project, you may not even have a project in the long term.

Versioning and Extensibility Issues

The first question that most organizations set out to answer at this stage is, "How do we version schemas?" There are many different techniques applicable to the schema versioning problem, ranging from forcing instant incompatibility at one end of the spectrum (and thus forcing systems to upgrade), through to the opposite end of the spectrum where schema constraints are relaxed sufficiently to allow steadily broader ranges of content in service payloads.

These versioning techniques provide enough material for an article in their own right, so we will avoid the temptation to go into too much detail here. A quick summary is that while it is not impossible to version schemas and the systems that depend on them, nothing comes for free, and there are very few robust mechanisms and procedures that work well.

Extensible Schemas

Having experienced the pain of schema versioning, the next question that organizations inevitably ask themselves is this: "Is there a way of designing our schemas from the outset so that they are extensible?" Architects and system analysts are delighted to discover that XML Schema provides various ways of designing in extensibility, thus ensuring that schemas can be modified (read: "extended") without affecting existing systems.

Each extension, however, has the disadvantage of making the schema considerably more complex, with the logical conclusion in the long term that your metadata reaches a level of complexity that is unmanageable. Again, the subject of schema extensibility is large enough to warrant a separate article. For those who are interested, David Orchard has written an excellent article at www.xml.com/lpt/a/2003/12/03/versioning.html.

Impact in Deployed Environments

The consequence of any schema change forms the basis of the next question: "Okay, we've changed our schemas, how do we realign deployed services without any downtime?" Managing the change in metadata is one thing, managing the related changes in application code can be another problem altogether. What are the dependencies, and how can we automate the change management process? Again, in a contained environment with few developers,

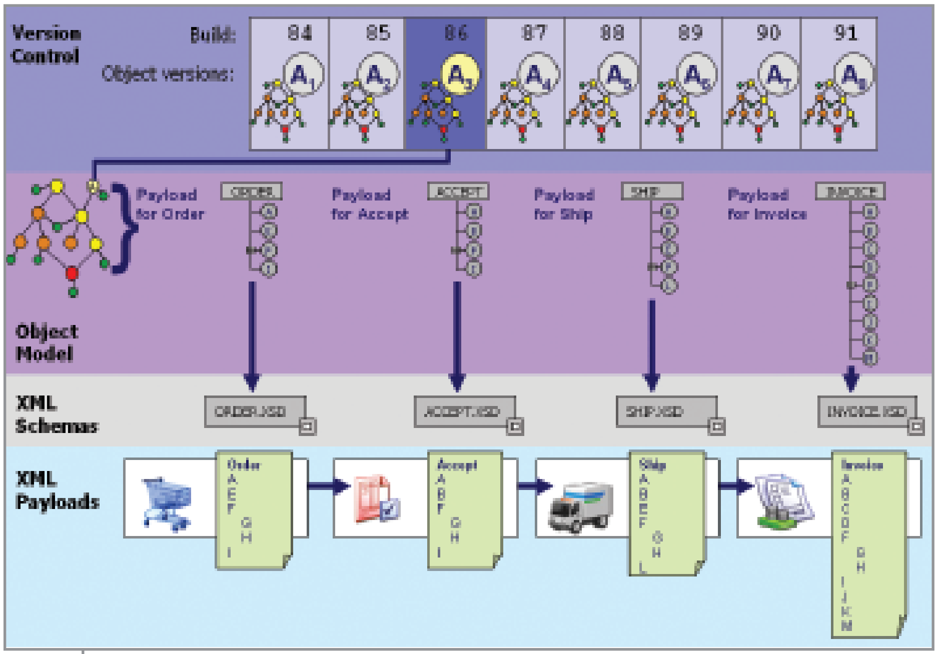


FIGURE 2 | **Managing XML metadata evolution through technology**

these issues do not present many problems (at least, nothing that cannot be overseen).

Pain Point for Life-cycle Management

However, the crucial question for life-cycle management is, "At what point does the evolution of your metadata become a challenge that adversely affects the life cycle of your Web services development?" Managing the evolution effectively is critical when the following apply:

- The organization has invested heavily in XML metadata-driven systems
- There are multiple developers and multiple teams working with the metadata
- There is ongoing change
- There is ongoing internal or external integration

Managing the life cycle of your Web services development, particularly from the perspective of the evolution of metadata, is not a schema-versioning problem. Versioning schemas is about technical constructs and development processes, not about the management of metadata evolution. Metadata evolution management is the real problem facing the long term lifecycle management of Web services development projects.

Metadata evolution management is not scalable with most current technology. The current technology treats metadata as passive, as a reflection of what exists in the applications landscape. XML, and SOA in particular, are altering the role of metadata, however. Metadata is now active. We build new inter-enterprise and inter-application systems by first agreeing the contracts (the schemas or metadata), and then writing the code. The schemas determine how we program. It logically follows, therefore, that changes to such systems must first be implemented in the metadata and subsequently in the code.

New Mix of Technologies

To support active metadata, we need a new mix of technologies. An enterprise data dictionary platform is necessary to make all service-related metadata centrally visible to developers, wrapped in a development environment that conforms to a model-driven architecture. Such a system allows changes to metadata to be powerfully implemented in one central place (within the model contained in the dictionary) and deployed out to the system via automated processes and generators, as one would expect of a model-driven development environment. The visible metadata for the community of consumers appears as a strongly version-aware and variation-aware enterprise metadata registry.

The following functionality and technology are necessary to support this concept:

- Facilities for loading and managing existing metadata as an integrated data model
- A means to remove all duplication and redundancy in the integrated data model
- Design and development tools for ongoing metadata development or modification
- Impact analysis tools
- Change management
- Fine-grained version control
- Model-driven architecture
- Central repository
- Collaborative development across multiple teams
- Release management

In essence, this list represents a way of elevating ourselves above the level of schema-versioning issues, and ridding ourselves of all redundancy and duplication (see Figure 2).

Loading Existing Metadata

To create a model-driven environment for the development, management, and deployment of enterprise metadata, we first need to create an object model. This can be done either manually or by "vacuuming up" all existing schemas, XML representations of application models, and so on.

Remove Duplication and Redundancy

The essence of a true object model of metadata is that it is single source. This means that objects exist only where they have been defined, and all objects are unique. Every possible reference to, or reuse of, an object is managed as a reference link. During an import process, all redundancy and duplication is removed through the enforcement of unique object names in namespaces. Objects that cannot be resolved at import time can be managed as such, for future attention.

A distinction is made here between the objects that constitute schemas, and the schemas themselves. A schema is merely an assembled structure that pulls together objects from an object model and applies certain deployment properties.

Accordingly, importing a schema results in a record of how the schema was assembled, and–separately–a model of the objects in that schema.

Design and Development Tools

Having created a single-source object model, it is imperative that development happens in the model and not in schemas. This part of the process represents the biggest break with the most commonly applied working methods – modifying a schema should never mean editing a schema file when you are working at this level of sophistication or complexity. Rather, editors must be available to allow object-level edits within the context

*Offer subject to change without notice

Learn Web Services. Get a New Job!

Only $69^{99}

1 year (12 issues)*

* Newsstand price $83.88 for 1 year

Subscribe ONLINE
www.wsj2.com or
CALL
888 303-5252

*Offer subject to change without notice

Get Up to Speed with the Fourth Wave in Software Development

The Best
.NET
Coverage
Guaranteed!

Subscribe today to the world's leading Web Services resource

- **Real-World Web Services**: XML's Killer App*!*
- How to Use **SOAP** in the Enterprise
- Demystifying **ebXML** for success
- **Authentication**, **Authorization**, and **Auditing**
- **BPM** - Business Process Management
- Latest Information on **Evolving Standards**
- Vital technology **insights** from the nation's leading Technologists
- Industry **Case Studies** and **Success Stories**
- Making the Most of **.NET**
- **Web Services Security**
- How to Develop and Market Your Web Services
- **EAI** and Application Integration Tips
- **The Marketplace**: Tools, Engines, and Servers
- Integrating **XML** in a Web Services Environment
- **Wireless**: Enable Your **WAP** Projects and Build **Wireless Applications** with Web Services*!*
- Real-World **UDDI**
- **Swing**-Compliant Web Services
- *and much, much more!*

SYS-CON MEDIA

WebServices JOURNAL
.NET J2EE XML

SYS-CON Media, the world's leading *i*-technology publisher of developer magazines and journals, brings you the most comprehensive coverage of Web services.

of the object model New schemas are "assembled" from collections of objects in the pool of objects that the model represents. Existing schemas can easily be redeployed without further modification because they are simply descriptions of the objects they contain and not the objects themselves.

Impact Analysis

Whenever an object is modified in an editor, the system is aware of its place in the object model and all instances of its use in assembled schemas. It is therefore possible to generate impact analysis reports that chart exactly what is impacted by any given change.

Change Management

If it is possible to analyze the impact of change, it is possible to automate all or part of the implementation of that change. For pure metadata implementations (i.e., schemas, transformations, etc.), this means creating a clean, identifiable, version-controlled build of the object model and generating out the deployable objects to the required consumers. Such an action should be automated and centrally driven by a build manager or administrator.

WSJ ADVERTISER INDEX

ADVERTISER	URL	PHONE	PAGE
Active-Endpoints	www.active-endpoints.com		17
EclipseCon 2005	www.eclipsecon.org		27
Ektron	www.ektron.com/ws		47
EV1Servers	www.ev1servers.net	800-504-SURF	6
Global Knowledge	www.globalknowledge.com/wsj		23
HostMySite.com	www.hostmysite.com/mxdj	877-248-4678	Cover III
Itemfield	www.itemfield.com/itsolutions	800-838-1055	Cover IV
LinuxWorld Conference & Expo	www.linuxworldexpo.com		57
Mindreef	www.mindreef.com/tryout		9 & 11
OpenLink Software	http://virtuoso.openlinksw.com	800-495-6322	Cover II
Parasoft	www.parasoft.com/soaptest_WSJ	888-305-0041	5
Promind Systems	www.xslmaker.com		3
Secrets of the XML Masters	www.sys-con.com/freecd	888-303-5282	53
SYS-CON Publications	www.sys-con.com/2001/sub.cfm	888-303-5282	49
Web Services Edge	www.sys-con.com/edge	201-802-3066	35 - 40
Web Services Journal	www.wsj2.com	888-303-5252	33
WebAppCabaret	www.webappcabaret.com/wsj.jsp	866-256-7973	21

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

Fine-grained Version Control

Object-level edits to metadata make it possible to manage edits to schemas at the object level. Storing an incremented version of an object in such an environment is easy. The great fallacy of version control in schemas, however, is that anyone would want to access single objects according to their version level.

Object-level versioning is important in the concept of schemas because schemas are based on a complete, coherent version of an object model, and we often want to introduce a single change to a single object in a whole family of schemas. Such a change must result in a re-release of the schemas at a new version level. The editing work is possible at the object level, however, and it must therefore be possible to cycle back to pre-edit versions of the model and branch off, or bug-fix the earlier version and merge again with the later version. This is only possible if object-level versioning is supported.

Model-Driven Architecture

The model-driven architecture is essential for supporting the Construction► Assembly ►Deployment workflow and paradigm. Only in this way can we move away from modifying an object by physically locating and manually modifying all references to, and instances of reuse of, that object. In a model-driven architecture, modifying the single object in the model is usually sufficient to equip the system with all the information it requires to regenerate all schemas, transformations, etc. where the object is referenced.

Central Repository

A repository mechanism using transaction-aware database technology is necessary to enable all of the above.

Collaborative Development

The following are essential to collaborative development environments:

- Access control
- User administration system with users, groups, roles, and permissions
- Single-user project workspaces on client machines
- Check-in and check-out in the central repository
- Conflict resolution at check-in and check-out time, managed by integrating checked-in tasks in a build mechanism

Conflict resolution solves the conflicts caused by parallel edits on the same object by different developers.

Release Management

A build mechanism enables us to uniquely identify and version-control an entire object model, including all the schemas, transformations etc., that have been assembled from that model. Thus, a release that is based on a particular build can identify a version-controlled set of all the deployable objects for the model.

Summary

Doing all of the above correctly gives us the perfect solution to the life-cycle management of metadata-driven Web service development. In a nutshell, when metadata drives evolution, we manage the metadata evolution process by applying a model-driven methodology for exposing, standardizing, and federating the metadata. Such an approach makes it possible for all consumers to access metadata cleanly, simply, and safely through an enterprise metadata registry.

About the Author

Jim Gabriel is an architect, author, entrepreneur, and inventor. He got to where he is today thanks to a disproportionate number of metadata evolution management problems. Jim designed and engineered the production of CortexML for XML metadata evolution management. He works with London-based digitalML Ltd, where he is responsible for the CortexML division of the company.

