SYS-CON MEDIA

WebServices JOURNAL

.NET J2EE XML

Tools for Taming Web Services Management

pg 18

New technology moving to new maturity

Horses for Courses: Services, Objects, and Loose Coupling

pg 34

Integration without compromise

WS-I Basic Profile

pg 8

Not just another Web service specification

CONFERENCE PROGRAM

INSIDE

Page **39**

Why Web Services

Every year about this time, I like to take a moment to reflect on things. Some people make resolutions, but I've never been very good at that, so instead I try to take a look forward and see where the year is going to take us, and what we need to accomplish.

I visit many companies in a variety of industries during the course of a year. While there are significant technology differences from industry to industry, perhaps the most common themes for organizations today are integration and agility.

Notice that neither of these themes is innately technical in nature. Rather, they reflect high-level desires – and frustrations – for the ability of business processes to be as adaptable in software as they are in the business world.

Integration is one of the most elusive goals of any organization. At a high level it seems easy to imagine the company as one large knowledge and information repository that can be tapped to answer any question necessary to make business function properly. But dig deeper and you find that one large repository is actually hundreds of systems, applications, files, and databases, some of which were written decades ago and haven't been updated since because the impact on the business of changing these legacy black boxes has become too severe. These systems run the business, but the business can't change them, either because they're written in some obscure language that uses shorthand and meaningless variable names to implement business rules, or because corporate memory has been erased and no one really knows how the programs do their work anymore. There's no way to make these monoliths interoperate with other programs, no way to get to the mythical single system of record. And so, instead of a vast repository of knowledge we have islands of inefficiency.

And that's how we get to the desire for *agility*. Corporations need to be able to react to changing business climates. Be it governmental action such as HIPAA, Sarbanes-Oxley, or the PATRIOT Act, or market pressure from large customers such as Wal-Mart, business as usual means constant change. Businesses that can manage this changing environment survive and flourish. Businesses that cannot cope simply cease to exist.

The *Harvard Business Review* article "IT Doesn't

WRITTEN BY
SEAN RHODY

Matter" highlighted some of the problems with trying to use technology to gain operational advantage, but the reality of life is that IT does matter – it's part of every corporation. While IT may not provide a competitive edge (a claim I'd dispute myself) any longer, it's a requirement for any large business.

And that's why Web services are important. The basic components of Web services (XML, SOAP, WSDL, and UDDI) are the common plumbing that the industry has needed for over 40 years. I recently heard Web services described as the RCA jack of the IT industry – a simple, widespread standard allowing connectivity and modularity among software. And it's true. With the basics of Web services we can unite disparate systems quickly and effectively, without having to mortgage the company to pay the price of implementation.

Beyond the basics is where we may see companies addressing operational agility. The combination of Web services security, Web services management, and the standards for transactional integrity will allow the construction of long-running, secure, shared Web services. By utilizing standard connection mechanisms corporations can "wire" together their systems and finally achieve a measure of integration at a price that's cost effective.

But the real efficiencies await businesses that realize that true agility is seldom hardwired – for businesses to be truly agile it must be as simple to change a process in software as it is to issue a corporate policy directive.

And that's where Web services and business process management provide the needed technology. Web services, with abstract, business-oriented APIs reflecting business processes rather than technology processes, can be combined, redefined, and constantly adapted to changing business conditions. Business process management finally provides the tools and mechanisms for creating the agile organization.

And that's why Web services are important. @

■ **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

Hot New Features in SOAPscope 3.0

Graph View

When you're debugging a Web service performance problem and you have many messages in your SOAPscope database, you want to be able to easily discover the exceptional messages, or "outliers," that may be causing the problem. SOAPscope 3.0 can build a graphic visualization of your message log that allows you to quickly drill down to the messages that need closer inspection. Graph View renders data in the following modes:

• Average and maximum response time vs time.

• Average and maximum message size vs time.

• Successful transactions and faults in a specified time period

IDE Integration

We know where you live — in your IDE! So we've integrated SOAPscope right into Microsoft® Visual Studio® .NET. Writing a Web service? Analyze the WSDL at any time with the click of a button. Debugging a Web service? No need to write a bunch of small tests to call the different methods — just use SOAPscope's WSDL Invoke feature from within the IDE. Debugging a Web service client? Look at the message log to see what really got sent on the wire. Eclipse version coming soon.

Start and Stop from the Taskbar

SOAPscope 3.0 can now run as a Windows® service. Just click on the SOAPscope icon in the taskbar to start and stop SOAPscope.

Message Analysis

Do you have a Web service that you suspect is sending invalid messages? Check the message out with SOAPscope's Analyze feature. It will pinpoint the exact part of a message that is invalid and highlight it for you, providing you with a detailed, understandable description of each problem. The Analyze feature checks for:

• Conformance of request and response to SOAP specifications

• Conformance of request and response to the applicable WSDL file

• Proper formatting of request and response

The Easiest Way to Use the WS-I Testing Tools

While SOAPscope has its own message and WSDL analysis which includes a complete suite of tests based on the WS-I Basic Profile 1.0, we understand that some users may also need to run the actual WS-I Testing Tools. We've had users tell us that these standalone tools are not the easiest to configure and use, so we've radically simplified the process. Whether you're using the C# or Java version, with SOAPscope 3.0 it's just a push of a button. There are no XML files to modify. Simply select a WSDL or a set of one or more messages from SOAPscope's database, click a button, and we will generate a WS-I Testing Tools report for you to view, print, or save. And SOAPscope gives you instant feedback on the pass/fail status of the overall test run.

Advanced HTTP Support

SOAPscope 3.0 now helps you debug clients and services that make use of advanced HTTP features, including:

• SSL/TLS

• Basic Authentication

• Compression (gzip/deflate)

Protect your Investment
With New Subscription Service

Now that you have the best Web services diagnostic tool available, keep it up-to-date automatically! For just $99 we will send you all minor and major updates for a year.

"My purchase of SOAPscope has already more than paid for itself"
- Jim Albers

Introducing

SOAPscope 3.0

4 Ways to Know Your Web Services

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

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

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

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

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

Look What's New in 3.0

- Microsoft® Visual Studio® .NET Integration
- Graph Message Statistics
- Interactive Message Analysis
- Interoperability Testing System
- SSL Support
- HTTP Authentication Support
- HTTP Compression Support
- Support for multi-byte encoding
- Best usability of any Web Services tool

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

"Comparing SOAPscope and other SOAP sniffing tools is like the difference between shooting a bullet and throwing it."

Scott Hanselman
Chief Architect
Corillian Corp.

Mr. SOAPscope says -
"Try SOAPscope free at www.mindreef.com!"

Try It online at ✗**METHODS.net** or download a trial at Mindreef.com

"The Web Services Diagnostic Experts" Mindreef

WebAppCabaret™

J2EE Web Hosting

www.webappcabaret.com

Quality Web Hosting at a reasonable price...

<JAVA WEB HOSTING AND OUTSOURCING>

You have developed the coolest mission-critical application. Now you need to deploy it.
Outsource your hosting and infrastructure requirements with WebAppCabaret so you can save time and money and concentrate on other important things. WebAppCabaret is the leading JAVA J2EE Web Hosting Service Provider. From shared hosting to complex multi-dedicated server hosting, WebAppCabaret has the right solution for you. 30 Day Money Back Guarantee and SLA.WebAppCabaret offers the latest Standards based Servlet containers, EJB servers, and JVMs. We provide options such as e-Commerce, EJB 2.x, Failover, and Clustering. Our Tier 1 Data Center ensures the best in availability and performance.

At WebAppCabaret you have the flexibility to choose the LATEST hosting technology best suited for your
WEB application or your programming skill. If you are a programmer or consultant,WebAppCabaret has
the right hosting solution for your project or client's dynamic web application/services requirements.

OUTSOURCING:
Do you really need an IT department for your web applications, mail systems, and data backups when we can perform the same functions more efficiently at a fraction of the cost - with competent technical expertise and redundant hosting facilities.

J2EE HOSTING:
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

$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
900MB Disk
200MB DB
MySql
PostgreSql
Dedicated Apache
Telnet . SSH . FTP
5 Domains
100 Emails
Web Mail . POP . IMAP
more...

$191/mo Dedicated

Managed Dedicated
Server starts at $191
per month for:
256MB RAM
Pentium 4
40GB RAID
Firewall
Unlimited Domains
Unlimited Web Site Hosting
NGASI Control Panel
with your own Logo
Each dedicated
server configured
for standalone
web application
and web hosting (resellers)
at no extra charge.
more...

$17/mo Professional

Latest Tomcat/Jetty
Latest JSP/Servlets
Private JVM
Choice of latest JDKs
NGASI Control Panel
PHP and Perl
Web Stats
200MB Disk
30MB DB
MySql
Telnet . SSH . FTP
2 Domains
20 Emails
Web Mail . POP . IMAP
more...

$99/mo Reseller

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

5 Professional Plans
10 Basic Plans
NGASI Control Panel
with your own Logo

more...

Predictions, Predictions

Where are Web services going?

■ *Web Services Journal recently asked its editorial advisory board for their answers on several questions about where Web services is going next year. The respondents include Paul Lipton, director and technology strategist, Computer Associates International, Inc.; Andy Astor, vice president for enterprise Web services, webMethods; Anne Thomas Manes, research director, Burton Group; Simeon Simeonov, principal, Polaris Venture Partners; Dave Chappell, vice president and chief technology evangelist, Sonic Software; and Bernhard Borges, senior technologist, IBM Business Consulting Services, EAI practice.*

What Is Still Missing and Likely to Stay That Way in 2004?

Lipton: The larger challenges beyond Web service orchestration, such as business-level taxonomies and higher-level choreographies, will not be easily resolved in just one year. There still will not be widespread use of protocols and standards that function at higher business levels. We are still building the complex, distributed infrastructure that will be needed first. The higher-level semantics will start to come to fruition in 2005 and 2006.

Astor: Almost everything! Maybe we'll solve security next year, but I don't expect anything else to be truly resolved by 12/31/04.

Manes: For next year, I'm just looking forward to widespread deployment of SOAP and WSDL – let's walk before we run.

Simeonov: Customers still won't see the business case for significant SOA (service-oriented architecture) adoption. Problems will continue with SOA deployment and operations due to lack of standards and poor tools and management. Interoperable enterprise-grade Web services won't get much traction.