■■■ jgabriel@digitalml.com

See what everyone is talking about… ***The Application Server Shoot-Out!***

International Web Services Conference & Expo

Keynotes

Matt Ackley

eBay

Ari Bixhorn

Microsoft

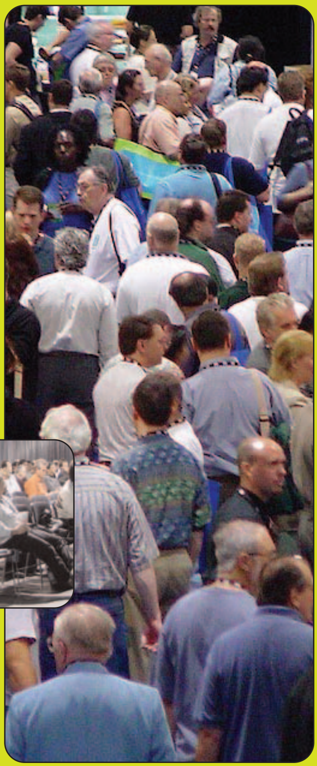
Mike Milinkovich

Eclipse.org

web services conference & expo EDGE

Register by JANUARY 21, 2005 SAVE Up To $300

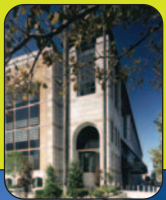
WEB SERVICES EDGE 2005 EAST



Product Teams Delivering SOA, Web Services, Java, .NET, & XML Solutions

New for 2005…

- Cabana Night Developer Exchange
- Application Server Shootout
  facilitated by Burton Group
- Hands On VS and .NET Labs
- Software Giveaway



Free Tutorials presented by critical sites Novell. Microsoft Java Community Process

For more information:
visit: www.sys-con.com/edge
call: 201-802-3066
e-mail: jimh@sys-con.com

February 15-17, 2005
John B. Hynes Convention Center
Boston, MA

PRODUCED BY: SYS-CON MEDIA

PRESENTED BY: SYS-CON EVENTS

DIAMOND SPONSOR: Microsoft

GOLD SPONSORS: Novell. SAP

ASSOCIATION SPONSORS: INTEGRATION CONSORTIUM OASIS ineta eclipse

CORPORATE SPONSOR: Java Community Process

APP.-SERVER SHOOTOUT FACILITATOR: burton GROUP

MEDIA SPONSORS: information STORAGE + SECURITY journal, JDJ, WebServices JOURNAL, .NET DEVELOPER'S JOURNAL, queue, CoDe, XML JOURNAL, wldj, asp.netPRO, WebSphere JOURNAL, COLDFUSION Developer's Journal, SD Times, MX developer's journal

SPECIAL INSERT: WEB SERVICES EDGE 2005
WWW.SYS-CON.COM/EDGE
FEBRUARY 15-17, 2005
JOHN B. HYNES CONVENTION CENTER • BOSTON, MA

International Web Servic

web services conference & expo EDGE

Web Services Edge 2005 East

February 15–17, 2005
John B. Hynes Convention Center
Boston, MA

The Largest *i*-Technology Event of the Year!

Who Should Attend

- Software Developers
- Software Engineers
- Application Developers
- Development Managers
- Technical Directors
- Analysts/Programmers
- IT Managers
- CEOs
- CTOs
- CIOs
- Technical Architects
- Team Leaders
- Software Consultants
- IT Directors
- Project Managers

Features & Attractions

- 3 Days Packed with Education and Training
- Keynotes & Panel Discussions from Industry Leaders
- 50+ Hard-hitting and Informative Seminars
- FREE Web Services Tutorial Presented by Novell
- FREE .NET Tutorial Presented by Microsoft
- FREE Web Services Security Tutorial Presented by Critical Sites
- Cabana Night Developer Exchange Presented by INETA
- Panel Discussions Presented by JCP
- Opening Night Welcome Reception
- Compelling Case Studies & Best Practices
- Hands-On Labs Featuring .NET and Visual Studio Presented by Microsoft Partners
- Featured Product Demonstrations
- Exhibit Floor featuring hundreds of products
- Real-time SYS-CON Radio Interviews

VISIT **www.sys-con.com/edge** FOR TIMES & SCHEDULES

The program, including topics and times, is subject to change. Please refer to www.sys-con.com for all updates.

SPECIAL INSERT: WEB SERVICES EDGE 2005
WWW.SYS-CON.COM/EDGE
FEBRUARY 15-17, 2005
JOHN B. HYNES CONVENTION CENTER • BOSTON, MA

ices Conference & Expo

Keynote Speakers

Tuesday, February 15, 11 a.m.

Matt Ackley
Senior Director, eBay
Developers Program

Web Services for eCommerce

eBay, The World's Online Marketplace, has more than 114 million registered users, 10,000 developers, and over 700 live, third-party applications. Four years ago, eBay began allowing third parties to build applications that tap into eBay, and today eBay hosts one of the leading Web services platforms. Through its developer program, eBay enables third parties to create cutting-edge Web services applications that benefit the buyers and sellers on eBay. At present, 40% of eBay's listings come through its API, which handles more than a billion Web services calls a month. Ackley will discuss the rewards and challenges of building and maintaining one of the world's leading Web services platforms, and share insights and practical guidelines for others.

Matt Ackley is senior director of the eBay Developers Program. He supports eBay's vision to be the leading platform for global online commerce, and is chartered with creating a thriving ecosystem between eBay, its community of users, and third-party developers and solution providers. Ackley joined eBay in 2003 as part of eBay's acquisition of FairMarket, which provided technology solutions and services to online marketplaces.

Wednesday, February 16, 11 a.m.

Ari Bixhorn
Director, Web Services Strategies,
Microsoft Corporation

Introducing Indigo: The Unified Programming Model for Building Service-Oriented Applications

Indigo is Microsoft's unified programming model for building service-oriented applications on the Windows platform. It enables developers to build secure, reliable, transacted solutions that integrate across platforms and interoperate with existing investments. Indigo combines and extends the capabilities of existing distributed application technologies, including .NET Enterprise Services, System.Messaging, Remoting, ASMX, and WSE to deliver a unified development experience spanning distance, topologies, hosting models, protocols, and security models. This keynote will provide an inside look at Indigo and show you how Indigo will radically simplify the development of distributed, service-oriented applications.

Ari Bixhorn is the director of Web Services Strategy in the Developer and Platform Division at Microsoft Corp. He is responsible for product planning and technical evangelism for Microsoft's Web services offerings, including "Indigo," the code name for a component of the next version of the Windows operating system, code-named Windows "Longhorn." Bixhorn has spent the past five years at Microsoft, driving product management efforts for the Visual Basic and Visual Studio development systems.

Thursday, February 17, 11 a.m.

Mike Milinkovich
Eclipse.org

An Open Development Platform for Web Services

Open source technology runs the Internet. Linux, Apache, PHP and Eclipse are highly successful open source communities that provide the backbone for today's Web applications. All indications point to a continued value proposition for organizations for leveraging open source when developing and deploying SOA-based applications. This keynote will examine the benefits of using open source technologies, the decision-making process used when adopting these solutions and the potential for contributing back to the open source community.

Mike Milinkovich has held key management positions at Oracle, WebGain, The Object People, and Object Technology International Inc. (which subsequently became a wholly owned subsidiary of IBM), assuming responsibility for development, product management, marketing, strategic planning, finance, and business development. Mike earned his MS degree in information and systems sciences and a bachelor of commerce degree from Carleton University in Ottawa, Canada.

VISIT **www.sys-con.com/edge** FOR TIMES & SCHEDULES

The program, including topics and times, is subject to change. Please refer to www.sys-con.com for all updates.

SPECIAL INSERT: WEB SERVICES EDGE 2005
WWW.SYS-CON.COM/EDGE
FEBRUARY 15-17, 2005
JOHN B. HYNES CONVENTION CENTER • BOSTON, MA

DAY 1 FEBRUARY 15

	Java	.NET	Web Services
7:30	Registration		
8:00	**FREE Tutorial** – Ashish Larivee, Novell, **Using a Web Services Framework to Build SOA Applications**		
9:00	(J-1) What's New In JDO 2.0	(.NET-1) Intro to SPOT	(WS-1) Ensuring Web Services Interoperability
10:00	(J-2) Using Java Messaging in Real-Time Trading Systems	(.NET-2) An Introduction to SQL Server Reporting Services	(WS-2) Web Services Standards: Going Behind the Mask
11:00	**Opening Keynote** – Matt Ackley, Senior Director, eBay Developer Program, eBay		
12:00	**EXPO OPEN** (12 P.M.–5 P.M.)		
3:00	**Keynote Panel** Presented by JCP – **Web Services and Security** Moderator: Onno Kluyt, Sr Director & Chair, JCP Program, Sun Microsystems		(WS-2B) Solving Complex Business Problems Though SOA
4:00	(J-3) The ROI of a Java-Rich Client	(.NET-3) Go With The Flow – Human Workflow Services in BizTalk 2004 (.NET-3B) Techniques with Visual Basic.NET	(WS-3) The XML Data Challenge
5:00	Opening Night Reception		

DAY 2 FEBRUARY 16

	Java	.NET	Web Services
7:30	Registration		
8:00	**FREE Tutorial –** Thom Robbins, Microsoft – **The Next Generation of Visual Studio** (free with VIP preregistration)		
9:00	(J-4) Web Services End-to-End Security on J2EE: Gaps and Proposed Solutions	(.NET-4) The Microsoft Framework: An Agile Software Development Process for Building Web Service Applications	(WS-4) How To Bulletproof Your Web Services
10:00	(J-5) J2ME and Eclipse	(.NET-5) Securing Service-Oriented Architecture with Microsoft's WSE 2.0	(WS-5) The Role of Policy in Web Services Integration – It's More Than Just Security
11:00	**Keynote** – Ari Bixhorn, Director, Web Services Stategies, Microsoft Corporation		
12:00	**EXPO OPEN** (12 P.M.–4 P.M.)		
3:00	**Application Server Shootout**		
4:00	(J-6) The Impact of JBoss and Mono on the Application Server Market	(.NET-6) Web Services Security for Dummies with WSE2 (.NET-6B) J2EE to .NET Interoperability and App. Integration	(WS-6) B2B Policy Enforcement: The Third Rail of SOA Implementation
5:00	(J-7) Migrating Enterprise Applications Between J2EE Application Servers	(.NET-7) So You THINK You Know What an Object Is… (.NET-7B) Building and Using Advanced ASP.NET Web Controls	(WS-7) Driving SOA Governance
6:00	**Cabana Night –** Hosted by INETA		

DAY 3 FEBRUARY 17

	Java	.NET	Web Services
7:30	Registration		
8:00	**FREE Tutorial** – Patrick Hynds and Duane Laflotte, Critical Sites – **Security, The New Reality** (free with VIP preregistration)		
9:00	(J-8) Design Patterns and Project Organizational Techniques for "Write Once, Debug Everywhere"	(.NET-8) Migrating ASP to ASP.NET	(WS-8) SOA: From Pattern to Production
10:00	(J-9) Using Grid Computing with Web Services and J2EE to Create Internet-based SOAs	(.NET-9) Smart Client Development with the Offline Application Block	(WS-9) High Performance Web Services – Tackling Scalability and Speed
11:00	**Keynote** – Mike Milinkovich, Executive Director, Eclipse Foundation		
12:00	**EXPO OPEN** (12 P.M.–4 P.M.)		
3:00	(J-10) Java Web Services Programming Tips & Tricks	(.NET-10) CLR Internals	(WS-10) So You Want an SOA: Best Practices for Migrating Toward Service Orientation in the Enterprise
4:00	(J-11) JCP Program: How the Java Technology Binary Software Standard is Managed and Evolves	Visit Web site for update	(WS-11) Four Abilities SOA Will Lack Without a Registry

The program, including topics and times, is subject to change. Please refer to www.sys-con.com for all updates.

WS Security	Case Study
FREE Tutorial –Aaron Williams, JCP, **Developing Web Services Using Java Technology**	
(WSS-1) Identity in SOA	(CS-1) Developing E-Commerce Applications with Web Services
(WSS-2) Securing Web Services with WS-Security	(CS-2) Developing Enterprise Class Web Services
	(CS-3) Service-Oriented Development on NetKernel – Patterns, processes and product to reduce the complexity of IT systems
(WSS-3) Anatomy of a Web Services Attack	

Novell

FREE Web Services Security Tutorial Presented by Novell

Using a Web Services Framework to Build SOA Applications

Tuesday, Feb. 15
8:00 A.M. – 11:00 A.M.

FREE Tutorial Presented by Java Community Process

Developing Web Services Using Java Technologies

Tuesday, Feb. 15
8:00 A.M. – 11:00 A.M.

Java Community Process

Free Tutorials with VIP Preregistration ONLY!

WS Security	Case Study
(WSS-4) Using Mobile Phones as an SSO Authentication Device in SOA Solutions	(CS-4) Orchestrating FORCEnet Engagement Packs with BPEL for Web Services
(WSS-5) Building Intelligent Enterprises with Novell's Identity-Driven Computing	(CS-5) CPI: A Globally Integrated Problem-Tracking and Resolution System Using Java Web Services
(WSS-6) XML Content Attacks	(CS-6) The Transformation of SiteRefresh into a Web Services
(WSS-7) The Interoperability Challenge of Web Services Security Standards	

Microsoft

FREE .NET Tutorial Presented by Microsoft

The Next Generation of Visual Studio
Thom Robbins

Wednesday, Feb. 16
8:00 A.M. – 11:00 A.M.

Free Tutorials with VIP Preregistration ONLY!

WS Security	Case Study
n)	
(WSS-8) Transitioning Successfully to SOA and Web Services: Building the Infrastructure for SOA Growth	(CS-8) Using SOA and Web Services to Issue Business Licenses in the District of Columbia
Visit Web site for update	(CS-9) Developing Web Services with Eclipse

critical sites

FREE .NET Tutorial Presented by Critical Sites

Security, The New Reality
Patrick Hynds

Thursday, Feb. 17
8:00 A.M. – 11:00 A.M.

Free Tutorials with VIP Preregistration ONLY!

The program, including topics and times, is subject to change. Please refer to www.sys-con.com for all updates.

SPECIAL INSERT: WEB SERVICES EDGE 2005
WWW.SYS-CON.COM/EDGE
FEBRUARY 15-17, 2005
JOHN B. HYNES CONVENTION CENTER • BOSTON, MA

SPECIAL INSERT: WEB SERVICES EDGE 2005
WWW.SYS-CON.COM/EDGE
FEBRUARY 15-17, 2005
JOHN B. HYNES CONVENTION CENTER • BOSTON, MA

WEB SERVICES TRACK

WS-1 Ensuring Web Services Interoperability

Chris Ferris, IBM

Tuesday, Febuary 15, 2005 9:00 A.M. – 9:50 A.M.

Despite the open industry standards that underlie Web services, interoperability has been a key challenge for vendors and customers implementing Web services. One reason for this is that the relevant industry standards often permit multiple acceptable implementation alternatives. This presentation will discuss in detail the challenge of Web services interoperability and the role played by the premier industry organization formed to address it, the Web Services Interoperability Organization. In particular, the presentation will cover the critical importance of WS-I profiles to an organization's Web services initiatives, including the manner in which companies can put WS-I profiles immediately to work.

BIO: Chris Ferris is chair of the WS-I Basic Profile Working Group and a senior technical staff member with IBM's Emerging Technology Group. He has been actively engaged in open standards development for XML and Web services since 1999 and is an elected member of the OASIS Technical Advisory Board. Chris is also a coauthor and editor of the WS-Reliable Messaging specification.

WS-2 Web Services Standards: Going Behind the Mask

Glen Daniels, Sonic Software

Tuesday, Febuary 15, 2005 10:00 A.M. – 10:50 A.M.

Web services and service-oriented architectures (SOAs) are emerging as an integral part of the enterprise IT strategy. According to a recent IDC study, Web services – related revenue is expected to triple from $1.1 billion worldwide in 2003, to $3.4 billion in 2004, and $16.6 billion by 2008. As SOAs proliferate and the number of Web services added to them increases, standards will play an increasingly significant role. This session will look at the state of key Web services standards such as WS-Choreography, WS-Reliability and WS-ReliableMessaging, SOAP/MTOM/XOP, WSDL, XPath, XQuery, and WS-Notification as well as related Java standards and open source efforts. It will also look at the organizational impact of standards adoption in the industry.

BIO: Glen Daniels is manager of standards and consortia at Sonic Software and coauthor of *Building Web Services with Java.* He has been working with Web services technologies since their inception in the late '90s, and in addition to developing products and helping to found Apache's Axis project, he has been an active participant in standards bodies such as the W3C, and a member of the SOAPBuilders interoperability group.

WS-2B Solving Complex Business Problems Though SOA

John Daly, netNumina

Tuesday, Febuary 15, 2005 3:00 P.M. – 3:50 P.M.

WS-3 The XML Data Challenge

Nancy Vodicka, DataDirect Technologies

Tuesday, Febuary 15, 2005 4:00 P.M. – 4:50 P.M.

Most businesses store and query data with relational databases but need to use Extensible Markup Language (XML) to exchange and display data on the Web and with vendors and partners. As a result, programmers need to deal with both relational and XML data, often at the same time. Emerging standards such as XQuery, XQJ, and SQL/XML, promise to revolutionize data exchange and the ways applications are developed, deployed, and utilized. Learn the key facts about these standards, including what they mean, when they will be available, and what you, the developer, can do to prepare.

BIO: As the XML Product Manager at DataDirect Technologies, Nancy Vodicka is responsible for DataDirect Connect for SQL/XML, a database-independent SQL/XML implementation that is currently shipping, and DataDirect XQuery, a database-independent XQuery implementation that is currently in development. Nancy has more than 15 years experience in the software industry working with technologies such as XML, Web services, relational databases, and SQL.

WS-4 How To Bulletproof Your Web Services

David McCaws, Parasoft

Wednesday, Febuary 16, 2005 9:00 A.M. – 9:50 A.M.

Web services are gaining industry-wide acceptance and usage and are moving from proof-of-concept deployments to actual usage in mission-critical enterprise applications. Web services range from major services such as storage management and customer relationship management to much more limited services such as furnishing stock quotes or providing weather information. As companies and consumers begin to rely more and more on Web services, the need for developing reliable, high-quality Web services is even stronger. This session will explain issues specific to Web services and will illustrate solid engineering and testing practices required to ensure complete Web service functionality, interoperability, and security. Whether creating Web services from scratch or integrating a legacy back-end server via Web services, the practices and principles outlined in this session will be of great benefit.

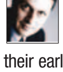
BIO: David McCaw has over eight years of experience in helping software development teams improve quality throughout the development process. Over the last three years, he has led the Parasoft Web Services Solutions team, which has developed an industry-leading approach for Web services testing. He has implemented Web service quality solutions for development groups in organizations such as Sabre-Holdings, Yahoo! Overture, and McGraw-Hill. McCaw has an extensive background in the areas of Java and Web service reliability, performance, and security. He is involved with OASIS and WS-I, and is a frequent speaker at industry events.

WS-5 The Role of Policy in Web Services Integration – It's More Than Just Security

Toufic Boubez, Layer 7 Technologies

Wedneday, Febuary 16, 2005 10:00 A.M. – 10:50 A.M.

Too often today the preferences, terms, and conditions describing how a Web service behaves when discovered and invoked is programmed right into the business logic. Hard-coding this behavior logic however introduces cost, complexity, and rigidity into a Web services architecture. A better approach is to abstract a Web services usage "policy" out of code where this metadata can be managed as need be. This session introduces the concept of Web Services Policy and describes how the construct can be used to implement a more customized and versatile Web service infrastructure.



BIO: Toufic Boubez is a well-respected and renowned Web services visionary. Prior to cofounding Layer 7 Technologies, Toufic was the chief Web services architect for IBM's Software Group and drove their early XML and Web services strategies. He is a sought-after presenter and has chaired many XML and Web services conferences. He is an author of many publications and his most recent book is the top-selling *Building Web Services with Java: Making Sense of XML, SOAP, WSDL ,and UDDI.*

WS-6 B2B Policy Enforcement: The Third Rail of SOA Implementation

Alistair Farquharson, Digital Evolution

Wednesday, Febuary 16, 2005 4:00 P.M. – 4:50 P.M.

One of the great benefits of a service-oriented architecture is the ability it gives you to extend programmatic, integration capabilities to business part-

VISIT **www.sys-con.com/edge** FOR TIMES & SCHEDULES

The program, including topics and times, is subject to change. Please refer to www.sys-con.com for all updates.

ners. Going beyond simple sharing of data with partners, SOA enables true B2B application integration. At the same time, this capability creates a vexing security policy enforcement dilemma. How can you be sure that a user from a partner organization is actually authorized to integrate with your applications? How can you authenticate that user? Do you even want that headache in the first place? This session will discuss the issues that arise in B2B security policy enforcement and explore several proven approaches to solving the problem. In particular, it will focus on the emerging technology of XML Virtual Private Networks (XML-VPNs) and their potential to mitigate security policy enforcement issues in B2B SOA implementations.

BIO: Alistair Farquharson is the CTO of Digital Evolution, where he spearheads product development and provides thought leadership to enterprise customers implementing Web services. His skills span many industries and include designing and implementing system architectures, as well as spearheading initiatives such as development/team lead. He is an expert in custom-application development, distributed environments, architecting scalable hardware and software applications and systems, and Web services application development.

WS-7 Driving SOA Governance

Brent Carlson, LogicLibrary

Wednesday, Febuary 16, 2005 5:00 P.M. – 5:50 P.M.

In the past year, Web services and service-oriented architectures (SOAs) have become mainstream because of their ability to provide business agility and flexibility through integration, productivity, and reuse. With SOA enablement on the rise, IT groups must address SOA governance as a means of controlling what and how services located within an SOA are deployed. This session will discuss SOA governance, specifically how an organization can manage and control assets and artifacts located within an enterprise, while ensuring that deployed assets meet an organization's business and technical architectural standards. It will also outline governance best practices such as monitoring the UDDI publish process in order to seamlessly tie together the development and operational views of Web services within the enterprise.

BIO: Brent Carlson drives the development and delivery of LogicLibrary's products. He is a 17-year veteran of IBM, where he served as lead architect for the WebSphere Business Components project and held numerous leadership roles on the "IBM San Francisco Project." He is a member of the Eclipse Board of Stewards and a BEA Regional Director.

WS-8 SOA: From Pattern to Production

David Chappell, Sonic Software

Thursday, Febuary 17, 2005 9:00 A.M. – 9:50 A.M.

Service-oriented architecture (SOA) represents the opportunity to achieve broad-scale interoperability, while providing the flexibility required to continually adapt technology to business requirements. No small feat, particularly when one considers the extent and complexity of today's IT environments. As both a technology concept and IT discipline, the challenge inherent in SOAs is maintaining the right architectural approach. If all services in an SOA are treated as interdependent point-to-point interfaces, then the complexity of implementing and maintaining them in this spaghetti-like architecture becomes enormous. The enterprise service bus (ESB) has emerged as one of the first true SOA product offerings, bringing SOA from pattern to production. ESBs provide a framework for building and deploying an event-driven, enterprise SOA and accommodates the configuration, hosting, and management of integration components as services across the business.

BIO: VP and chief technology evangelist for Sonic Software, Dave Chappell has over 18 years of experience in the software industry covering a broad range of roles including R&D, codeslinger, sales, support, and marketing. He also has extensive experience in distributed computing, including message-oriented middleware, CORBA, COM, and Web application server infrastructure.

WS-9 High Performance Web Services – Tackling Scalability and Speed

Sameer Tyagi, Sun Microsystems

Thursday, Febuary 17, 2005 10:00 A.M. – 10:50 A.M.

Web services facilitate application-to-application integration and interoperability across different platforms. However, critics usually point to an inefficient processing model and bandwidth requirements for developing Web services. This is often cited as a reason why Web services cannot perform and scale well in production environments. This session takes a detailed look at performance and scalability issues around Web services in the real world, as well as strategies that architects and developers can adopt to mitigate such risks in these applications. Some analytical and modeling strategies that enable acceptable application performance will also be covered.

BIO: Sameer Tyagi works as a senior Java architect with Sun Microsystems. He remains focused on architecture, design, and implementation of large-scale enterprise applications with Java technology. His publications include industry periodicals and books on Java and J2EE technologies including *Java Web Services Architecture.*

WS-10 So You Want an SOA: Best Practices for Migrating Toward Service Orientation in the Enterprise

Eric Newcomer, IONA

Thursday, Febuary 17, 2005 3:00 P.M. – 3:50 P.M.

Replacing complex, monolithic applications with nimble applications built from exposed services promises increased developer productivity, greater flexibility, and ultimately reduced cost. The adoption of Web services and SOA can also remove a significant level of complexity and integration problems from enterprise application development projects. But, as with any large-scale project, IT departments must have the right plan and the right resources in place to ensure a successful transformation of their computing infrastructure. This article will explore what IT organizations need to know to be successful in their attempts to migrate the enterprise to a service-oriented architecture.

BIO: In the role of chief technology officer at IONA, Eric Newcomer is responsible for IONA's technology roadmap and the direction of IONA's e-business platforms as relates to standards adoption, architecture, and product design.

WS-11 Four Abilities SOA Will Lack Without a Registry

Luc Clement, Systinet

Thursday, Febuary 17, 2005 4:00 P.M. – 4:50 P.M.

A service-oriented architecture (SOA) is the design blueprint for seamless connectivity between business processes and IT infrastructure, enabling innovation and improving productivity. SOA provides the most efficient, standard way to dynamically interoperate with any customer, supplier, product or employee. SOA makes integration intrinsic. Web services are the foundation building blocks of an SOA, and they are already proliferating inside most enterprises. In an SOA, Web services become business services with the ability to perform a particular function or access data dynamically. This presentation will discuss the four abilities that a registry provides for an SOA.

BIO: Luc Clement is director of product marketing, SOA Registry for Systinet. He is also cochair for the UDDI Specification Technical Committee. Formerly Microsoft UDDI Program Manager, Luc is well known in the UDDI community and has been heavily involved with the UDDI specification for several years.