Chappell: What will still be missing is for an organization to be able to rely purely on Web services standards and interfaces to solve the problem of integrating on a global basis across departments and business units, and between partners. There's no big switch on the wall that someone can pull that will suddenly enable all applications within an organization to speak the same style of interface. IT is too busy just trying to "get stuff done" to worry about wholesale rip-and-replace of communications infrastructure. We will have to continue to live with the fact that disparate systems abound, each with its own means of connectivity

and interfacing technologies. Technology leaders will need to innovate by bringing endpoint abstractions to the apps in ways that the apps can deal with it on their own terms, yet still provide a consistent manageable view to the "integration archtitect."

Borges: We will still lack semantic standards and security.

What Issues Will Be Resolved in 2004, and How?

Lipton: Web services management will make major strides in 2004 with the completion of the OASIS WSDM (Web Services Distributed Management) specification and a broader understanding in IT shops that Web services truly need management, especially in a B2B setting. As Web services move from inside the corporate firewall to higher-value, mission-critical Web services, expect to see Web services management solutions that are fully SOAP aware from the major management vendors like CA.

Astor: See my earlier comments.

Manes: We've come a long way in 2003, so I expect that we'll achieve similar headway in 2004. We'll see broader adherence to the WS-I Basic Profile and use of Document/Literal services. We'll see widespread adoption of standards for advanced operational semantics, enabling security and reliability. I expect that we'll finalize the specs to support coordinated transactions, although adoption won't happen until 2005. I also think we'll make good headway with regards to policy definition, representation, and enforcement. Big issues that won't get resolved in 2004 include BPEL complexity, federated identity and trust relationships, and semantic integration.

WS-I Basic Profile

Not just another Web service specification

On August 12, 2003, the Web Services Interoperability Organization (WS-I), released the Final Material version of the WS-I Basic Profile 1.0 specification. This publication represents an important milestone for WS-I and the Web services community as a whole. It specifies the standards and technologies required for interoperability between Web services implementations running on different software and operating system platforms.

The Promise of Interoperability

WRITTEN BY
CHRISTOPHER FERRIS

The promise of interoperability is possibly the most important aspect of Web services technologies. That promise stems from the fact that Web services has its foundations in XML, which itself is interoperable across all platforms and programming languages. However, because Web services leverages heavily on the extensible nature of XML, the interoperability aspect of Web services is significantly challenged.

While most, if not all, vendors provide support for the established Web services standards, they are still motivated to provide added value to their customers in the form of advanced feature support for things such as security, reliability, transactions, and business process orchestration. Because many of the advanced Web services features are still in the early stages of development and adoption, developers and IT managers need more than just a checklist of (emerging) standards when making project implementation or product purchasing decisions. They need help in being able to determine when they are "coloring outside the lines" so that they can weigh the merits of incorporating these advanced features against the importance of ensuring broad interoperability of the deployed solution.

WS-I was founded with a mission to provide users of Web services technology with the guidance and tools that help them better understand where the boundary lies between the interoperable and not-necessarily-interoperable solution spaces so that they can make well-informed decisions.

About WS-I

The Web Services Interoperability Organization is an open industry effort chartered to promote Web services interoperability across platforms, applications, and programming languages. The organization brings together a diverse community of Web services leaders to respond to customer needs by providing guidance, recommended practices, and supporting resources, such as testing tools and sample applications, that enable the development of interoperable Web services.

WS-I Deliverables

The Basic Profile 1.0 is the first of a set of deliverables being produced by WS-I related to the Basic Profile. When complete, the package of deliverables produced in conjunction with all WS-I Profiles will be as follows:

- ***Use cases and usage scenarios:*** Use cases and usage scenarios capture (respectively) business and technical requirements for the use of Web services. These requirements reflect the classes of real-world requirements supporting Web services solutions, and provide a framework to demonstrate the guidelines described in WS-I Profiles.
- ***Profiles:*** A set of named Web services specifications at specific revision levels, together with a set of implementation and interoperability guidelines recommending how the specifications may be used to develop interoperable Web services.
- ***Sample applications:*** Demonstrate the implementation of applications that are built from Web services usage scenarios and use cases, and that conform to a given set of profiles. Implementations of the same sample application on multiple platforms, languages, and development tools demonstrate interoperability in action, and pro-

vide readily usable resources for the Web services practitioner.

• ***Testing tools:*** Used to monitor and analyze interactions with a Web service to determine whether or not the Web service instance or its artifacts (such as messages, WSDL, and UDDI registration components) conform to WS-I Profile guidelines.

At the time of this writing, each of the WS-I deliverables related to the Basic Profile 1.0 has been either formally approved as Final Material, or has been made public in the form of a Working Group Approval Draft.

Philosophy of the Profile

The WS-I Basic Profile was developed by the Basic Profile Working Group with a set of guiding principles that have been outlined in the Profile. These guiding principles form the "philosophy of the Profile."

Possibly the most important of these guiding principles is that there can be no guarantee of interoperability. The best that we could hope to achieve would be to improve the potential for interoperability since we were only dealing with the very basics of Web services technologies and we did not intend to address application-level semantics.

Another key guiding principle is that the Profile never relaxes requirements of an underlying specification. That is to say that the Profile never changes a MUST to a SHOULD. However, the Profile often seeks to improve interoperability by reducing the optional features of an underlying specification by changing SHOULDs and SHOULD NOTs to MUSTs and MUST NOTs.

The Profile also focuses on interoperability, not functionality. While the underlying specifications may contain design flaws and inconsistencies, the Profile focuses only on those that directly affect interoperability. WS-I leaves the work of addressing any inadequacies of a specification to the standards body that is assigned stewardship of the standard.

Scope of the Profile

Each Profile has a scope that is defined by the set of referenced specifications. A Profile attempts to improve interoperability within its own scope by placing constraints on optional features of the referenced specifications, clarifications of ambiguities in the referenced specifications, and guidelines for use of the

referenced specifications. A Profile does not impose constraints on that which is out of the scope of the Profile.

A key aspect of Web services is the composable nature of the specifications. WS-I Profiles are also intended to exhibit this same composable nature. They do so by defining the set of extensibility points, the extension mechanisms and parameters defined in the underlying specifications that may require out-of-band negotiation and/or agreement explicitly outside the scope of a Profile. While their use may impair interoperability, it is not subject to claims of conformance.

A Profile may place constraints on the use of extensibility points without constraining their range, so that specific uses of extensibility points may be further constrained by other Profiles to improve their interoperability when used in conjunction with the Profile.

The WS-I Basic Profile specification defines conformance of a Web service instance and its artifacts such as the messages it sends, its WSDL description and UDDI registration. The profile consists of the following set of nonproprietary Web services specifications:

• SOAP 1.1
• WSDL 1.1
• UDDI 2.0
• XML 1.0 (Second Edition)
• XML Schema Part 1: Structures
• XML Schema Part 2: Datatypes
• RFC2246: The Transport Layer Security Protocol version 1.0
• RFC2459: Internet X.509 Public Key Infrastructure Certificate and CRL Profile
• RFC2616: HyperText Transfer Protocol 1.1
• RFC2818: HTTP over TLS
• RFC2965: HTTP State Management Mechanism
• The Secure Sockets Layer Protocol version 3.0
  The Profile adds constraints and clarifica-

tions to those base specifications with the intent to promote interoperability. Where the Profile is silent (i.e., imposes no clarification or constraint), the base specifications are normative. If the Profile prescribes a requirement in the form of a clarification or constraint, the Profile supersedes the underlying base specification. Some of the constraints imposed by the Profile are intended to restrict, or require, optional behavior and functionality so as to reduce the potential for interoperability problems resulting from impedance mismatch between implementations that have made different choices with regard to implementation of the optional functionality. Other Profile

> " The WS-I Basic Profile 1.0 is, of course, just the tip of the iceberg "

requirements are intended to clarify language in the base specifications that have been the source of frequent misinterpretation, resulting in interoperability problems. Where possible, the Basic Profile WG has tried to ensure that the Profile clarifications are aligned with the thinking and direction of the Working Group responsible for the stewardship of the underlying specification to which the clarification applies. For example, clarifications to the SOAP1.1 specification were often aligned with issue resolutions made by the W3C XML Protocol WG responsible for the development of the SOAP1.2 specification.

Profile Highlights

The following list highlights some of the key constraints imposed by the Profile:
• Precludes the use of SOAP encoding
• Requires the use of HTTP binding for SOAP
• Requires the use of HTTP 500 status response for SOAP Fault messages
• Requires the use of HTTP POST method
• Requires the use of WSDL1.1 to describe the interface of a Web service
• Requires the use of RPC-literal or document-literal forms of WSDL

- Precludes the use of RPC-encoded–style WSDL
- Precludes the use of solicit-response and notification style operations
- Requires the use of WSDL SOAP binding extension with HTTP as the required transport
- Requires the use of WSDL1.1 descriptions for UDDI tModel elements representing a Web service

What's Relevant to the Developer?

The WS-I Basic Profile 1.0 specification is a rather complex document. A majority of the specification is targeted at the audience of runtime platform and development tool vendors working on vendor-specific implementations of SOAP processors, WSDL parsers, code generators, and the like. You could reasonably consider the Profile to be a concerted effort by those tools and platform vendors to ensure that their respective products will either generate or host interoperable Web services instances.

However, it isn't enough that each of the major vendors adopt the Profile for their product offerings since each will likely retain support for certain features that the Profile does not sanction (such as RPC-encoded Web services) and most will offer support for features that are outside the scope of the Profile. A Web services developer or IT manager should be familiar with all of the profile specification's contents. However, certain sections of the Profile are specifically relevant to the implementation of interoperable Web services.

The following lists each substantive section of the profile specification and its relevance to a Web service practitioner.
- **Section 4:** Relates to SOAP and the use of HTTP binding for SOAP. As such, it is mostly of interest to those developers writing SOAP processor implementations rather than Web services developers.
- **Section 5:** Pertains to conformant use of WSDL, and as such should be of interest to Web services practitioners, especially those who handcraft their WSDL descriptions.
- **Section 6:** Pertains to Web service discovery using UDDI. This, too, should be of interest to Web services practitioners. It describes conformant approaches to registration and categorization of a Web service in a UDDI registry.
- **Section 7:** Relates to security of Web services using HTTP/S and should also be of interest to Web services practitioners who require

security for the Web services they develop.