Register by

JANUARY 21, 2005

SAVE UP TO $300

Register Today!

www.sys-con.com/edge2005

SPECIAL INSERT: WEB SERVICES EDGE 2005
WWW.SYS-CON.COM/EDGE
FEBRUARY 15-17, 2005
JOHN B. HYNES CONVENTION CENTER • BOSTON, MA

The program, including topics and times, is subject to change. Please refer to www.sys-con.com for all updates.

International Web Servic

WEB SERVICES SECURITY TRACK

WSS-1 Identity in SOA

Sekhar Sarukkai, Oblix

Tuesday, Febuary 15, 2005 9:00 A.M. – 9:50 A.M.

The mainstreaming of SOAs requires a more general approach to the notion of identities – beyond simply central management of people identities and into the realm of managing applications, devices, and other identities that represent entities that are first-class participants in this application network while also providing this as a pluggable service into the larger enterprise SOA. Enterprises should view identity as a service that is ubiquitously available and is a shared infrastructure service necessary for application networking, rather than as being managed by a server, such as an Authentication or Access server. While it makes architectural sense to consider an Identity service, there are business and related drivers that may force the need to deploy such an architecture.

BIO: Sekhar Sarukkai is currently a technical architect at Oblix. He was the original founder and CTO of Confluent Software, a leading Web services management company, which was acquired by Oblix in 2004. He holds a PhD in computer science from Indiana University.

Attend a FREE* Tutorial

Thursday, February 17, 2005, 8 – 11 a.m.

Using a Web Services Framework to Build SOA Applications

Ashish Larivee

Ashish Larivee, Novell

Service-oriented architecture (SOA) has quickly taken center stage as the primary development style of the next decade and beyond. Businesses of all types are preparing for the SOA revolution that promises consistency of process, reduction in duplicate work, ease of maintenance, service reusability and broad interoperability. The Web Services Framework (WSF) is the foundation that can deliver on the promise of SOA. Come learn about the components of an SOA including the core WSF standards. Attend this free Novell tutorial and learn about the future of SOA-style development, including legacy system enablement, platform interoperability, open source in SOA and building composite applications that leverage SOA services using Novell exteNd. In this session, we will create SOA application logic that orchestrates legacy services, JBoss4 Web services and MS.Net Web services. We will create Web services in Novell exteNd, Eclipse and Visual Studio respectively. We will then orchestrate these Web services and expose a single course, process level interface to public Web service consumers. Finally, we will cre-

Novell.

* Free Tutorials with VIP Preregistration ONLY!

WSS-2 Securing Web Services with WS-Security

Dr. Jothy Rosenberg, Service Integrity

Tuesday, Febuary 15, 2005 10:00 A.M. – 10:50 A.M.

An up-to-date, comprehensive, and practical discussion of Web services security, and the first to cover the final release of new standards SAML 1.1 and WS-Security. Comprehensive coverage and practical examples of the industry standards XML Signature and XML Encryption will be presented.

BIO: Dr. Jothy Rosenberg is a serial entrepreneur. He is a founder and CTO of Service Integrity, a company that helps Web service operators see, measure, understand and fully leverage operational and business information flowing across their Web service networks. Prior to this venture, Jothy cofounded GeoTrust, the world's second largest certificate authority.

WSS-3 Anatomy of a Web Services Attack

Mamoon Yunus, Forum Systems

Tuesday, Febuary 15, 2005 4:00 P.M. – 4:50 P.M.

A broad range of new security threats is facing enterprises implementing XML Web services, leaving the enterprises open to financial risks, loss of property, and tarnished reputations. The basic rules of security – authentication, authorization, and auditing – no longer provide adequate security in the new world of straight-through processing paths into mission-critical systems. What's worse, WSDL documents provide a guide book to security exposure. Most attacks on traditional Web-based applications exploit weaknesses in HTML-enabled custom, or packaged, applications. However, hackers and other malicious users are quickly uncovering new techniques at the SOAP/XML data level that bypass HTML and target weaknesses in Web services programming, technology, and architecture. This session will outline the innovative techniques that hackers use to map out the vulnerabilities of an organization's network, and how Web server security must now complement Web services security in order to provide an adequate defense.

BIO: Mamoon Yunus, CTO of Forum Systems, was previously a global systems engineer for webMethods, where he developed business integration strategy and architecture for Global 2000 companies. He is an industry-honored CTO in advanced technological solutions for enterprise customers.

WSS-4 Using Mobile Phones as an SSO Authentication Device in SOA Solutions

Dr. Michael Juntao Yuan, University of Texas

Wednesday, Febuary 16, 2005 9:00 A.M. – 9:50 A.M.

Federated identity management across multiple single-sign-on domains is a major challenge for SOA-based solutions to fully realize its business potential. The traditional username/password combination is often too weak to protect the extremely sensitive single-sign-on credentials. The new-generation mobile phones could be used to identify and authorize users for SOA services. The device-based authentication scheme depends on not only "what you know" but also "what you own." This session will discuss new advances in Java-based mobile devices to interoperate with Sun's Liberty Alliance Services.

BIO: Dr. Michael Juntao Yuan is an author, developer, and software architect for end-to-end mobile software. He is a contributing editor to *JDJ* and a frequent contributor to many developer forums and publications. He is the author of two books. Michael has a PhD from the University of Texas at Austin and teaches information systems at the college level.

WSS-5 Building Intelligent Enterprises with Novell's Identity-Driven Computing

Ashish Larivee, Novell

Wednesday, Febuary 16, 2005 10:00 A.M. – 10:50 A.M.

Companies are now facing complexities dealing with issues such as regulatory compliance and security while still providing for company-wide collaboration between employees, partners, and suppliers. Identity systems are becoming a crucial component of applications, enabling developers to take advantage of a new set of services that know who you are, where you are, what you are trying to do, and can adapt to your changing business needs. Identity-driven computing addresses these problems by applying best practices learned from Novell's leadership in identity management for the management of people to all aspects of an enterprise, including servers, PCs, devices, applications, and even Web services.This presentation will outline identity-driven computing, describe the attributes of an identity-driven application, and discuss steps enterprises can take to make the transition to an identity-driven computing environment.

BIO: With more than nine years of experience in the software industry, Ashish Larivee has designed and developed many enterprise applications across a variety of platforms, including Microsoft, Lotus Notes/Domino, and J2EE. In 1999, Ashish joined SilverStream Software, acquired by Novell in July 2002, and has served in various roles in consulting, development, and technical marketing. In her current role, she helps define the strategy and product direction across Novell's Web Application Development Products.

VISIT **www.sys-con.com/edge** FOR TIMES & SCHEDULES

The program, including topics and times, is subject to change. Please refer to www.sys-con.com for all updates.

SPECIAL INSERT: WEB SERVICES EDGE 2005
WWW.SYS-CON.COM/EDGE
FEBRUARY 15-17, 2005
JOHN B. HYNES CONVENTION CENTER • BOSTON, MA

WSS-6 XML Content Attacks

Girish Juneja, Sarvega, Inc.

Wednesday, Febuary 16, 2005 4:00 P.M. – 4:50 P.M.

This talk defines a new class of threats, XML Content Attacks, and differentiates these threats from more general Web services attacks and XML security-based attacks. These three related but distinct threat areas are explained. The talk covers XML Content Attacks with regard to tree-based parsing exploits related to coercive parsing, node-depth attacks, and DOM. XML grammar validation exploits such as schema poisoning and lax-content models are discussed, and why traditional schema validation cannot ensure content-model consistency. Web services attacks like WSDL scanning and parameter tampering (SQL Injection, SOAP array attack) are discussed – highlighting common mistakes made when applying message-level security (WS-Security).

BIO: Girish Juneja has more than 15 years' experience in the high technology industry with extensive product management, product strategy, engineering management, and technology marketing expertise. He is the cofounder of Sarvega. Since Sarvega's inception, Girish has led the Sarvega engineering and customer services organizations to develop Sarvega's industry-leading core XESOS technology and XML Networking products.

WSS-7 The Interoperability Challenge of Web Services Security Standards

Eve Maler, Sun Microsystems

Wednesday, Febuary 16, 2005 5:00 P.M. – 5:50 P.M.

The Web Services Interoperability Organization chartered its Basic Security Profile Working Group to develop an interoperability profile involving transport layer security, SOAP message layer security, encryption, signatures, and other security considerations. This session will discuss the interoperability challenges presented by current Web services security standards and the work of the WS-I Basic Security Profile. The session will highlight typical Web services security threats and countermeasures and the related design goals, usage conventions, and conformance testing of the soon-to-be-released Basic Security Profile.

BIO: Eve Maler is an XML standards architect at Sun Microsystems, where she coordinates Sun's involvement with Web services security standards such as SAML and the WS-I Basic Security Profile.

WSS-8 Transitioning Successfully to SOA and Web Services: Building the Infrastructure for SOA Growth

Dan Foody, Actional

Thursday, Febuary 17, 2005 9:00 A.M. – 9:50 A.M.

This session will address how to approach service-oriented architecture (SOA) management from a project-based level while still allowing room for future expansion and incremental growth to an enterprise-wide SOA. The session will provide valuable insight into how SOA management can help organizations ease the complexity of moving toward a loosely coupled environment.

BIO: As CTO at Actional, Dan Foody leverages his extensive experience in enterprise systems software toward designing robust and manageable service-oriented architectures. He is an active participant in the Web services standards community, including WS-I and OASIS, where he spearheads Actional's contributions on the OASIS Web Services Distributed Management Committee (WSDM).

WEB SERVICES EDGE CONFERENCE PAST SPONSORS & EXHIBITORS

Actional · AltoWeb · Antenna House · Covasoft · CTI · DIRIG Software · eXcelon · FatWire Software · webGain · attachmate · Cachéon · Connotate · Apress
Microsoft · Corda · atg · Full Tilt · Sun microsystems · webputty · Iopsis Software · Rational the e-development company · TIBCO The Power of Now
Seagull · aligo · OneWebSystems · bea · Bestning XML Security · Cape Clear · CodeHound.com · Compuware · Data Junction · DataMirror · Entegrity Solutions · Vincera
HiT Software · Parasoft · InCert · Inferdata · infoShark · ReportMill Software · LearningPatterns.com · Merant · Novalabs · pingtel · PyBiz · Rogue Wave Software
spiritsoft · Oracle · Borland · Altoweb · Matisse · IPEDO · PolarLake · Vordel · Web Age · Altaworks
bowstreet · CSC · hp invent · iam · Information Laboratory · pingtel · iWay Software · McCabe & Associates · Nextel · Object Design · Pramati · Infravio · Killdara corporation
Mercury Interactive · isavvix · Jinfonet Software · KeyLabs · macromedia · MelissaData · Metaserver · Micro Focus · Vultus · webMethods · oakgrove
OpenDemand · Our World Live · PointBase · Sybase · Trivera · Wakesoft · ObjectVenture · Optimyz · plumtree · teamshare · Trellix

VISIT **www.sys-con.com/edge** FOR TIMES & SCHEDULES

The program, including topics and times, is subject to change. Please refer to www.sys-con.com for all updates.

SPECIAL INSERT: WEB SERVICES EDGE 2005
WWW.SYS-CON.COM/EDGE
FEBRUARY 15-17, 2005
JOHN B. HYNES CONVENTION CENTER • BOSTON, MA

International Web Services Conference & Expo

Register by JANUARY 21, 2005 SAVE Up To $300

For Assistance Tel: 201-802-3066 Fax: 201-782-9601

CONFERENCE & EXPO: February 15 – 17, 2005

John B. Hynes Veteran Memorial Convention Center • Boston, MA

THREE WAYS TO REGISTER FOR CONFERENCE

1) On the Web: Credit Cards or "Bill Me" Please make checks payable to SYS-CON Events
2) By Fax: Credit Cards or "Bill Me" 201-782-9601
3) By Mail: 135 Chestnut Ridge Road, Montvale, New Jersey 07645, Attention: Registration

Please note: Registrations are not confirmed until payment is received.

Please complete sections 1, 2, 3 and 4

1 YOUR INFORMATION (Please Print) □ Mr. □ Ms. □ Mrs.

First Name ______________________Last Name ______________________

Title ______________________

Company ______________________

Street ______________________

Mail Stop ______________________

City ______________________

State______________________Zip ______________Country ______________

Phone ______________________

Fax______________________E-Mail ______________________

2 PAYMENT METHOD: (Payment in full due with registration)

□ Check or Money Order Enclosed (Registration confirmed upon receipt of payment)

Check #______________________ Amount of Check $ ____________

Charge my □ Visa □ MasterCard □ American Express □ Discover

Name on card______________________

Card #______________________ Exp. Date____________

Signature______________________

Billing Address (if different from mailing address)

____

3 PLEASE INDICATE YOUR CONFERENCE CHOICE:

Total Registration fee $____________

	Before 12/31/04	Before 1/21/05	Before 2/11/05	On Site
□ **3D: Three Day Conference** Good for all three days of the conference program, including keynotes, all conference sessions.	$1,195.00	$1,395.00	$1,495.00	$1,695.00
□ **2D: Two Day Conference** Good for two days of the conference program, including keynotes, all conference sessions.	$995.00	$1,195.00	$1,295.00	$1,395.00
□ **1D: One Day Conference** Good for one day of the conference program, including keynotes, all conference sessions.	$595.00	$695.00	$695.00	$795.00
□ **VIP PASS** FREE Tutorials with VIP preregistration □ FREE - Novell Tutorial (Feb. 15) □ FREE - Developing Web Services Using Java Technology (Feb. 15) □ FREE - The Next Generation of Visual Studio (Feb. 16) □ FREE - Security, The New Reality (Feb. 17)	FREE	FREE	FREE	$75.00
□ **EO: Expo Hall Only** Includes keynotes, expo presentations, and workshops	FREE	FREE	FREE	$50.00

SPECIAL GROUP RATES

Registering a group of four or more? Please call Jim Hanchrow at 201-802-3066, and receive an additional discount.