Many of the Profile requirements are often accompanied by examples of SOAP messages or WSDL descriptions that demonstrate both conformant and nonconformant adherence to the constraints and clarifications provided. The requirements associated with examples are likely to be of specific interest to Web services practitioners. However, the other WS-I deliverables related to the Profile may be more appropriate and relevant to the IT manager and Web service developer.

Scenarios, Sample Applications, and Testing Tools

The WS-I Sample Applications Working Group has developed deliverables based on the Basic Profile that a Web services practitioner will find useful.
- A mock supply-chain sample application that demonstrates most of the key features of the WS-I Basic Profile
- A Usage Scenarios specification that defines the most common design patterns for Web services and maps those scenarios to the Profile requirements that apply

The sample application serves a dual purpose. For vendors, it provides a means by which they can demonstrate and test their product's support for the requirements set forth by the Profile. To date, 10 vendors have produced independently developed implementations of the sample application, typically based on their respective runtime platform and/or development tooling. Each vendor has provided the source of their implementation so that Web services developers can better understand what they need to do to develop their own interoperable Web services.

The Testing Tools Working Group has delivered approval drafts of their reference testing tools for each of the major runtime platforms (Java and C#). They have also translated the constraints and requirements defined in WS-I Basic Profile 1.0 into formal test assertions that are used to configure the WS-I Testing Tools.

Web services practitioners can use the published reference testing tools to test their Web service instances, WSDL descriptions, and UDDI registrations for conformance to the Profile's requirements. IT managers can use the reports produced by the WS-I Testing Tools as a means of determining whether the Web services their developers have developed

conform to the requirements of the Profile.

Future versions of the WS-I Testing Tools reports will be augmented to identify the extensibility points that are used in a Web service instance so that IT managers (and developers) can make informed decisions as to whether the solutions they develop and deploy meet the specific interoperability requirements of a given situation. If a Web service requires broad interoperability, such as might be the case with an Internet deployment of a service, they might wish to constrain the use of extensibility points to those covered by a WS-I Profile(s). Conversely, if a Web service is being deployed for use within an intranet, interoperability may not be considered as high a priority as the advanced features provided through the use of an extensibility point. IT managers can leverage the information provided by the testing tools to make an appropriate, well-informed decision based on the requirements of the given situation.

Looking Beyond WS-I Basic Profile 1.0

The WS-I Basic Profile 1.0 is, of course, just the tip of the iceberg. WS-I has already begun work on a number of follow-on profiles for Web services, including Attachments and Basic Security. Work will begin on future profiles, tackling some of the more advanced Web services features as the various specifications upon which they are based mature and stabilize and as the interoperability requirements associated with these advanced features are better understood by the community.

As WS-I releases these future profiles and their associated testing tools and sample applications deliverables, the Web services community benefits by reducing the tension induced by having to choose between the need for broad interoperability and the need for advanced functionality that is not yet broadly adopted.

References
- *WS-I:* http://ws-i.org
- *WS-I Basic Profile 1.0:* http://ws-i.org/Profiles/Basic/2003-08/BasicProfile-1.0a.htm
- *WS-I Testing Tools:* http://ws-i.org/implementation.aspx ⓔ

■ About the Author
Christopher Ferris is chair of the WS-I Basic Profile Working Group. He is a senior technical staff member in Emerging e-business Technology at IBM.
■ ■ ■ chrisfer@usibm.com

British American Tobacco

In the driver's seat with Web services–based dashboards

■ With brands sold in 180 markets around the world, British American Tobacco, the world's most international tobacco company, was looking for a way to access and analyze data to improve supply-chain performance. The IT department was charged with finding a new approach that would provide significant improvements over the traditional approach of gathering and storing data, transforming it into information, and generating reports. The complexity of report generation, married with the inability to easily connect disparate information sources, makes this a costly information technology problem.

C ritical supply-chain information at British American Tobacco is stored in applications from SAP and i2 Technologies. The problem that the company faced was that the data and information contained in the enterprise was difficult to get at, couldn't be easily created or administered, and was not updated in a timely fashion. Conventional wisdom suggested traditional business intelligence tools on top of a standard data warehouse were the only option to effectively access the data and run complex, technically administered reports. However, building a data warehouse would take months, and British American Tobacco wanted to demonstrate business results in a much shorter timescale.

With the dawn of Web services, there had to be an innovative new technology concern that was applying the principles of "distribution" to that of intelligence or data for large global corporations. British American Tobacco identified CXO Systems, a two-year-old startup that had applied the same principles of distribution to business intelligence that leading technology companies

WRITTEN BY
**KEVIN POULTER
& MICHAEL CARTER**

such as Cisco and Sun had applied to the network and computing before them.

Fueling BAT's Web Services Initiative

The technology standards movement brought on by Web services has enabled organizations to attain new levels of business visibility that will allow companies to more nimbly react to market and business changes, and more profitably serve their employees, customers, and shareholders.

Having recently embarked on a Web services initiative, which included Librados Enterprise Integration Component Server and Infravio's Web service management tool, IT managers at British American Tobacco were compelled by the idea of implementing a Web services–based dashboard that would extract and integrate information from operational and analytical systems, providing executives with direct access to the information they need to make important business decisions without new investments in data warehousing, business intelligence, or EAI tools.

Working with executive dashboard specialists CXO Systems as its partner in executing

the next step in the company's Web services strategy, the company created a pilot dashboard for a major part of British American Tobacco's European business that extracts information from legacy SAP systems and i2 supply planning applications to create interactive reports that show metrics on average lead time for product delivery, forecasting metrics, and other manufacturing capabilities that help to ensure demand can be met. The product, CXO System, consumes Web services and processes the data, which is displayed as a dashboard interface. Now the supply-chain team within that part of the business can view one portal screen with a series of tabs, each showing metrics for different groups.

Dashboards: Outside Looking In

The concept of dashboards is simple – they provide a straightforward, consistent way for decision makers to view up-to-date data from every enterprise system. Dashboards allow companies to view and monitor business processes in real time. Applied to the supply chain, dashboards allow workers to make critical business decisions based on forecasted market demand, product inventory, and the varying lead times within different global markets.

Web services are a natural solution for dashboards. They allow British American Tobacco to integrate the information from multiple sources, providing the company with a seamless view of supply and demand factors for a fraction of the cost of creating a data warehouse. Not only was creating a Web services–based dashboard less expensive than creating a data warehouse, but it took only eight weeks. And while most large enterprises have point solutions for each piece of production information (logistics, delivery, etc.) and a degree of integration across some applications, most enterprises still cannot look at supply and demand in real time, in one place, at one time, the way that parts of British American Tobacco now can.

Now it is possible to monitor the performance of the supply and demand chain from one place without relying on alerts from multiple systems or phone calls from the field. Additionally, each local portion of the supply chain can be given a customized view into the chain, showing exactly the information they need to manage their operation.

.NET and Java: Fork in the Road?

While the great debate about the supremacy of .NET and Java continues, the British American Tobacco implementation is further proof that the two technologies can coexist. CXO is a .NET-based application, but products based on J2EE were used in the project to enable integration with back-end business systems.

The Road Ahead

British American Tobacco plans to spread the dashboard concept across its business, not just in the supply chain arena but into other areas of the business too. With the core foundation for distributed intelligence in place, the company can now activate dashboards within weeks in any business segment for a fraction of the price and time of the traditional data warehouse–based method. Additionally, due to Web services, once the dashboards are active they can stream data in "right time," based on the users' needs.

> " While the great debate about the supremacy of .NET and Java continues, the British American Tobacco implementation is further proof that the two technologies can coexist "

Together, British American Tobacco and CXO Systems think they have found the "holy grail" of Web services – the ability to solve the age-old data problem in a seamless and cost-effective manner, by treating the enterprise and all the data stored in any application or file as the data warehouse. ⓔ

■ **About the Authors**

Kevin Poulter is the application technology manager at British American Tobacco and is charged with defining global strategy for application infrastructure.

■■■ kevin_poulter@bat.com

Prior to cofounding CXO Systems, Michael Carter founded and was president of Dashboard Systems, which recently merged with Beacon Hill Group, creating CXO Systems. He currently serves as the company's executive vice president of Marketing and Business Development.

■■■ mcarter@cxosystems.com

Simeonov: We should be able to get SOAP 1.2 tooling and finally get a solid level of interoperability

Chappell: The term "Web Services" will no longer be heralded as the next big trend to solve everything. IT will be focused on integration more than ever, and the thought leaders in the industry will focus on articulating the architecture for enabling a highly distributed integration network based on standards. A new technology category, which is already being coined as the "enterprise service bus" and adopted by many vendors and analysts, will become a battleground for mindshare.

Borges: I see improvements in environments – partial and incremental resolution of architecture, design, (improved) best practices; incremental resolution of BPEL implementations and management; and BPEIL engines – as the first tangible step toward business process as opposed to "mere" plumbing

What New Ideas Are Likely?

Lipton: There's bound to be an explosion, but I'm not going to hazard a guess here. That would spoil the fun!

Astor: The need for environments that support Web services from any environment, and that put the intelligence for this function into the network, rather than in the endpoints.

Manes: New ideas? I suspect we haven't thought of them yet.

Simeonov: There will be a focus on monitoring at the business level (real-time business intelligence leveraging SOA as opposed to the traditional BAM approaches). Maybe we shouldn't replace everything!

Chappell: It's very likely that new ideas around XML data storage, XML caching and aggregation, and intelligent routing of XML will emerge.

Borges: There will be a (continued) emphasis on SOA and process and on the service bus coupled with integrated semantic brokers

Will Business Process Take Off in 2004?

Lipton: Take off is a relative term. BPM (business process management) will certainly continue to grow. Specifications like WSBPEL and W3C Choreography will help that happen. As we move towards more complex composite Web services composed of myriad subordinate services, enterprises will find themselves compelled to use solutions that allow them to monitor and maintain service levels, quality of service, and more. The days of simple, unmanaged Web services are numbered for many applications.

Astor: Take off, yes. And by 12/31/04, I believe we'll know what the answer is going to be. But it won't be in place and finalized.