4

A. Your Job Title

- □ Senior Software Developer/Engineer/Architect/Analyst
- □ Director of IT/Development/Engineering
- □ Software Developer/Engineer/Architect
- □ Systems Architect/Engineer/Analyst
- □ Software Consultant/Analyst
- □ Project Manager/Team Leader
- □ Business Development
- □ General Manager/Division or Department Head
- □ CTO,CIO, Chief Architect
- □ CEO, President, Owner, Partner/Principal
- □ VP/Technical Director
- □ Network Administration/Manager/Consultant
- □ Web Developer/Programmer
- □ Database Administrator
- □ VP/Sales & Marketing Director /Product Management
- □ Other (please specify)______________________

B. Business/Industry

- □ Agriculture/Mining/Oil/Gas
- □ Computer Hardware and Electronics
- □ Construction/Architecture/Engineering
- □ Consulting (non-computer)
- □ Education
- □ Banking/Insurance/Accounting
- □ Government (including Military)
- □ Healthcare/Pharmaceuticals/Biotech/ Biomedical
- □ Internet/Web/e-Commerce
- □ Manufacturing and Process (other than computer related)
- □ Media/Marketing/Advertising
- □ Non-Profit/Trade Association
- □ Real Estate/Legal
- □ Telecommunications
- □ Transportation/Rail/Sea/ Air/Ground
- □ Travel/Hospitality/Entertainment
- □ Utilities
- □ Wholesale/Trade/Distribution/ Retail
- □ Other (please specify) ______________

C. Total Number of Employees at Your Location and Entire Organization (check all users that apply):

	Location	Company
10,000 or more	01 □	01 □
5,000 - 9,999	02 □	02 □
1,000 - 4,999	03 □	03 □
500 - 999	04 □	04 □
100-499	05 □	05 □
100 or less	06 □	06 □

D. Please indicate the value of software products and services that you recommend, buy, specify or approve over the course of one year:

- □ $10 million or more
- □ $1 million - $9.9 million
- □ $500,000 - $999,999
- □ $100,000 - $499,999
- □ $10,000 - $99,999
- □ Less than $10,000
- □ Don't know

- □ $10 billion or more
- □ $1 billion - $9.9 billion
- □ $100 million - $999 million
- □ $10 million - $99.9 million
- □ $1 million - $9.9 million
- □ Less than $1 million
- □ Don't know

E. What is your company's gross annual revenue?

F. Do you recommend, specify, evaluate, approve or purchase mainframe products or services for your organization?

01 □ Yes 02 □ No

G. Which of the following products, services, and/or technologies do you currently approve, specify or recommend the purchase of?

- □ Application Servers
- □ Web Servers
- □ Server Side Hardware
- □ Client Side Hardware
- □ Wireless Device Hardware
- □ Databases
- □ Java IDEs
- □ Class Libraries
- □ Software Testing Tools
- □ Web Testing Tools
- □ Modeling Tools
- □ Team Development Tools
- □ Installation Tools
- □ Frameworks
- □ Database Access Tools / JDBC Devices
- □ Application Integration Tools
- □ Enterprise Development Tool Suites
- □ Messaging Tools
- □ Reporting Tools
- □ Debugging Tools
- □ Virtual Machines
- □ Wireless Development Tools
- □ XML Tools
- □ Web Services Development Toolkits
- □ Professional Training Services
- □ Other [Please Specify]

____

SYS-CON Events, Inc, and SYS-Con Media make no warranties regarding content, speakers or attendance. The opinions of speakers, exhibitors and sponsors do not reflect the opinion of SYS-CON Events and SYS-CON Media and no endorsement of speakers, exhibiting companies products, or sponsors is implied.

SYS-CON EVENTS

If you require special assistance covered under the Americans with Disabilities Act, please call 201-802-3066 by February 4, 2005.

CANCELLATIONS, SUBSTITUTIONS, REFUNDS

Fax written request to SYS-CON Registration 201-782-9601. Requests for refunds received prior to January 7, 2005 will be honored, less a 10% handling charge; requests received after January 7, 2005, and before February 4, 2005, will be honored less a 20% handling charge. No requests for refunds will be honored after February 4, 2005. Requests for substitutions must be made in writing prior to February 11, 2005. No one under 18 is permitted to attend. No warranties are made regarding the content of sessions or materials.

Speakers, sessions and schedule are subject to change without prior notice.

No solicitation by anyone other than official exhibitors, sponsors or marketing partners is permitted. Such behavior is cause for expulsion without refund.

XML JOURNAL
SUPPLEMENT

HOME
ENTERPRISE SOLUTIONS
CONTENT MANAGEMENT
DATA MANAGEMENT
XML LABS

This Month

XAML: A World of Opportunity

BY PAUL COLTON

Programming techniques continue to evolve, and with evolution comes opportunity. The success of the Web, in conjunction with the demand for rich clients and rapid development, has spawned a new generation of declarative languages. XUL, Macromedia's Flex, Microsoft's XAML, and Xamlon, Inc.'s Xamlon Pro 1.1 are designed to enable rapid, XML-based application develoment with access to structured programming languages for application logic.

XML Schema Binding with XMLBeans

BY DEEPAK VOHRA

XMLBeans is an open source XML-Java binding tool used to generate Java classes and interfaces from an XML Schema. The generated Java classes may be used to parse or generate an XML document that conforms to the Schema. Some of the advantages of XMLBeans over JAXB are the ability to parse an XML document and support for all of the XML Schema constructs; the JAXB-generated classes do not have a parse method to parse an XML document, and JAXB does not support all of the Schema constructs.

Combining XQuery and Web Services

BY JEFF DEXTER

The XML world is driving new emerging technologies that can be utilized to provide solutions to a variety of problems. This article focuses on two of these technologies: Web services and XQuery. As separate entities, these technologies provide a powerful set of features; but when combined they have the potential to present ever more sophisticated feature sets designed for very specific goals.

XML-Based Interop, Close Up

In 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

XML Labs:
Product reviews, book reviews, tutorials, and standards analysis

XAML: A World Of Opportunity

New declarative language opens many doors for many entities

WRITTEN BY
PAUL COLTON

Programming techniques continue to evolve, and with evolution comes opportunity. The success of the Web, in conjunction with the demand for rich clients and rapid development, has spawned a new generation of declarative languages. XUL, Macromedia's Flex, Microsoft's XAML, and Xamlon, Inc.'s Xamlon Pro 1.1 are designed to enable rapid, XML-based application development with access to structured programming languages for application logic.

There is a myriad of advantages to declarative user interface development. Markup-based application interfaces are quick to build and easy to modify. HTML and other markup languages suffer from limited programmatic functionality and control, but the new generation of declarative languages circumvents this traditional limitation by offering back-end scripting languages, in the case of XUL and Flex, or access to any .NET language for the back end in the case of XAML from Microsoft or Xamlon. Finally, as described below, XAML's separation of application interface from application logic offers benefits to developers and customers.

In the remainder of this article I will define XAML and describe its advantages. Then I will discuss the benefits of XAML for various software businesses: consulting firms, independent software vendors (ISVs), and corporations with internal customers and in-house development teams. Finally, I'll speculate on how XAML might evolve and how the market will grow beyond the Windows platform.

What Is XAML?

XAML is an XML-based syntax for separating user interface definition from program logic code. Put another way, XAML is a declarative syntax that provides user interface control while relegating programmatic functionality and application logic to "code behind" files that use structured languages. XAML was developed by Microsoft as the user interface language for its next-generation presentation layer, code-named Avalon. Xamlon's XAML is available now and Microsoft's XAML is currently slated for release in 2006.

The XAML Revolution

XAML is a revolution in Windows application development. For the first time, Windows application interfaces can be written declaratively, and all the benefits of markup apply. XAML offers those benefits to current .NET developers, opens Windows application creation to a whole new group of programmers, and for the first time enables Windows applications to be published unchanged on the Web.

With XAML developers can quickly create and easily modify Windows application interfaces. Markup development is fast. It requires less code and results in fewer bugs than traditional structured programming. The separation of user interface code from application logic provided by the XAML model means that programmers can easily modify interfaces without changing the underlying application code. The interface/logic separation also protects code; changes to the user interface cannot introduce new bugs into the application logic.

> "XAML is the natural choice for a new cross-platform interface language"

Current .NET developers benefit from XAML. User interface development, traditionally a chore for programmers, is faster and more flexible. The XAML methodology makes it easy for one developer or development team to create the application interface while another developer or team implements the application logic. Once interface events are defined, user interface code can be built in isolation from application code. A .NET developer can choose to hand interface creation off to designers and markup developers, focusing on the application logic instead.

XAML opens Windows development to two new, large constituencies: graphic designers and markup developers.

Developers and designers who cut their teeth creating Web pages can now build Windows interfaces, harnessing the full power of the Windows presentation layer. At the same time, XAML offers far more user interface flexibility, freeing designers, developers, and ultimately end users from the tyranny of the boring blue rectangular window and enabling a boom in skins and interesting custom interfaces. Markup developers and designers become "A-level" contributors, and the huge Windows application space is now a market for them.

Finally, by using XAML, Windows applications can be delivered unchanged via the Web to Windows clients. .NET developers can build rich Internet applications ("smart clients" in Microsoft's terminology) with the full functionality of Windows, and deliver them to any connected Windows machine in the world.

Business Opportunities

This new model for Windows and Web development has tremendous potential for forward-thinking developers and businesses. A new development methodology, created and heavily promoted by Microsoft, provides an unmatched opportunity for astute developers and consultancies to gain early experience and become recognized XAML experts. ISVs can be first-to-market with forward-compatible, XAML-based applications, and corporations can easily build and deliver applications across geographically distributed operations.

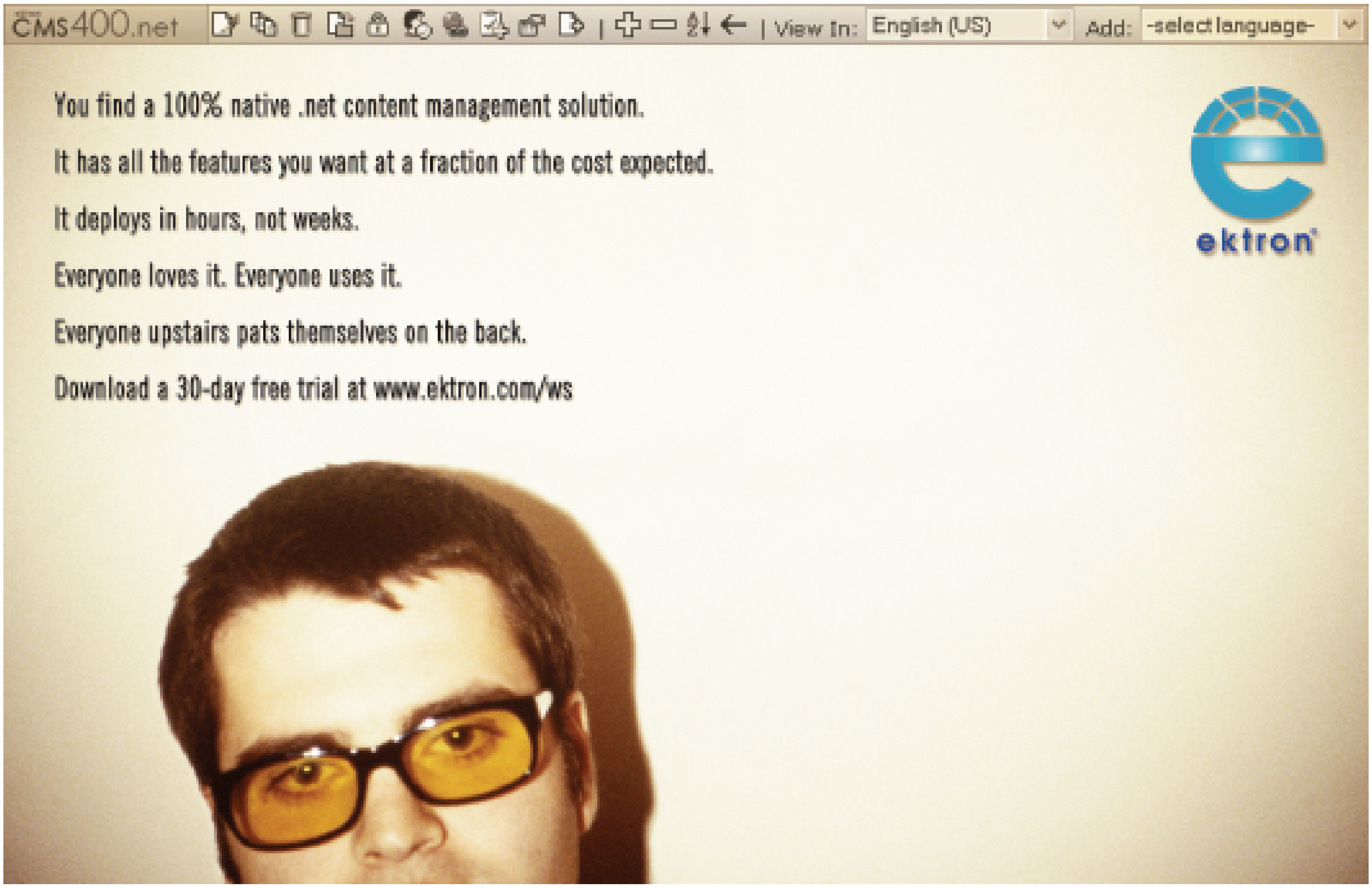
Consulting Firms

Independent developers and consulting firms have a brief window of opportunity to be early adopters of XAML. As Avalon's release date nears, Microsoft will flex its marketing and educational muscles to promote Avalon and XAML. As XAML awareness grows, more organizations will move to XAML development and look for developers and consultancies able to ease this transition.