Manes: Many people already do business process – although today they typically use proprietary technologies. I expect that most vendors will convert their products over to BPEL in 2004. I just don't think we'll solve the complexity problem next year and I don't think BPEL will become a mainstream technology until we solve the complexity issue.

Simeonov: No, something like this doesn't "take off." The key impediment to this market is not the level of tooling but the state of business.

Chappell: End-to-end business process across departmental boundaries and business units will continue to be a challenge, and will continue to hinder its adoption. However, this will be top of mind for our readers, who will continually be looking for solutions.

Borges: Yes, but it will be limited by a lack of semantic standards and the inherent difficulty of (larger scale) ontology design ⓔ

Bulletproof Web Application Deployments

Best practices in testing

■ Much has happened to the World Wide Web since its start, with continuing and dramatic improvements that have created one of the most powerful information sharing and communications tools worldwide. During the past few years, Web applications and services have burst onto the scene, expanding on the Web's ability to deliver on its original promise of rich functionality, features, and integration. Today's most successful deployments integrate robust networking and application designs along with stringent testing in order to ensure solid, ongoing performance and reliability.

W eb application infrastructures, like any significant network deployment, can suffer from myriad issues and vulnerabilities. At best, these issues result in performance and scalability problems that annoy end users. But the ongoing drive to reap the benefits of a Web applications–enabled enterprise have driven the integration of Web infrastructure into core business functions, with performance implications and security vulnerabilities that, if left unresolved, can result in millions of dollars in lost productivity and serious liabilities from stolen personal and financial data along with confidential corporate information.

What must enterprises solve in order to deliver on the promise of Web applications? Web applications arguably share many of the same issues as any system deployment. At a minimum, the list usually includes:

- **Reliability:** The ability of the system to function properly under various conditions.
- **Availability:** The ability of the system to pro-

WRITTEN BY

PHILIP JOUNG

vide nonstop services over a particular service period, which for the Web is often "forever."

- **Scalability:** How effectively and how easily a system can adequately respond to changes in demand.
- **Security:** While security has always been a consideration for most system deployments, Web applications and services must often contend with a lack of built-in security while running in an environment that often leaves them continually exposed to security threats.
- **Performance:** The ability of the system to serve responses rapidly.
- **Interoperability:** The ability of the system to integrate and work with other connected systems and enterprises. This is indeed one of the advantages of Web applications and services.

Your environment may have different pressures from the list above, but will almost inevitably include cost and time – the bane, and indeed driving force, of most deployments. Given all these pressures, requirements, and

goals, what can help ensure success? In a word, testing. Testing provides valuable insight to understand how things are progressing during design and deployment and how well things will go once deployed. As with any network and application deployment, testing benefits from knowledge and expertise. This article provides a foundation towards robust testing and ultimately, a reliable, high-performing Web application deployment. As you read the tips presented here, remember to adapt and apply the ideas to your specific requirements and environment rather than adopting them blindly.

A doctor receives a call from his patient, who says, "A month has passed since I saw you and I'm still feeling sick."

The doctor replies, "Did you follow the directions on your bottle?"

"Yes, I did", said the patient, "It says 'Keep tightly closed'."

Critical Success Factors

All too often we sacrifice quality and proper deployment testing to meet development schedules based on time-to-market pressures. The following are the key ingredients toward avoiding that trap.

Have a Plan, Man!

You have an idea that might "have some legs," ask several colleagues to have a working meeting with you, and the idea develops into a proposal for a Web application.

Rigorous planning can make a dramatic difference in the success of a Web application deployment. By outlining the entire process from start to finish, participants can better understand their role in the process and propose improvements early. A plan provides clarity and a sense of direction to what can seem like a daunting process. Start by documenting the goals, answering important questions such as the features that the deployment will support and its intended users. Next, consider the success factors, including cost and deadlines along with specifics such as availability (e.g., 99.5% or 99.99%, 24X7 or weekdays from 8 am to 6 pm) and performance (e.g., sub 2-second response time). Incorporate the important project steps and phases into the plan, such as design, testing, development, deployment, final assessment, and maintenance. Finally, obtain agreement and sign-off on the plan from involved parties, including marketing, management, end users, engineering, and operations/information technology.

Test Early and Often

During early design of the Web application network, you consider two approaches for data exchange that could work. You prototype the ideas that afternoon and test them, discovering that your first idea not only performs better, but is simpler and easier to build.

Getting an early start to testing, which may seem to delay a project, will actually end up saving both time and money. Projects have particular costs associated with them; the simplified graph in Figure 1 depicts the cost life cycle for development and testing/ updates. As a project progresses from the design phase to deployment, development costs increase. After deployment, the project moves into the maintenance phase and development slows down. For testing and updates, the costs start very low – most testing during the design phase will involve comparing prototypes and testing ideas, with changes easily made. As the project proceeds, the complexity increases, requiring increased effort to test and implement changes spurred from testing. Early testing will help find issues and design flaws earlier, resolve performance problems, locate bottlenecks and failures, and ultimately deliver a final system that is more reliable and has fewer unexpected issues.

'Think Test' from the Onset

Everyone on your project team agrees performance is an important goal so that all meet early on to discuss ways to ensure excellent performance in the final web application.

Incorporate testing into your Web application right from the start by having its developers and architects integrate visibility and testability into their designs. Providing visibility into a Web application ensures testers can obtain details on its status during testing, while also enabling operations to monitor status in production. For example, if performance is a critical goal, the application could provide timings for certain operations (database insert took 350 ms, record search took 1249 ms) so that bottlenecks can be quickly identified. If reliability is a goal, the application could provide status codes to highlight important internal events. Of course, do not overlook the value of documentation for testability, which could provide important details such as cookie format, dynamic session IDs, state diagrams, etc.

Having the developers, architects, and

testers work together from the start not only fosters improved communications throughout the test, but also promotes a mindset of high quality early on.

Web Services Performance – Surviving to Tell the Tale

Part of your new system includes a Web services component to allow partners to access important data quickly and easily. Initial performance tests of this feature reveal performance issues as the number of simultaneous accesses increase. You work with IT to add another server, and meet performance targets.
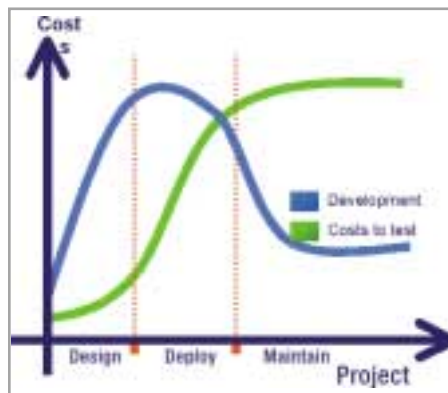
Many times, enterprises choose to migrate a legacy, proprietary service over to Web services in order to gain some valuable benefits. Often, this migration comes with an unexpected cost in performance when the new architecture fails to perform anywhere near the legacy application. Because Web services work with XML formatted data in clear text (typically less efficient than binary data), it ultimately increases time and resource utilization on both the client and server to parse and marshal the data. Incorporating SSL and WS-Security also reduces performance while increasing the amount of data that needs to be transmitted. Web services deployments should include both functional and performance testing to ensure that the delivered service meets expectations.

'Get Real' When Testing, a.k.a. 'Garbage In, Garbage Out'

You are now ready to move testing to the next stage, incorporating realism into the test by simulating several populations of users that use different browsers, connect at

different link speeds, and exercise your Web application in different ways.

While testing provides dramatic benefits, its value stems from proper test design and usage. Having a highly advanced and expensive testbed means little if used improperly, and can lead to incorrect conclusions that end up with disastrous results in production. Incorporate realism into your testing, which for Web applications means considering two aspects: user behaviors and network realism.

Quantifying users involves capturing their particular behaviors, which can include their Web application usage patterns, browser versions, page think times, and user abandonment. Networks themselves also have certain issues that can dramatically affect Web application performance – this becomes more and more likely with larger and more complex networks, with the Internet as a prime example. Issues faced by networks include end-to-end latencies, link speeds, packet loss, IP fragmentation, jitter, and bursty traffic patterns, all of which can adversely affect Web application performance and stability.

How much realism is needed? This question is not easily answered, but ultimately ends up being a judgment call on the tester's part, balancing the effort and expense of incorporating increased realism with the costs and criticality of failure in the Web application. However, prioritization will help, choosing the user and network behaviors that have the most dramatic effect on your Web application.

System Failures Are Fine, as Long as They're During Testing

During one early morning test run, the Web application fails, causing concern in one of the senior developers, who comes to talk to you. You point out that this is actually good news.

Try to "break" your Web applications during testing. This can mean tactics such as sending incorrect data, overrunning it with double its expected traffic (or more!), sending a large denial of service attack, trying to access the system with cookies turned off, etc. Every failure discovered during testing not only prevents it from appearing in production, but also helps answer many important questions:

- Was the failure expected? What was the performance near failure?
- What systems/functions failed? Which ones first? These answers point to candidates for current or future upgrades.

- What were the signs and system behaviors leading up to failure? This information can be included in system monitoring, helping to predict failures before they occur.
- Is recovery from failure quick? Easy? Reliable? Automatic?
- Did backup systems take over during failure? How did they do in performance and function?

Security Must Be Tested

You arrive on Monday to learn that a distributed denial of service attack effectively brought down your company's Web site that weekend and quickly realize that your predeployment Web application is currently just as vulnerable.

Network attacks continue to increase in potency as their creators learn to exploit the prevailing weaknesses of network devices and software. No single strategy can effectively mitigate security issues, but security testing plays an important role in validating and ensuring the overall security of a security infrastructure. Beyond helping to discover security and privacy issues in a Web application, security testing determines the performance impact of the security deployment – increasing security usually comes at the cost of reduced performance.

When testing for security, test both the inside and outside of the network perimeter – network intrusions and attacks can just as easily come from internal people or hackers that have successfully infiltrated other parts of the network. Also, closely examine network security while under network load – networks that seem secure without traffic can, under load, overload and send traffic to backup systems that do not have the same network protections.

Test Holistically

During testing, you systematically test each network component individually, starting with the Web server. You discover a serious performance issue and post these issues on a forum. A few hours later, the forum contains several suggestions; the second suggestion triples the performance, and the other suggestions provide another 20% increase.

A modern Web application infrastructure consists of many systems interconnected in a large network that simultaneously attempts to address performance, security, availability, reliability, and scalability. Holistic testing starts from the early deployment stages,

dividing each component of the infrastructure to test each one individually. This means firewalls, intrusion detection and prevention systems, caches, server load balancers, Web servers, application servers, databases, and file servers should all be candidates for testing. This helps to focus attention on each component, ensuring that each is tuned to properly address the needs of the overall Web application while also allowing poorly performing components to be replaced early.

After individual testing, connect the individual components and test again. This test locates overall system bottlenecks, discovers interoperability issues, and validates proper network functionality and performance.

At the Finish Line, Get a Baseline

You and your team have put in many hours to create a system that everyone is proud of. Before turning the Web application live onto the Internet, you conduct a final performance test and locate a performance issue tied to an application that was still running in debug mode.

Conduct a final test suite to get a baseline of performance of the new system. This baseline provides valuable information:
- Validates expected performance.
- Reduces the chance of surprises in production.
- Provides visibility into performance changes: System performance often changes with use. For example, a database with 500 records will perform better than when it grows to 500 million records.

To increase the effectiveness of baselining, consider "bracketing" your baselines just like photographers do when capturing that important shot. Because of film's limited exposure latitude, photographers take photos at the exposure that they expect will produce the best result and then take extra shots below and above the expected exposure, just in case. Baselining can incorporate this same idea, testing at various settings and levels to obtain performance and behavior under various conditions.

Finally, use baselines as the starting point for future upgrades to the Web application, comparing upgrades to the baselines' performance to ensure that an improvement has actually been realized.

Your Work Is Never Done

With your Web application in production,

you find that your assumptions on how it would be used were off-target on a few items, and tweak your tests to incorporate these differences.

Testing is a continual learning cycle. The successful deployment of a Web application is certainly one of the more important milestones in this cycle. Once in production, the Web application moves into maintenance while becoming one of the most valuable sources of information for future deployments and testing. It provides information that can often only be "guesstimated" beforehand, such as usage patterns and peak loads. The effectiveness of testing can be determined and adjustments made to improve future testing.

Monitoring usually plays a critical role in production, continually recording important system parameters, performance, and availability. Effective monitoring can often be the first indication of system failures, sometimes even being able to point out failures before they happen. Testing points out candidates for monitoring, but the live deployment will ultimately determine which monitoring candidates matter while adding new ones.

Summary

Today's Web applications contribute billions of dollars to our world economy, and there is no reason to believe this growth and innovation will not continue. The most successful deployments will deliver rich functionality and compelling features along with users that will expect top performance and reliability. Competitive pressures and concerns with cost only serve to augment the necessity of delivering superior applications and networks. Indeed, the demands are many, and there is no magic bullet, no easy way out. However, rigorous testing provides the foundation to answer these demands effectively, saving costs, improving quality, increasing security, and delivering high performance. Ⓔ

■ About the Author

Philip Joung is director of technical marketing at Spirent Communications. He has more than 15 years of industry experience, and focuses on helping the company develop solutions that solve many of the scalability and testing issues IT departments and technology vendors face as they develop and deploy new technologies. He is a frequent presenter on the importance of testing at industry conferences and technical seminars.

■■■ philip.joung@spirentcom.com

Tools for Taming Web Services Management

New technology moving to new maturity

■ As Web services move beyond opportunistic implementations and proof-of-concept deployments to support enterprise-wide services in mission-critical applications, the need for Web services management becomes ever more pressing. While Web services present a tremendous opportunity for organizations to improve coordination and integration both within the enterprise and with old and new business partners, dealing with a multitude of partners with changing requirements in an open heterogeneous environment also provides new opportunities for failure. These problems can be mitigated through the use of policies representing proper procedures and best practices. It is the role of Web services management to allow users to choose procedures, establish best practices, and then enforce them in real time.

I will discuss the touch points for applying these policies and some means of organizing them so your Web Services Management Platform (WSMP) doesn't become, itself, unmanageable. We use two organizational techniques for this: *service views*, a way of organizing Web service operations for different audiences; and *policy groups*, a way of organizing policy decisions applicable to multiple operations. First we will show how management concerns arise from an examination of Web service messages and then show how these two concepts provide an

WRITTEN BY

MATTHEW FUCHS

effective means of organizing management policies.

Web Services Management Areas of Concern

In choosing what to manage, there is an inevitable trade-off between processing that happens in the manager and processing that happens in the Web service. At one end is information that must be tracked globally for the whole network to function, or policies that must be consistently applied across many services. At the other is processing that is more application specific, but more easily

handled by the manager and can insulate the application from changes, such as transformations into and out of standard formats. Perhaps the best way to show the many ways management can impact message processing and provide stability is to observe the handling of a message or two by a Web Services Manager. Listing 1 presents a short Web service message – in particular, a Simple Object Access Protocol (SOAP) message. Looking at this message, we can ask a number of WSM-related questions:

- *Mitigating outside attacks:* Along with the message, we have the (alleged) IP address of the originator. How many times has this IP attempted to access this service? Have there been a slew of unsuccessful attempts indicating an attempt to break into this service or related ones? Should we block this IP? Should we shut down the service? Who do we inform?

- *Information security (encryption/signature):* The message contains a WS-Security header block indicating a signature for certain parts of the document. Who signed these elements? Where is the Certificate Authority (CA) for this signature? Is there critical information that hasn't been signed? Is there information that needs to be encrypted given its source or destination? If we need to sign/encrypt it, which key do we use?

- *Access control:* Who is this message from/ to? How do we identify them? Are they

how many people does it take to change the web?

macromedia®
COLDFUSION®
MX

See for yourself. Upgrading to ColdFusion MX gives you amazing new features and a powerful Java™ architecture. Incorporate XML and web services with ease. Build flexible and maintainable applications with ColdFusion Components. Integrate with J2EE™ and .NET to deliver rich user interfaces with native connectivity to Macromedia Flash.™ Easily migrate existing applications and save time with the powerful new Macromedia MX development tools. Try ColdFusion MX today and see what you can do now.

Download the free 30-day trial at www.macromedia.com/go/cfmxad
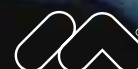
allowed to access the service they are contacting? Once we identify them, do we need to switch credentials to access the ultimate service? Do we need to filter information in the Web service's response? If they have no access rights, who do we inform?

- **Metering/resource management:** Is there a cost to access this service? What account should be used? How much do we charge?
- **Quality of service and service-level agreements:** Is the service up? What is its response time? Given two simultaneous requests, which is serviced first?
- **Quality assurance:** Is the message valid according to its schema? If I've transformed the message, is the new message valid? Do the values make sense? Is it the expected response?
- **Standards implementation and conformance:** Because SOAP itself allows so much variability, other groups, such as WS-I, have specified best practices for conforming SOAP messages. Is this a SOAP message? Is it SOAP1.1 or SOAP1.2? Is it WS-I Basic Profile compliant?

Because of the heterogeneity of the network and the variety of clients that need to access an ever-burgeoning list of Web services, the answers to these questions can vary considerably depending on who (requestor) is

menting a management capability on a Web service–by–Web service basis prohibitively expensive. Clearly the work required to build a management capability from scrap far overshadows the cost of implementing any single Web service.

Organizing Policies and Services

XML provides the possibility to directly examine and operate on the message stream, and many common policies require examining, if not actively manipulating, the message. Therefore, WSMPs tend to use an intermediary or broker architecture; messages between the requestor and the base service first pass through the manager. The manager sees a sequence of structured messages from a particular network port destined for another network port. While it is possible to design policies exclusively at the message level, this inevitably leads to diminishing returns. In a complex environment, the set of if-then-else conditions to first determine the (sender, message, channel, receiver) tuple and then answer our list of questions will be torturously complex. With no abstraction level above the message, all the decision logic must be specified explicitly, while a more abstract view would answer many of them implicitly. Additional layers up the stack include the operation level, the port level, the service

The key standard for organizing Web services is the Web Services Description Language (WSDL). WSDL organizes operations by endpoint and specifies the structures for messages to those endpoints. It has also become the focal point for specifying other information about services. A <definition> element contains a list of services, ports, bindings, and operations. Nevertheless, what may appear as a logical organization of operations from the developer's viewpoint may no longer be so when looked at from other perspectives. Different sets of users will be allowed access to different sets of operations with different access rights using different authentication mechanisms. Mixing all of these in a single WSDL is dangerous.

But there's no reason to see a WSDL <definition> as universally valid. We can treat a WSDL <definition> element as a protocol for a provider to communicate a permitted set of operations and policies to a specific set of users. Internally, we can link a given <definition> with a particular set of policies about messages, ports, and operations. We take a particular set of (sender, message, channel, receiver) decisions about some set of services and formulate those as a set of policies. From those policies we generate a <definition> element reflecting those policies, and send it to the particular requestors to whom we wish to apply those policies. The <definition> can therefore serve as a tool for management and a mechanism for conveying policies, with different sets of users receiving different WSDLs, even with different addresses, although the underlying operations may be the same.

We call each of these a *service view* by analogy to the database view of SQL. In many ways, this is an extension of the WSDL <binding>, which provides for multiple serialization formats for Web services messages, depending on the characteristics of the recipient. Another analogy is to interfaces in traditional programming languages, such as Java. In Java, a single object can support any number of interfaces, with each different interface including a subset of the methods of the class. A client object sees another object by the interface provided, without knowing about the underlying class and any other methods available.

For example, in Listings 2a and 2b we see abbreviated WSDL for two services – purchase orders and support – where there may be a variety of clients, or service requestors, some internal to the organization and some exter-

> " ... dealing with a multitude of partners with changing requirements in an open heterogeneous environment also provides new opportunities for failure "

sending what (message) over the network (channel) to whom (receiver/base service). This tuple (sender, message, channel, receiver) is the key to these different answers. Any set of answers represents a policy, a specification of how your system will react to a particular set of circumstances.

The sheer amount of work required to support the entirety of this agenda makes implement-

level, and even, in the other direction, the XML element level. In addition to messages, there are the requestors to consider as well. *Service views* and *policy groups* are two techniques to organize these various levels according to the changing requestor/message/channel/operation mix so they can be managed in concert, rather than in conflict, in a way that simplifies rather than complicates.

nal. Both services, though, are designed for internal customers.