Cutting-edge expertise is a powerful reason for consulting firms to start learning XAML today, however, there are other persuasive arguments as well. Because markup development is fast, projects are completed faster and at a lower cost. Since markup developers are generally less expensive than other programmers, projects built with XAML can have higher profit margins than traditional .NET development projects while delivering great value to the client. Finally, since XAML is for Windows XP and beyond, budgets for XAML may well include hardware and operating system upgrades.

ISVs

Independent software vendors (ISVs) also gain from early XAML adoption. First and foremost, no ISV wants to create software that is outdated when released. .NET applications released around 2006 that do not take advantage of XAML will be in that unenviable position. Customers want software that uses current technologies, and in the near future that will mean XAML-based applications.



XAML offers other important benefits to ISVs. Rapid declarative development can significantly reduce development costs. An easily modifiable user interface makes it simple to create custom screens and data views for clients, and easy to "brand" software with a client's look and feel. Finally, seamless delivery across the Web with no code changes allows ISVs to offer product deployment via the Web with no additional development effort.

ISVs are already exploring the XAML opportunity. As I write this, I'm corresponding with three companies interested in building their next-generation products with XAML. One sells management software for communications infrastructure providers. The second provides cross-platform business framework software, and the third makes controls and widgets for .NET. All three of these ISVs want to take advantage of the benefits discussed above. The first two want to easily customize interfaces for their clients and offer Web deployment, without increasing their product cost or development time. The third ISV wants to be among the first to offer XAML-based controls; they have a history of success in the .NET market, believe there will be a significant pool of XAML developers, and see an opportunity to be the leading provider of XAML controls.

Corporations

Corporations with internal development teams and internal customers can profit from XAML as well. With it, corporations gain the flexibility of client-side applications with the convenience of server-side control. Far-flung employees can access rich corporate software over the Web, ensuring that new and updated applications are available immediately across a company's entire operation.

> "XAML opens Windows development to two new, large constituencies: graphic designers and markup developers"

Corporations also benefit from the other advantages of XAML. The ability to quickly and easily customize application interfaces for internal customers helps in-house development teams satisfy corporate needs. Decreased development time, less code, and fewer bugs are a boon for internal developers, and the associated reduced development costs are a boon for corporations.

To sum up, the rapid development, reduced costs, and increased customer satisfaction offered by markup-based user interface development are a tremendous win for corporate developers, for ISVs, and for development consulting firms.

Beyond XAML

XAML is a new syntax, but already developers and software firms are exploring ways to build on the foundation it provides. Separating application interface and application logic development is a great step, but limitations and challenges remain. Two potential directions for XAML evolution are extending its syntax by adding programmatic functionality, and extending its reach by enabling cross-platform XAML development.

Extended XAML

The addition of programmatic functionality to XAML will make it more attractive to many markup developers. As has been discussed, XAML is designed to leave most logic in code behind files. Markup developers, however, often find it useful to have certain functionality, such as data binding, variable manipulation, and flow control, available via markup. The demand generated by this large group of developers makes it inevitable that companies will provide products that extend XAML, and fortunately XAML has been designed from the ground up to be extensible.

Cross-platform XAML

XAML is an obvious choice for a cross-platform user interface language. XAML was invented by Microsoft for building Windows applications, but many programmers using XAML have markup experience on the Web and other platforms. A logical next step, and one that developers will eventually demand, is an engine that converts XAML to other languages, such as Flash or Java. With this evolution, developers will be able to create a user interface with XAML that can be deployed, unchanged, across multiple platforms.

Conclusion

XAML is a winner. As the cornerstone of Microsoft's new presentation subsystem, XAML will be heavily promoted, and over time, an abundance of XAML products and services will be available from Microsoft and others. As a simple method for developing Windows user interfaces, XAML will be embraced by developers and organizations looking for decreased development time and increased flexibility when building Windows applications. With its declarative syntax, Windows functionality, and Web friendliness, XAML is a natural choice for a new cross-platform interface language. Take a good look at XAML today.

AUTHOR BIO

Paul Colton is the founder and CEO of Xamlon, Inc., a leading provider of cross-platform markup development tools. Prior to Xamlon, Paul founded Live Software, creator of JRun, the leading Java servlet engine now sold by Macromedia, Inc.

FEEDBACK@XAMLON.COM

A LIMITED TIME SAVINGS OFFER FROM *SYS-CON MEDIA*

SUBSCRIBE TODAY TO MULTIPLE MAGAZINES

AND SAVE UP TO $340 AND RECEIVE UP TO 3 FREE CDs!*

CALL TODAY! 888-303-5282

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

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

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

□ **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	You Pay: $69.99 /	Save: $12
Can/Mex - Two Years (24) $168	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 Pay: $89.99 /	Save: $6
Int'l - Two Years (24) $216	You Pay: $170 /	Save: $46 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

□ **.NET Developer's Journal**

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

□ **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 + FREE $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 /	Save: $2

□ **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
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

□ **ColdFusion Developer's Journal**

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

□ **WebSphere Journal**

U.S. - Two Years (24) Cover: $216	You Pay: $129.00 /	Save: $87 + FREE $198 CD
U.S. - One Year (12) Cover: $108	You Pay: $89.99 /	Save: $18
Can/Mex - Two Years (24) $240	You Pay: $159.99 /	Save: $80 + FREE $198 CD
Can/Mex - One Year (12) $120	You Pay: $99.99 /	Save: $20
Int'l - Two Years (24) $264	You Pay: $189.00 /	Save: $75
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
Can/Mex - Two Years (24) $360	You Pay: $179.99 /	Save: $180 + FREE $198 CD
Can/Mex - One Year (12) $180	You Pay: $169 /	Save: $11
Int'l - Two Years (24) $360	You Pay: $189.99 /	Save: $170 + FREE $198 CD
Int'l - One Year (12) $180	You Pay: $179 /	Save: $1

□ **WLDJ**

U.S. - Four Years (24) Cover: $240	You Pay: $99.99 /	Save: $140 + FREE $198 CD
U.S. - Two Year (12) Cover: $120	You Pay: $49.99 /	Save: $70
Can/Mex - Four Years (24) $240	You Pay: $99.99 /	Save: $140 + FREE $198 CD
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

RECEIVE YOUR DIGITAL EDITION ACCESS CODE INSTANTLY WITH YOUR PAID SUBSCRIPTIONS

3-Pack

Pick any 3 of our magazines and save up to $210⁰⁰ Pay only $99 for a 1 year subscription plus a FREE CD

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

6-Pack

Pick any 6 of our magazines and save up to $340⁰⁰ Pay only $199 for a 1 year subscription plus 2 FREE CDs

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

9-Pack

Pick 9 of our magazines and save up to $270⁰⁰ Pay only $399 for a 1 year subscription plus 3 FREE CDs

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

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

SYS-CON MEDIA

HOME
ENTERPRISE SOLUTIONS
CONTENT MANAGEMENT
DATA MANAGEMENT
XML LABS

WRITTEN BY **DEEPAK VOHRA**

XML Schema Binding with XMLBeans

Parse an XML document and generate an XML document

XMLBeans is an open source XML-Java binding tool used to generate Java classes and interfaces from an XML Schema. The generated Java classes may be used to parse or generate an XML document that conforms to the Schema. Some of the advantages of XMLBeans over JAXB are the ability to parse an XML document and support for all of the XML Schema constructs; the JAXB-generated classes do not have a parse method to parse an XML document, and JAXB does not support all of the Schema constructs.

XMLBeans provides the XMLObject, XMLCursor, and SchemaType APIs to parse or construct an XML document with the Java classes and interfaces generated from a schema. In this tutorial, Java classes and interfaces are generated from an example schema. They are used to parse and construct an XML document. The example schema, catalog.xsd, is illustrated in Listing 1.

The example XML document catalog.xml is parsed with the Java classes and interfaces generated with the XMLBeans compiler (catalog.xml is listed in Listing 2). The XMLBeans classes are required to generate Java classes and interfaces from an XML Schema. Obtain the XMLBeans Version 1.03 Binary and Development Kit. Extract the xmlbeans-current.zip to an installation directory. Add <XMLBeans\xmlbeans-1.0.3\bin to the PATH variable. Make <XMLBeans >\xmlbeans-1.0.3\lib\xbean.jar available on the Classpath. <XMLBeans> is the directory in which XMLBeans is installed.

XML Schema Compilation

An XML schema is compiled with the XMLBeans scomp compiler. The scomp parameters and options are listed in Listing 3.

In this section, the example schema catalog.xsd is compiled into Java classes and interfaces with the scomp compiler.

```
>scomp -src java catalog.xsd
```

The –src option specifies the directory in which the Java classes and interfaces get generated. The scomp compiler might generate the compilation errors shown in Listing 4.

The scomp compiler is a shell script installed in the <XMLBeans>\bin directory that you can modify as needed. The scomp compiler generates a Java class and interface for each of the top-level elements. In the example schema, catalog.xsd, Catalog Document.java, JournalDocument.java, and ArticleDocument.java interfaces get generated in the noNamespace package. Classes CatalogDocument Impl.java, JournalDocumentImpl.java, and ArticleDocumentImpl.java get generated in the noNamespace.impl package. A scomp-compiler-generated interface consists of the getter and setter methods for each of the attributes and subelements of a schema top-level element. A compiler-generated interface also consists of public static final class Factory classes with methods to parse and create an XML document. The scomp compiler also generates a xmltypes.jar file. Add xmltypes.jar to the Classpath. In the following section the example XML document will be parsed with the Java classes and interfaces generated from the example schema.

> “Schema binding with JAXB does not support all of the schema constructs, and does not provide parse methods to parse an XML document”

XML Document Parsing

In this section an example XML document, catalog.xml, is parsed with the classes and interfaces generated with the scomp compiler. Parse catalog.xml with a parse method in the Factory class

AUTHOR BIO

Deepak Vohra is a Sun Certified Java 1.4 Programmer, a Web developer, and a Nubean developer.

of the CatalogDocument interface.

```
CatalogDocument
catalogDocument=CatalogDocument.Facto
ry.parse(file);
```

“file” is specified as a java.io.File object. Obtain a CatalogDocument.Catalog class object from the CatalogDocument object. The Catalog class object corresponds to the <catalog> element in catalog.xml.

```
CatalogDocument.Catalog
catalog=catalogDocument.getCatalog();
```

Get an array of type JournalDocument.Journal[] from the Catalog object. A Journal class object corresponds to a <journal> element in catalog.xml.

```
JournalDocument.Journal[]
journalArray=catalog.getJournalArray();
```

Iterate over the Journal[] array to output the “publisher” attribute of the <journal> elements.

```
for (int i = 0; i <
journalArray.length; i++)
        {
System.out.println("Journal: " + i);
            System.out.println(
                " publisher : " +
journalArray[i].getPublisher());}
```

Obtain the ArticleDocument.Article[] array, which corresponds to the <article> elements in a <journal> element.

```
ArticleDocument.Article[]
articleArray=journalArray[i].getArtic
leArray();
```

Iterate over the Article[] array and output the attributes and subelements for each of the <article> elements.

The output from parsing an XML document with the XMLBeans-generated classes is illustrated in Listing 5. The Java program, XMLBeansParser.java, used to parse catalog.xml is shown in Listing 6.

In the following section an XML document will be constructed from the Java classes and interfaces generated from the example schema with the XMLBeans compiler.

XML Document Construction

Here an example XML document (catalog.xml) is constructed with the Java classes generated with XMLBeans. Create an object of type CatalogDocument.

```
CatalogDocument catalogDoc =
CatalogDocument.Factory.newInstance();
```

Add a CatalogDocument.Catalog to the CatalogDocument object. A Catalog represents the <catalog> element in the example XML document that is generated.

```
CatalogDocument.Catalog catalog=cat
alogDoc.addNewCatalog();
```

Add a JournalDocument.Journal object, which represents a <journal> element, to the Catalog object.

```
JournalDocument.Journal
journal=catalog.addNewJournal();
```

Set the value for the publisher attribute of the journal element.

```
journal.setPublisher(“IBM
developerWorks”);
```

Add an ArticleDocument.Article object, which represents an <article> element, to the Journal object.

```
articleDocument.Article article=jour-
nal.addNewArticle();
```

Set the “level,” “date,” and “section” attributes of the <article> element.

```
article.setLevel(“Intermediate”);
article.setDate(“January-2004”);
article.setSection(“Java
Technology”);
```

Set the value for the “title” and “author” subelements of the “article” element.

```
Article.setTitle(“Service Oriented
Architecture Frameworks”);
article.setAuthor(“Naveen Balani”);
```

Add the other <article> elements to the XML document. The example XML document catalog.xml shown in Listing 2 is generated.

The Java program XMLBeansConstructor.java which was used to construct an example XML document is listed in Listing 7.

Conclusion

Schema binding with JAXB does not support all of the schema constructs, and does not provide parse methods to parse an XML document. XMLBeans’ schema-binding supports all of the schema constructs and provides parse methods to parse an XML document, an advantage in binding with the XMLBeans compiler.

Resources

- *XMLBeans*: http://xmlbeans.apache.org/

DVOHRA09@YAHOO.COM

LISTING 1 • Catalog.xsd

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
xmlns:xs="http://www.w3.org/2001/XMLSch
ema">    <xs:element name="catalog">
   <xs:complexType>
    <xs:sequence>
     <xs:element ref="journal"
minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
   </xs:complexType>
 </xs:element>
 <xs:element name="journal">
   <xs:complexType>
    <xs:sequence>
     <xs:element ref="article"
minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
      <xs:attribute name="publisher"
type="xs:string"/>
   </xs:complexType>
 </xs:element>
 <xs:element name="article">
  <xs:complexType>
   <xs:sequence>
    <xs:element name="title"
type="xs:string"/>
    <xs:element name="author"
type="xs:string"/>
   </xs:sequence>
   <xs:attribute name="level"
type="xs:string"/>
   <xs:attribute name="date"
type="xs:string"/>
   <xs:attribute name="section"
type="xs:string"/>
  </xs:complexType>
 </xs:element>
</xs:schema>
```

LISTING 2 • Catalog.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<catalog>
  <journal  publisher="IBM
developerWorks">
    <article level="Intermediate"
date="January-2004" section="Java
Technology">
        <title>Service Oriented
Architecture Frameworks</title>
        <author>Naveen Balani</author>
</article>
```

```
    <article level="Advanced" date="October-2003"
section="Java Technology">
       <title>Advance DAO Programming</title>
       <author>Sean Sullivan</author>    </article>
  <article level="Advanced" date="May-2002" sec-
tion="Java Technology">
      <title>Best Practices in EJB Exception
Handling</title>
      <author>Srikanth Shenoy </author>  </article>
</journal>
</catalog>
```

LISTING 3 • scomp

```
Compiles a schema into XML Bean classes and metada-
ta.
Usage: scomp [opts] [dirs]* [schema.xsd]*
[service.wsdl]* [config.xsdconfig]*
Options include:
     -cp [a;b;c] - classpath
     -d [dir] - target binary directory for .class
and .xsb files
     -src [dir] - target directory for generated
.java files
     -srconly - do not compile .java files or jar
the output.
     -out [result.jar] - the name of the output jar
     -dl - permit network downloads for imports and
includes (default is off)
     -noupa - do not enforce the unique particle
attribution rule
     -nopvr - do not enforce the particle valid
(restriction) rule
     -compiler - path to external java compiler
     -jar - path to jar utility
     -ms - initial memory for external java compiler
(default '8m')
     -mx - maximum memory for external java compiler
(default '256m')
     -debug - compile with debug symbols
     -quiet - print fewer informational messages
     -verbose - print more informational messages
     -license - prints license information
     -allowmdef "[ns] [ns] [ns]" - ignores multiple
defs in given namespaces
```

LISTING 4 • scomp compilation errors

```
java.lang.NoClassDefFoundError: Beans\xmlbeans-
1/0/3\bin\//\build\ar\xbean/jar/;C:\XML
Exception in thread "main"

'javac' is required on the path.
java.io.IOException: CreateProcess: "C:\javac"
```

LISTING 5 • Parsed output from catalog.xml

```
Catalog has 1 journal elements
Journal: 0
 publisher : IBM developerWorks
Article: 0
Level : Intermediate
Date : January-2004
Section : Java Technology
Title : Service Oriented Architecture Frameworks
Author : Naveen Balani
Article: 1
Level : Advanced
Date : October-2003
Section : Java Technology
Title : Advance DAO Programming
Author : Sean Sullivan
Article: 2
Level : Advanced
Date : May-2002
Section : Java Technology
Title : Best Practices in EJB Exception Handling
Author : Srikanth Shenoy
```

LISTING 6 • XMLBeansParser.java

```
package noNamespace;
import java.io.File;


public class XMLBeansParser
{
    public  void printElements(File file)
    {
      try{
        CatalogDocument catalogDocument =
            CatalogDocument.Factory.parse(file);

          CatalogDocument.Catalog
catalog=catalogDocument.getCatalog();

          JournalDocument.Journal[]
journalArray=catalog.getJournalArray();


        System.out.println("Catalog has " +
journalArray.length + " journal elements");



     for (int i = 0; i < journalArray.length; i++)
        {
            System.out.println("Journal: " + i);
            System.out.println(
                " publisher : " +
journalArray[i].getPublisher());
            ArticleDocument.Article[]
articleArray=journalArray[i].getArticleArray();

            for (int j = 0; j <
articleArray.length; j++)
        {
            System.out.println("Article: " + j);
            System.out.println(
                "Level : " +
articleArray[j].getLevel());
            System.out.println(
                "Date : " +
articleArray[j].getDate());
             System.out.println(
                "Section : " +
articleArray[j].getSection());
             System.out.println(
                "Title : " +
articleArray[j].getTitle());
             System.out.println(
                "Author : " +
articleArray[j].getAuthor());


        }

        }
}catch(org.apache.xmlbeans.XmlException e){}
 catch(java.io.IOException e){}

    }

public static void main(String[] argv){

XMLBeansParser parser=new XMLBeansParser();
parser.printElements(new
File("c:/XMLBeans/catalog.xml"));


  }
}
```

LISTING 7 • XMLBeanConstructor.java

```
package noNamespace;
public class XMLBeansConstructor{

public CatalogDocument createCatalog()
{
     CatalogDocument catalogDoc =

CatalogDocument.Factory.newInstance();
     CatalogDocument.Catalog

catalog=catalogDoc.addNewCatalog();
   JournalDocument.Journal

journal=catalog.addNewJournal();

journal.setPublisher("IBM developerWorks");
ArticleDocument.Article

article=journal.addNewArticle();
article.setTitle("Service Oriented Architecture

Frameworks");
    article.setAuthor("Naveen Balani");
article.setLevel("Intermediate");
article.setDate("January-2004");
article.setSection("Java Technology");

 article=journal.addNewArticle();
article.setTitle("Advance DAO Programming");
    article.setAuthor("Sean Sullivan");
article.setLevel("Advanced");
article.setDate("October-2003");
article.setSection("Java Technology");

article=journal.addNewArticle();
article.setTitle("Best Practices in EJB Exception

Handling");
   article.setAuthor("Srikanth Shenoy");
article.setLevel("Advanced");
article.setDate("May-2002");
article.setSection("Java Technology");

System.out.println(catalogDoc);
    return catalogDoc;

}


public static void main(String[] argv){

XMLBeansConstructor xmlDocument=new

XMLBeansConstructor();
xmlDocument.createCatalog();


}
}
```

XML JOURNAL
PRESENTS
XML
Resource CD
Secrets of the XML Masters
$198 VALUE
Inside:
SPECIAL
Every Issue of XML-J Ever Published on One CD!

WRITTEN BY **JEFF DEXTER**

Combining XQuery and Web Services

Combined entities can lead to sophisticated solutions

The XML world is driving new emerging technologies that can be utilized to provide solutions to a variety of problems. This article focuses on two of these technologies: Web services and XQuery. As separate entities, these technologies provide a powerful set of features; but when combined they have the potential to present ever more sophisticated feature sets designed for very specific goals.

Web services, while not strictly an XML technology, depends heavily on XML for both its definition language, Web Services Description Language (WSDL), and its messaging protocol, Simple Object Access Protocol (SOAP). Alongside its various descendants, WSDL itself builds on previous XML technologies, such as XML Schema, to provide an abstraction layer over which service endpoints can be defined in terms of messages and operations. Similarly, SOAP provides a mechanism for issuing remote procedural calls and document transactions between distributed systems.

Correspondingly, XQuery provides a rich, data-oriented language for interacting with XML and XML-mapped data sources. It builds on existing XML technologies such as XPath and XML Schema to provide a unified query language over both weakly typed (streams, Web sites) and strongly typed (databases, messages) XML sources. Unlike other query languages, such as SQL, it is intended to not only be a database query language, but a general language for natively interacting with XML as well.

Both technologies have starring roles in new methodologies for solving old problems. Service-oriented architecture (SOA) promises to open up all of the applications in the enterprise via Web services. With it, application developers can build new applications from the functions and features of existing applications. Enterprise Information Integration (EII), utilizing XQuery, makes a similar promise: to present a single front end to all of the databases and data sources available on the network. In both cases, the goal is to bring together the processes and data we interact with individually in a manner that reduces the amount of custom mapping required for each source.

Our ultimate goal herein will be to demonstrate how Web services and XQuery combined can ease the task of developing applications on top of heterogeneous distributed systems. Essentially, we're looking to these technologies to hide all of the differences between the systems with which we wish to interact and allow developers to concentrate on the applications being developed.

Integrating XQuery with Web Services

Whether you look at adding XQuery to Web services or Web services to XQuery, the respective technology will gain something it did not previously have. On one hand XQuery, a language intended to manage, restrict, and combine static data from multiple sources of data (relational databases, XML databases, static Web content), can simplify interactions with XML-based Web services. On the other hand, Web services open XQuery up to more sophisticated interactions with systems outside its local scope. By combining XQuery and Web services, XQuery receives a new interaction model with external sources and Web services along with a sophisticated language for handling its input and output.

To illustrate just one possibility offered by the interaction, consider the following scenario. Super Travel Agency (STA) wishes

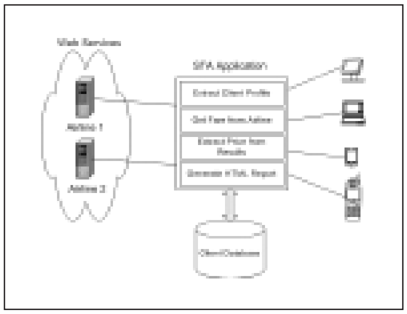


Figure 1 • Network architecture

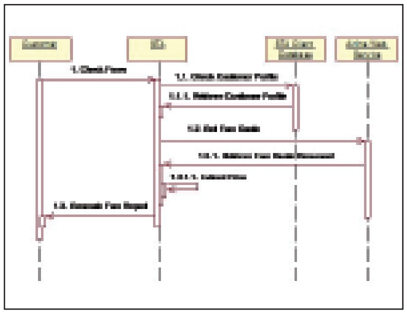


Figure 2 • Interaction sequence

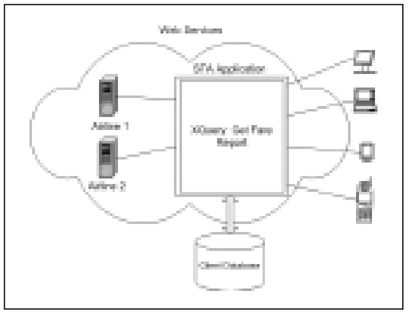


Figure 3 • Augmented integration architecture

Author Bios

Jeff Dexter is the lead architect of XML Query technologies at Raining Data Corporation, and their representative on the W3C's XML Query working group. Jeff has spent the last six years in the field of information aggregation and database query, specifically focusing on the realm of XML query at Raining Data. Jeff was the designer and main implementer of the TigerLogic 1.0 high-performance query engine, and the architect and principal engineer on the TigerLogic 2.0 data server.

to present a new feature on its Web site, awarding its repeat customers with fare estimates based not only on best fares but on their participation in frequent flyer programs. STA has a complete customer database with profiles listing their frequent flyer codes for each airline. For this system, STA needs to interact with the airlines via Web services that are defined using completely different result formats.

This fairly straightforward usecase can be accomplished in a discrete series of steps. First, the user profile is accessed and the relevant frequent flyer codes extracted. Combining these codes with destinations and dates, we then pass them to the appropriate airline services and receive a quote from each in a different XML format. We then have to define either a transformation or a custom extraction routine to extract the prices. Finally, we have to take this correlated data and produce a single report for consumption by the client.

In traditional modern programming languages, this problem could be solved by defining a number of discrete modules. One module would access and interpret the client database result, one would be defined for interacting with each airline Web service, one each for the extraction routines to parse the XML results and extract the relevant information, and finally a module for generating the final report. All of the interactions are defined in terms of XML messages, but given the disparate requirements of each such message format the solution will typically have to define a number of custom parsers to convert in and out of the data-oriented XML and object-oriented language paradigms. Simply put, this requires specialized coding and, while extensible, usually requires much more code to integrate new airlines and new client profiling data.

By allowing existing WSDL definitions to be referenced in the query and their services exposed as standard XQuery functions, the steps followed to achieve the solution can be more concise using XQuery. The simplicity is achieved by leaving the entire sequence of interactions in the XML world and taking advantage of XQuery's rich set of features for dealing with XML data to seamlessly interact with all of the various data and function points.

In order to facilitate this interaction between XQuery and Web services, we'll employ an extension to the XQuery language. This syntax parallels the built-in schema import facility in XQuery and was first proposed in XQuery at your Web service. We'll see below how it allows the query author to specify the services to be used in the query by simply importing the WSDL documents defining those services.

Bridging the Gap

In technology, gaps force us to build new technology to bridge disparate paradigms. XQuery sought to bridge the gap between modern programming languages and data. Web services sought to bridge the gap between those same programming languages and distributed processes. We've defined how to bridge the gap between XQuery and Web services; now in closing the gap between the posited airline scenario and the XQuery/Web services solution, we'll hopefully demonstrate how many of the gaps in existing solutions are now bridged.

Recall that the basic problem STA faced could be solved in a discrete series of steps. All steps essentially revolved around the disparate data requirements at each point in the interaction: the client, the database, and the airline services. We'll take each in turn and build up to a total solution. The inputs to our scenario are the client's identification ($client-id) and request parameters ($city and $date).

Accessing Client Records

Presuming the client records are located in a local XML database, we can access relevant client information using access to a simple collection.

```
for $client in collection
  ( 'clients' )//client
where $client/@client-id eq
  $client-id
return $client/frequent-flyer-
  program
```

This uses standard XQuery FLWOR syntax to extract the set of subscribed frequent flyer programs for the client of interest.

Interacting with Airline Services

Using syntax that parallels the schema import facility in XQuery, we define a series of service imports according to the airlines with which we wish to interact. In this example, we'll assume we're dealing with CheapAirlines (CA) and RidiculouslyExpensiveAir (REA). As such, we need to import the services provided by each.

```
import service namespace ca =
'http://cheapair.com/services.wsdl';
import service namespace rea =
'http://rea.com/fares.wsdl';
```

Further, we presume each provides a similar service for retrieving quotes, but each service returns totally different XML results.

Dispatching to the Appropriate Service

Just as each fare quote document follows a distinct but well-defined schema, we assume too that the frequent flyer codes follow distinct formats. These formats can be modeled as simple types in the XML Schema type system. Thus we can employ the XQuery typeswitch syntax to match codes to services.

```
 declare function local:get-fare-
price($frequent-flyer-program)
{
typeswitch(frequent-flyer-
program/@code)
case $code as rea:ffp-code
return rea:get-quote( $code, $city,
$date )/element(*,rea:price)
case $code as ca:program-code
return ca:return-quote( $code, $city,
$date )/element(*,ca:price)
default return ()
}
```

Note also how we use the price type from each quote to extract the price without having to concern ourselves with other extraneous data.

Generating a Report

Using the element construction in Listing 1, we can construct a complete report within our query, in this case in HTML.

Finally, bringing the total solution together, in Listing 2 we formulate the single query that will fulfill the use case.

Integration from Another Perspective

So far we've concentrated primarily on how Web services can be added to XQuery. This paradigm empowers XQuery to access a large number of new sources of data and functionality in a dynamic fashion. Nonetheless, looking at both the problem and the solution from the flip side may lead to even greater simplification and sophistication for applications wishing to leverage these two technologies.

XQuery at its basis is a language for interacting with XML and XML-mapped data models. This powerful paradigm can not only enable pulling data from sources, but with update extensions it can enable pushing to sources. Inherently, its ability to represent, aggregate, and multiplex data between a diverse set of data and process sources enables very sophisticated data flow modeling. Essentially, we're modeling very sophisticated services.

The flip side then is to look at XQuery + Web services as an enabling technology for defining more sophisticated aggregated services, and further services that can also seamlessly integrate with back-end databases and data sources.

An XQuery is essentially a function that has a discrete set of parameters, in the form of external variables, and produces a result document. An XQuery engine equipped with an appropriate transport layer (such as SOAP over HTTP) can expose such queries as value-add Web services that can coexist with other services in the architecture.

The benefits of using XQuery in this way parallel those already discussed. XQuery programs are inherently data driven, they provide an abstraction layer that hides the query/service author from the details of underlying databases and services, and they provide a rich set of services for interacting with existing data and composing new structures. In addition, it cannot be overstated that XQuery is a true cross-platform language with robust support from both the Java and .NET communities, and thus presents a cohesive paradigm for aggregating and augmenting assets from both worlds.

Conclusion

As technologies that arose as largely separate entities, XQuery and Web services have a surprisingly large amount to offer each other. Further, they blend together surprisingly well. That's not to say it's a match made in heaven. Integration between XQuery and some concepts from SOAP (sparse and multidimensional arrays) and MIME (binary data) underlying some Web services is incomplete and imperfect at best. Even some core WSDL concepts, such as multiple output parts, do not map cleanly to XQuery, though there are workaround methodologies.

Nonetheless, as we've seen, integration of the two can empower some fairly sophisticated solutions to a number of integration and aggregation scenarios. Whether we look at applications as data driven (the XQuery world), or process driven (the Web services world), real-world architectures such as EII and SOA must take into account that the applications are built on a combination of the two. The tighter the integration we can accomplish, and the more concise and efficient the representation of the solution, the more sophisticated the problems we are able to address.

The ultimate goal of emerging technologies such as XQuery and Web services, and architectures such as EII and SOA, is not merely to solve new problems, but to solve existing problems better. The only way to accomplish this is to diminish problems like data integration and process access from the solution space, and instead have engineers concentrate on directing the flow of this information. XML and XQuery provide a sufficiently flexible environment to provide data abstraction, while Web services and WSDL provide a similar level of process abstraction. Combined, applications can concern themselves less with mapping the problem into the technology space and concentrate more on mapping it into the solution space.

Acknowledgements

The author wishes to thank Ash Parikh, director of technology and development, EAG, Raining Data Corporation as well as John Bramley, VP New Applications, Raining Data Corporation; Gene Duquette, engineering manager, Raining Data Corporation; and Neil Rotstan, engineering lead, Raining Data Corporation for their contributions to this article.

References

- *XQuery at Your Web Service:* www.www2004 .org/proceedings/docs/ 1p603.pdf
- *XQuery 1.0: An XML Query Language:* www. w3.org/TR/xquery
- *Web Services Description Language (WSDL) 1.1:* www.w3.org/TR/wsdl

Note: Java and logos that contain Sun, Solaris, or Java, and certain other trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc., in the United States and other countries.

JEFFDEXTER@RAININGDATE.COM

LISTING 1

```
declare function local:report(
$quotes )
{
<html><body><table>
 <tr><th>Airline</th><th>Quote</
   th></tr>
{
  for $quote in $quotes
  order by $quote/@price
    ascending empty greatest
return
  <tr>
<td>{ $quote/airline }</td>
       <td>{ $quote/price }</td>
  </tr>
  }
  </table></body></table>

<<Listing 2>>
import service namespace ca =
'http://cheapair.com/services.
   wsdl';
import service namespace rea =
'http://rea.com/fares.wsdl';

declare variable $client-id as
     string external;
declare variable $city as
     string external;
declare variable $date as
     dateTime external;
declare function
local:report($quotes) external;
declare function local:get-
   fare-price($program) external;

local:report(
 for $client in collection(
'clients' )//client
for $program in $client/
   frequent-flyer-program
where $client/@client-id eq
   $client-id
return
 <quote>
{ $program/airline }
{ local:get-fare-price(
     $program/@code ) }
</quote>
)
```

Forum Systems Integrates with IBM Tivoli Access Manager to Simplify SOA Security

(Salt Lake City) – Forum Systems, a provider of trust management, threat protection, and information assurance Web services security solutions has completed integration of Forum XWall Web Services Firewall and Forum Sentry Web Services Security Gateway with IBM Tivoli Access Manager. IBM Tivoli Access Manager provides policy-driven security and access management for Fortune 1000 and government customers deploying service-oriented architectures (SOAs).

By integrating with IBM Tivoli Access Manager, Forum Systems' approach to managing, accelerating, and enforcing Web services security can now leverage centrally managed identities and access control rules to ensure policy consistency for XML-based transactions and communications.

Forum Systems' product portfolio is also fully integrated with IBM DB2 Universal Database, IBM WebSphere Application Server, IBM WebSphere MQ, and IBM Director. www.forumsys.com

GT Software and DreamFactory Partner To Provide Reuse of Legacy Applications in Service-Oriented Architectures

(Atlanta and San Francisco) – GT Software, a provider of mainframe integration solutions, and DreamFactory, a pioneer in providing rapid construction of rich client user interfaces for Web services, have announced a strategic partnership to enhance and extend legacy investments with newer technologies.

GT Software's Ivory Web Services offer an innovative solution for companies that want to move ahead with an SOA framework but still need to communicate with IBM legacy applications in order to provide enterprise-wide service routines. With Ivory, mainframe legacy applications can communicate seamlessly with open systems platforms, such as Java 2 Enterprise Edition, .NET and others.

DreamFactory's SBuilder is a rich-client for XML Web services. This technology empowers an exciting new class of sophisticated portals and Web applications built from native XML documents and driven by pure Web services transactions. DreamFactory offers standards-based integration with Java 2 Enterprise Edition and .Net Servers, enabling the seamless aggregation of multiple data sources on a single browser page without the installation of additional server software. This approach streamlines design, enhances user experience, reduces network traffic, and dramatically lowers development costs.

www.gtsoftware.com, www.dreamfactory.com

StrikeIron Continues to Gain Momentum with More Financial Services Offerings

(Research Triangle Park, NC) – StrikeIron Inc., developer of the StrikeIron Web Services Business Network (WSBizNet) that simplifies working with Web services, has announced the availability of four additional financial management Web services to its expanding network. With these additions, StrikeIron now provides comprehensive coverage of key real-time financial information for everything from broad econometric analysis, portfolio analysis, and in-depth stock, mutual fund, security, and company analysis.

The four Web services are available on-demand as a "pay-per-use" subscription. Users can also easily integrate this data directly into Excel spreadsheets with the recently launched StrikeIron OnDemand Web Services for Microsoft Excel beta release.

www.strikeiron.com

Parasoft BPEL Maestro 1.5 Simplifies Web Service Orchestration

(Monrovia, CA) – Parasoft has announced the release of Parasoft BPEL Maestro 1.5. Designed to help enterprises develop, manage, and orchestrate long-term, persistent Web service business processes. Parasoft BPEL Maestro 1.5 delivers a comprehensive BPEL engine and development toolkit that strictly adheres to Web services and BPEL standards.

Parasoft BPEL Maestro 1.5 is a 100% native BPEL integration platform. Companies will be able to deploy their Web service business applications faster and with few resources. Parasoft BPEL Maestro is available for Linux, Windows XP/2000, and Solaris.

www.parasoft.com

AmberPoint Completes Reseller Agreement with IBM for SOA Management Solutions

(Oakland, CA) – AmberPoint, provider of service-oriented architecture (SOA) management software, has completed a worldwide reseller agreement with IBM. Under the terms of the agreement, IBM Global Services will offer AmberPoint's management solutions as part of its recently launched SOA Management Practice.

Additionally, AmberPoint will deliver training and support for IBM's sales and consulting teams to ensure consistently excellent service from both IBM and AmberPoint. AmberPoint's solutions provide comprehensive management and security capabilities without requiring any changes to the services themselves.

www.ibm.com/services, www.amberpoint.com

WS-I Launches Advocates Program Broadens Community Support

(San Francisco) – The Web Services Interoperability Organization (WS-I) has introduced the WS-I Advocates Program, a newly launched recognition opportunity for companies that support the work of WS-I.

Although Advocates do not attend WS-I meetings or participate in the organization's technical working groups or committees, they do receive several benefits. Each Advocate's name, logo and testimonial are posted on WS-I's website. In turn, each Advocate may post a special "WS-I Advocate" logo on its Web site. Advocates will also receive a subscription to WS-Insider, WS-I's informative newsletter that is published several times each year.

The program is free, and any interested company may enroll by completing a registration form at www.wsi.org/join/AdvocateForm.aspx.

Cape Clear Delivers ESB with Native BPEL Technology

(Waltham, MA) – Cape Clear Software has Enterprise Service Bus (ESB): Cape Clear 6. This release unifies a complete Web services-based integration platform with full support for creating, deploying, and managing business processes using the Business Process Execution Language (BPEL4WS) 1.1 standard.

Cape Clear 6 will be available in January 2005 on IBM AIX, Linux, Microsoft Windows 2000/XP, and Sun Solaris.

www.capeclear.com/products