In Listings 3a and 3b we can see two different service views of these operations. In each case we've organized a subset of the operations into a single port with a new address, combining operations from both underlying services. Immediately we've divided our users into two groups that will access different mixes of our underlying base services through different routes. In Listing 3a, the "Acme-SpecialService", authentication is specified by adding username and password fields. This is

authentication – to be unified in a single view. While the number of different combinations of these is astronomic, in reality there will be very few. With a service view, the actual combinations can be brought together and easily managed by a set of consistent policies. The original service, now hidden behind and protected by the view, remains the same. Other policies not directly describable by the WSDL can still be segregated by view. For example, the authentication mechanism used can be assigned on a per-view basis. Then policies regarding

ing other organizations. The Web service in Listing 3b allows customers at Zenith access to the ListOpenIssues operation. A logical policy would be to restrict Zenith to viewing only its own issues, but the underlying operation may not be able to do so. A policy employing a filter could implement this for the Zenith view, but a more general policy would be to apply this restriction to any organization with access to the underlying operation. This can be implemented by a filter to stop the message, or a transformation to excise the information. The policy would apply across many operations, but the identity of the requesting organization comes from the view. As different operations have different structure, the filter or transformation would need to be attached to the underlying operation. The policy would be applied across all operations, using either a service view–supplied organization name and the operation-specific filter or information from the incoming message. Were these not present, the policy would fail.

At present there is no uniform way to describe the variety of policies that may be applied to Web services. WS-Policy, a set of proposals related to WS-Security, may provide a means for naming the policies being applied to a particular service view; however, policy implementation will undoubtedly remain proprietary to different Web service management products. The degree to which they are automated will also vary from WSMP to WSMP.

> ## "The field of Web services management is still in the early stages, but already it has shown itself to be a complex area requiring new techniques beyond previous management tools"

accomplished differently for the RPC- style SubmitIssue from the document-style SendPurchaseOrder. In the first case, extra parts can be inserted directly into the message. In the second, because everything needs to tie back to a W3C schema, we create a new namespace to house the extra elements and then add them to the message. In Listing 3b, the clearly more sophisticated Zenith-SpecialService, we've added features. For example, we've added a WS-Security header to the mix, specifying the use of WS-Security as the access mechanism. This also requires importing the WS-Security schema. We've added a WS-I Basic Profile (BP) conformance claim and associated header. But because BP doesn't support RPC encoding, a document-style equivalent needs to be generated (including a new schema corresponding to the namespace). These indicate different levels of integration, depending on the sophistication of the clients.

In essence, the service view allows a constellation of concerns – service requestors, operations, formatting, authorization, and

requestors from a particular location – such as Acme – need only be tested for messages coming to specified ports – as only requestors from the targeted group will be authenticated.

The other organizing principle, the policy group, provides a mechanism to specify policies applying to operations in multiple views, or to a subset of a view. For example, a service view can provide a mapping from requestor to role, but the rights granted to a role may be linked to the underlying operation and be consistent across views. A role can be used to control access to operations within a view (in a traditional access control fashion) or enable content-filtering policies to control the content of a message (limiting the size of a purchase order is a commonly used example).

Additionally, there may be policies applicable to a variety of operations, regardless of which views they appear in. A policy that cuts across both service view and policy groups would be a policy to prevent a requestor from seeing information concern-

Conclusion

I've described a number of practical issues related to forming effective policies for managing Web services as messages flow back and forth. Given the large number of these and the multitude of ways they can interact, we've described two mechanisms, service views and policy groups, that provide an effective means of organizing policies into different coherent groups and reflecting those groups out to partners as WSDL. By presenting different specific WSDLs to different groups, an organization can succeed in controlling the cacophony of different agreements and capabilities of many partners, as well as insulating the underlying base services from the filtering, authentication, authorization, and other ancillary processing performed at the management layer.

The field of Web services management is still in the early stages, but already it has shown itself to be a complex area requiring

new techniques beyond previous management tools. New complexity generally requires new abstractions to reorganize information and keep it from degenerating into chaos; service views and policy groups are techniques to control the additional requirements and possibilties of Web services. Without this, Web services themselves will become a drain on the organizations that deploy them. But they sit at the bottom of a new stack of management issues surrounding Web services. Above this are other exciting areas, such as business process orchestration and real-time business intelligence. What new techniques will be added to the list will become apparent as this technology moves to maturity. ⓔ

■ About the Author

Dr. Matthew Fuchs is a member of the technical staff at Westbridge Technology. Previously, he was chief scientist for XML Technologies at Commerce One, and pioneered the theory and practice of using domain-specific languages in XML and SGML for distributed applications and agent-oriented communication over the Internet. At Commerce One he developed a variety of XML technologies, including SOX, the first implemented, publicly available, object-oriented Schema language and parser for XML.

Listing 1
```
POST /TradeSecurity.jws HTTP/1.1
Host: demo.westbridgetech.com
Connection: close
Accept: */*
Content-Length: 1255
Content-Type: text/xml; charset=utf-8
SOAPAction: ""

<SOAP-ENV:Envelope xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/"
                    xmlns:SOAP-ENV="http://schemas.xml-
soap.org/soap/envelope/"
     xmlns:impl="http://example.com/TradeSecurity.jws-
impl"
     xmlns:intf="http://example.com/TradeSecurity.jws"
     xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"

xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
    <SOAP-ENV:Header>
        <wsu:Timestamp
xmlns:wsu="http://schemas.xmlsoap.org/ws/2002/07/utility">
            <wsu:Created>2003-10-
31T18:46:56Z</wsu:Created>
            <wsu:Expires>2003-10-
31T18:47:06Z</wsu:Expires>
        </wsu:Timestamp>
        <wsse:Security xmlns:wsse="http://schemas.xml-
soap.org/ws/2002/07/secext"
                SOAP-ENV:actor="http://tempuri.org/"
SOAP-ENV:mustUnderstand="1">
 ...
        </wsse:Security>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body>
        <m:BuySecurity
xmlns:m="http://example.com/TradeSecurity.jws"
                    SOAP-ENV:encodingStyle=
"http://schemas.xmlsoap.org/soap/encoding/">
            <tickerSymbol
xsi:type="xsd:string">XXX</tickerSymbol>
            <amount xsi:type="xsd:string">1212</amount>
        </m:BuySecurity>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Listing 2a
```
<definition name="Support" targetNamespace="http://exam-
ple.com/InternalSupport"
     xmlns:tns="http://example.com/InternalSupport"
     xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
     xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
     xmlns="http://schemas.xmlsoap.org/wsdl/"

xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/04/secext">
    <message name="SubmitIssueInput">
        <part name="contact" type="xsd:string"/>
        <part name="component" type="xsd:integer"/>
        <part name="description" type="xsd:string"/>
    </message>
    ...
    <portType name="Support">
        <operation name="SubmitIssue">
            <input message="tns:SubmitIssueInput"/>
    ...
```

```
        </operation>
        <operation name="ListOpenIssues">
 ...
 </operation>
 <operation name="AssignIssue">
 ...
 </operation>
    </portType>
    <binding name="SupportBinding" type="tns:Support">
        <soap:binding style="rpc"
transport="http://schemas.xmlsoap.org/soap/http"/>
        <operation name="SubmitIssue">
            <soap:operation soapAction="submit"/>
    <input>
        <soap:body use="encoded"
namespace="http://example.com/Support"

encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
        </input>
    ...
        </operation>
                ...
    </binding>
    <service name="SupportService">
        <port name="Support" binding="tns:SupportBinding">
            <soap:address
location="http://internalserver/Support"/>
        </port>
    </service>
</definition>
```

Listing 2b
```
<definition name="Support" targetNamespace="http://exam-
ple.com/PurchaseOrders"
     xmlns:tns="http://example.com/PurchaseOrder"
     xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
     xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
     xmlns="http://schemas.xmlsoap.org/wsdl/">
    <types>
        <schema targetNamespace="urn:purchaseProcessing">
            <element name="PurchaseOrder"/>
        </schema>
    </types>
    <message name="PurchaseOrderInput">
        <part name="body" element="po:PurchaseOrder"/>
    </message>
    <portType name="OrderProcessing">
        <operation name="SendPurchaseOrder">
            <input message="tns:PurchaseOrderInput"/>
        </operation>
        <operation name="ChangeOrder">
            ...
        </operation>
        <operation name="CancelOrder">
            ...
        </operation>
    </portType>
    <binding name="POBinding" type="tns:OrderProcessing">
        <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
        <operation name="SendPurchaseOrder">
            <soap:operation soapAction="send"/>
            <input>
                <soap:body parts="body" use="literal"/>
            </input>
            ...
        </operation>
        ...
```

Middleware is Everywhere. **Can you see it?**

```
        </binding>
        <service name="OrderProcessing">
            <port name="PurchasePort" binding="tns:POBinding">
                <soap:address
location="http://internalserver/PurchaseOrders"/>
            </port>
        </service>
</definition>
```

Listing 3a

```
<definition name="Combined" targetNamespace="http://exam-
ple.com/CustomerSupport"
        xmlns:tns="http://example.com/PurchaseOrder"
        xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
        xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
        xmlns="http://schemas.xmlsoap.org/wsdl">
    <types>
        <schema targetNamespace="urn:purchaseProcessing">
            <element name="PurchaseOrder"/>
        </schema>
     <schema targetNamespace="http://example.com/identity">
            <element name="username" type="xsd:string"/>
            <element name="password" type="xsd:string"/>
        </schema>
    </types>
    <message name="PurchaseOrderInput"
xmlns:id="http://example.com/identity">
        <part name="username" element="id:username"/>
        <part name="password" element="id:password"/>
        <part name="body" element="po:PurchaseOrder"/>
    </message>
    <message name="SubmitIssueInput">
        <part name="username" type="xsd:string"/>
        <part name="password" type="xsd:string"/>
        <part name="contact" type="xsd:string"/>
        <part name="component" type="xsd:integer"/>
        <part name="description" type="xsd:string"/>
    </message>
    <portType name="Combo">
        <operation name="SendPurchaseOrder">
            <input message="tns:PurchaseOrderInput"/>
            ...
        </operation>
        <operation name="SubmitIssue">
            <input message="tns:SubmitIssueInput"/>
            ...
        </operation>
    </portType>
    <binding name="binding" type="tns:Combo">
        <soap:binding
transport="http://schemas.xmlsoap.org/soap/http"/>
        <operation name="SendPurchaseOrder">
            <soap:operation style="document"
soapAction="send"/>
                <input>
                    <soap:body use="literal"/>
                </input>
                ...
        </operation>
        <operation name="SubmitIssue">
            <soap:operation style="rpc" soapAction="submit"/>
            <input>
                <soap:body use="encoded"
namespace="http://example.com/Support"

encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
            </input>
        </operation>
    </binding>
    <service name="AcmeSpecialService">
        <port name="AcmePort" binding="tns:Combo">
            <soap:address
location="https://example.com/AcmeServices"/>
        </port>
    </service>
</definition>
```

Listing 3b:

```
<definition name="Combined" targetNamespace="http://exam-
ple.com/ZenithServices" xmlns:wsi="http://ws-
i.org/schemas/conformanceClaim/" xmlns:tns="http://exam-
ple.com/PurchaseOrder"
xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
```

```
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/04/secext">
    <types>
        <xsd:schema
targetNamespace="urn:purchaseProcessing">
            <xsd:element name="PurchaseOrder"/>
        </xsd:schema>
        <xsd:schema
targetNamespace="http://example.com/PurchaseOrders">
            <xsd:import
namespace="http://schemas.xmlsoap.org/ws/2002/04/secext"/>
            <xsd:import namespace="http://ws-
i.org/schemas/conformanceClaim"/>
        </xsd:schema>
        <xsd:schema
targetNamespace="http://example.com/Support">
            <xsd:element name="SubmitIssue">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name="contact"
type="xsd:string"/>
                        <xsd:element name="component"
type="xsd:integer"/>
                        <xsd:element name="description"
type="xsd:string"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
        </xsd:schema>
    </types>
    <message name="PurchaseOrderInput">
        <part name="body" element="po:PurchaseOrder"/>
    </message>
    <message name="SubmitIssueInput">
        <part name="body" element="sup:SubmitIssue"/>
    </message>
    <message name="ws-security">
        <part name="header" element="wsse:Security"/>
    </message>
    <message name="wsi-basic">
        <part name="header" element="wsi:Claim"/>
    </message>
    <portType name="Combo">
        <operation name="SendPurchaseOrder"></operation>
        <operation name="SubmitIssue"> ... </operation>
        <operation name="ListOpenIssues"> ... </operation>
        <operation name="ChangeOrder"> ... </operation>
    </portType>
    <binding name="binding" type="tns:Combo">
        <soap:binding
transport="http://schemas.xmlsoap.org/soap/http"/>
        <operation name="SendPurchaseOrder">
            <soap:operation style="document"
soapAction="send"/>
            <input>
                <soap:body parts="body" use="literal"/>
                <soap:header message="tns:ws-security"
part="header"/>
                <soap:header message="tns:wsi-basic"
part="header"/>
            </input>
            ...
        </operation>
        <operation name="SubmitIssue">
            <soap:operation style="document" soapAction="sub-
mit"/>
            <input>
                <soap:body use="literal"
namespace="http://example.com/Support"/>
            </input>
            ...
        </operation>
        ...
    </binding>
    <service name="ZenithSpecialServices">
        <port name="ZenithPort" binding="tns:Combo">
            <documentation>
                <wsi:Claim conformsTo="http://ws-i.org/pro-
files/basic/1.0"/>
            </documentation>
            <soap:address
location="https://example.com/ZenithWebServices"/>
        </port>
    </service>
</definition>
```

WASP UDDI 4.6

Extra features add to a solid product

■ If you're looking to deploy a UDDI registry that provides strong standards support, a capable API, and security and management capabilities, look no further than Systinet's WASP UDDI version 4.6. WASP UDDI is a UDDI server that supports UDDI specification versions 1 and 2 as well as the version 3 subscription API. Systinet has also added extensions to the core UDDI specification to provide additional functionality around management, security, inquiries, and other operations. The server can run on top of a number of databases, including Oracle, SQL Server, DB2, PostgreSQL, Sybase, Cloudscape, PointBase, and Hypersonic SQL (included).

Working with UDDI

Along with a full-featured Web interface, Systinet's WASP UDDI provides a rich, open-source client API written in Java that developers may leverage to create applications that interact with the UDDI repository. For this review, I will be focusing on the supplied Java API and exploring various pieces of the UDDI server's functionality.

WRITTEN BY
BRIAN BARBASH

Publishing and Categorizing Services

Both the Web interface and the Java API for the UDDI registry allow elements to be published to the registry. Out of the box, the UDDI registry is populated with sample data and a set of common taxonomies to support the tutorials and exercises in the documentation and allow services to be classified with existing industry standards. For this review, I'll use a new business entity to store all of the information for this article.

Using the Web interface shown in Figure 1, I've created a new business entity. The entity's profile is divided across the tabs representing various data structures within the UDDI specification. To provide the best possible data for searching, the new entity includes a contact, discovery URLs, and category information taken from the set of prepopulated taxonomies in the UDDI registry. Now that the business entity is available, additional objects will be published via the API interface to the registry.

In terms of Web services, publishing information into the WASP UDDI registry is extremely easy. As shown in Listing 1, it takes a mere four lines of code to publish a WSDL document into the registry.

Essentially, the developer provides a WSDL specification located at a particular URL and the business key to which it should be attached. In this example, the business key is the key that was generated when the business entity was added from the Web interface. Behind the scenes, WASP UDDI publishes the WSDL document according to the OASIS Technical Note "Using WSDL in a UDDI Registry, Version 2.0" (www.oasis-open.org/committees/uddi-spec/doc/tn/uddi-spec-tc-tn-wsdl-v2.htm). The result is a new business service that corresponds to the Service element in the WSDL under the specified business entity, and the Port Type and Binding information created as tModels.

Once information has been published to the UDDI registry, it is easily categorized using either the preconfigured taxonomies or custom taxonomies provided by a server administrator. Custom taxonomies may be added to the registry using the Web interface or the Java API. When working with the Web interface, a taxonomy structure may be defined manually. Alternatively, both the Web interface and the Java API support the upload of a taxonomy structure from an XML file that adheres to a specific Schema definition. Conversely, any taxonomy may be exported to XML for storage or transfer to another registry.

Finding What's Out There

The WASP UDDI server provides inquiry capabilities that adhere to the UDDI version 1 and 2 standards as well as a "SuperInquiry" interface that adds search options to the standard interface. Similar to the WSDL publishing process, it's easy to search a WASP UDDI registry using the Java API. Listing 2 is an example of searching for the service that was published using the previous code example. The result is a ServiceList object from which the binding information may be retrieved.

Systinet also provides a custom API for looking up and invoking Web services that are stored in any UDDI registry adhering to the
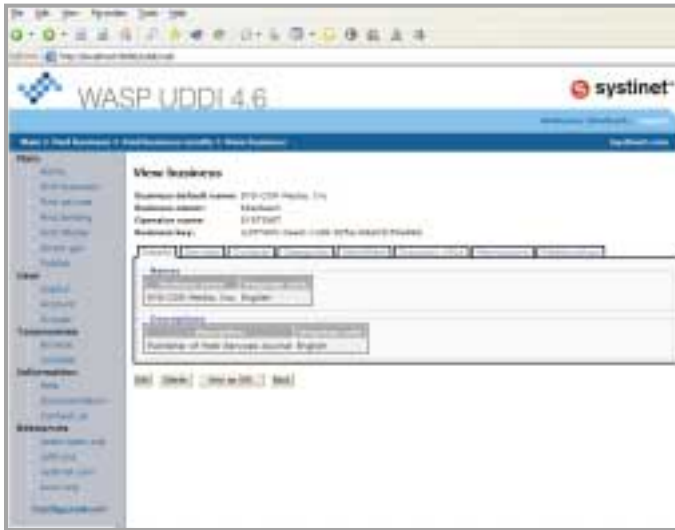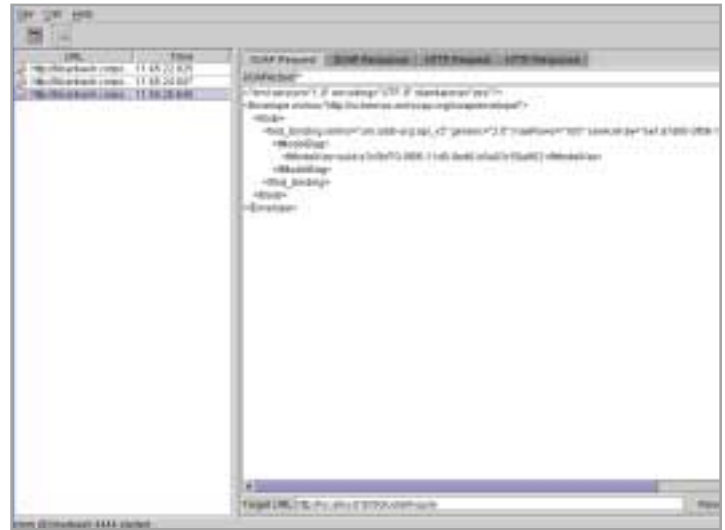
FIGURE 1 | **Business Entity**

FIGURE 2 | **SoapSpy**

OASIS Technical Note referenced earlier. The ServiceClient object accepts a specialized URL that locates a Web service in a UDDI registry by its binding key. For example, systinet-uddi:http://localhost:8080/uddi/inquiry?bindingKey=ee2b38a0-0f06-11d8-9ed6-b8a03c50a862. Once a handle to the service has been acquired, it may be executed using a generic WASP proxy or a specialized proxy that was generated from the WSDL using WASP tools.

Debugging Applications

Systinet provides the SoapSpy utility to assist with the debugging process when working with its UDDI registry. SoapSpy acts as a proxy that intercepts incoming requests and forwards them to the registry host. In Figure 2, an instance of SoapSpy has captured the inquiry message of a request to look up the WorkoutService. Messages that have been captured may be modified and re-sent to correct processing errors or perform tests on specific data conditions.

Administration and Security

Systinet has provided several administration and security features in the registry. One of the major pieces of administrative functionality is the approval process for new registry data. In this scenario, two UDDI registries are configured: staging and production. The staging repository is the endpoint for new services while the production repository is accessible in a read-only mode and contains only those services that have been approved by an administrator. When a service is published into the staging repository, the developer may mark the information as Ready-for-Approval. Once marked, an approval request is generated and the administrator is notified by e-mail. Once approved by an administrator, the service is automatically published to the production environment.

Security is a major component to the WASP UDDI product. WASP UDDI allows the assignment of users and groups to each UDDI element to control who can search for, retrieve, create, update, or delete any element within the registry. In addition to providing its own internal security mechanism, WASP UDDI may be integrated with LDAP, Microsoft Active Directory, and Kerberos. Access Control Lists within the UDDI registry may be administered from these external systems.

Summary

Systinet's WASP UDDI server is an easy-to-use, full-featured UDDI registry. In addition to providing support for versions 1 and 2, as well as parts of the version 3 UDDI specification, several enhancements have been provided. The Java API is full featured and does an excellent job of hiding the communications details of interacting with the registry. The additional security and administrative functions provide unique management capabilities for the product. Overall, Systinet's WASP UDDI server is a very solid product. ℮

■ **About the Author**

Brian R. Barbash is the product review editor for *Web Services Journal.* He is a consultant for the Consulting Group of Computer Sciences Corporation, where he specializes in application architecture and development, business and technical analysis, and Web design.

■ ■ ■ bbarbash@sys-con.com

Listing 1: WSDL Publishing
```
WsdlClient client = new WsdlClientImpl();
RegistrySpecification spec = new RegistrySpecification(
"admin",
"changeit",
"http://localhost:8080/uddi/inquiry", "http://localhost:8080/uddi/publishing");
PublishingInput input = new PublishingInput(
"http://localhost/DataManagerService/WorkoutService.asmx?wsdl",
new BusinessKey("c4ce67a0-0ef4-11d8-9ed6-b8a03c50a862"));
client.publish(spec, input);
```

Listing 2: Inquiry API
```
FindService finder = new FindService();
finder.addName(new Name("WorkoutService"));
UDDIApiInquiry inquiry = UDDILookup.getInquiry(
"http://localhost:8080/uddi/inquiry");
ServiceList list = inquiry.find_service(finder);
```

Download the code at
sys-con.com/webservices

Optimizing Web Services Using Java, part II

Increased flexibility, reusability, and power

■ In the first article of this series (**WSJ**, Vol. 3, issue 12), I described generic Java and examined the issues involved in supporting variant generic types in Java. That article also explained how generic variant types increase the readability, maintainability, and safety of our code.

I examined the implications of using variance annotations in class and interface type parameters for Web services and I demonstrated how the new concepts allow a better abstraction and maintainability, retaining and improving at the same time the static type safety of our code (code that is checked at compile time).

Based on previously defined concepts, I'll now build the service endpoints and clients with variant generic Java types. In this article we will use the Java Web Services Developer Pack (JWSDP 1.3) with JAX-RPC 1.1.

JAX-RPC and Generic Java

If a Web service is a software system designed to support interoperable machine-to-machine interaction over a network, what are the implications of generic types for developing Web services in Java? To answer the question I'll use the JAX-RPC (Java API for XML-based Remote Procedure Call), which supports interoperability across heterogeneous platforms and environments. How will the new generic types influence our development? A generics-enabled compiler could perform the type-checking, formerly done at runtime, at compile time. In the process, the code becomes safer to use.

If polymorphism is the capability of an

WRITTEN BY

JORDAN ANASTASIADE

object to take different forms, then parametric polymorphism is useful for defining generic behavior. To be able to utilize parametric polymorphism in an object-oriented, distributed computing programming environment could lead us into the development of what would probably be called a meta–Web service. Parametric polymorphism has become an integral part of many modern programming languages and will soon include Java. However, for the moment the main issue remains the mapping from the Java types to the XML data types; in other words, the standard mapping from the Java definitions to WSDL definitions. Thus, the values of a JAX-RPC supported Java type must be serializable to and from the corresponding XML representation. What, currently, are the Java types defined in the final version of JAX-RPC 1.1?

- All Java primitive types
- A subset of standard Java classes:
  - java.lang.String
  - java.util.Date
  - java.util.Calendar
  - java.math.BigInteger
  - java.math.BigDecimal
  - javax.xml.namespace.QName
  - java.net.URI
  - Classes in the Java Collection Framework

that are mapped using pluggable serializers and deserializers.
- JAX-RPC Value Types: Basically, a Java object whose state can be moved between a service client and service endpoint.

A Java class must follow these rules to be a JAX-RPC conformant value type:
- Class must have a public default constructor.
- Java class must not implement (directly or indirectly) the Remote interface.
- Java class may implement any Java interface (except Remote interface) or extend another Java class.
- Java class may contain public, private, or protected information. Fields must be supported JAX-RPC data.
- Java class may contain static or transient fields.
- Java class for a JAX-RPC value type may be designed as a JavaBeans class with simple properties.

A Java array has members of a supported JAX-RPC Java type.

You may have some legitimate concerns: "What about our generics types? How are we going to map a generic type to an XML type using the pretty tight set of rules of the JAX-RPC specifications?" A JAX-RPC implementation may need to support mapping between XML data types and Java types beyond what is defined today by the standard type-mapping specification. The good news is that the JAX-RPC specification also defines a set of APIs to support an extensible type-mapping framework, thus allowing the development of pluggable serializers and deserializers. The support for extensible mapping between any Java type and XML data type is defined by the JAX-RPC 1.1 specification.

Still, there is another important issue regarding the use of generic types in Web services. Let us suppose that you have a generic class defined like this:

```java
public class MyServiceImpl<T> { }
```

As you already know, the parameterized class or interface declarations define a set of types, one for each possible instantiation of the type parameter. Basically, the compiler completes the compile time checks of your code and then forgets about the type of the parameter in order to main-

tain compatibility with previous JVMs and with your class libraries. The process is called erasure, since no information about type parameters is maintained at runtime. Is there a way to discover the actual type parameter from all the possible instantiations of the type parameters? The short and sad answer is no.

On the other hand, in general you can enforce the client of your class to pass a class literal for the type parameter by the time it uses your class. The idiom is easily implemented, defining one of the constructors of your class as:

```
public class GenericServiceImpl<T> {
  private Class<T> t;
  GenericServiceImpl(Class<T> t) {
    this.t = t;
  }
}
```

The idiom is type safe and enforceable at compile time, but it is somewhat limited because the parameter type T cannot be a generic type itself. However, it will allow the client to rediscover the instantiation of the service endpoint based on the type parameter. On the other hand, for the information to be available in a distributed computing environment model, like Web services, we also need the tools that create proxies to maintain it. One such tool is wscompile, which uses a WSDL document to generate stubs. The tool should be extended to support the generics-type information in its generated artifacts. I built the Web services samples for this article with the tools that are available today, namely, the tools from JWSDP1.3. I compiled the generic code with a prototype implementation of a Java compiler. The compiler supports generics as described in JSR014 – Adding Generics to the Java Programming Language v. 2.2., as well as some other language constructs, like enhance "for loop", under consideration for the 1.5 release of the J2SE platform.

Hello Web Service: A Generify Model

If we were to build a Web service in JAX-RPC, I'd first have to describe to you the steps involved in setting the proper environment for allowing generics Java to compile and to be executed. First you have to use a special JAR file, "gjc-rt.ja"' that comes with the prototype implementation of a Java compiler. You should place the JAR file on your JVM's boot-

strap classpath. It contains both the generic compiler and a number of modified platform classes. The changes are easy to set if in your development process you use a wonderful tool like Ant. The first step is to modify the "compile target" in your xml target file like:

```
<javac srcdir="${src}"
        destdir="${build}"
        includes="*.java"
        fork="yes"
executable="${env.JSR14DISTR}/scripts/ja
vac.bat">
```

where JSR14DISTR is an environment system variable that could be defined in your Ant script with <property environment="env"/>.

The only outstanding issue is a little more subtle and it took me a while to figure it out. The problem is how to instruct the Tomcat servlet container to change its JVM settings when it runs. Under the present incarnation, the JWSDP1.3 has a directory called {jwsdp-1.3}/jwsdp-shared/bin, where {jwsdp-1.3} is the root directory of your jwsdp installation. Inside the jwsdp-shared/bin subdirectory there is a file called "launcher.xml". The only thing you have to do is set a property element in your XML file like:

```
<property name="jwsdp.vm.args" value="-
Xbootclasspath/p:${env.JSR14DISTR}/gjc-
rt.jar"/>
```

One more setting and you 're ready to develop Web services and clients via generics Java. In order for the client to run properly, in your target XML file you have to set the "run-client" property like the following:

```
<target name="run-client" >
      <java classname="${client.class}"
fork="yes" >
        <jvmarg value=
"-Xbootclasspath/p:${env.JSR14DISTR}/gjc-
rt.jar"/>
        <arg line="${endpoint.address}"
/>
        <classpath refid="run.classpath"
/>
      </java>
  </target>
```

To prove that your environment is set properly you would probably try, as I did, to

write a modified version of a Hello Web service, using generics. I won't go into details describing the JAX-RPC model; rather I will try to emphasize the issues related to the new concepts at hand (see Listing 1; the code for this article is online at www.sys-con.com/webservices/sourcec.cfm). The interface that describes your Web service and also the servant, the implementation class of the service, could be done using parameter types. Here is a simple definition:

```
public interface HelloIF<T>
                extends Remote {
  public LinkedList<T>
  sayHello(
      T t
    ) throws RemoteException;
}
```

For the implementation class (Listing 2), there is nothing out of the ordinary except the idiom that I mention above, where one of the class constructors has as a parameter, a class literal of the type variable:

```
public class HelloImpl<T>
        implements HelloIF<T> {
  private Class<T> t;
  public HelloImpl(Class<T> t) {
    this.t = t;
  }
```

The implementation would have allowed the client to rediscover the instantiation of the service endpoint based on the type parameter had we had in a generated stub a way to maintain the information about type parameters. Otherwise, besides the constructors, the class implements "sayHello", the single method of Hello Web service:

```
public LinkedList<T>
  sayHello(
      T t
    ) {
    LinkedList<T> list =
        new LinkedList<T>();
    list.add(t);
    return list;
}
```

What remains to be added for building your service are the classic configuration files like jaxr-pc-ri.xml, config-interface.xml, web.xml. The

complete code implementation with the directory structure and all the necessary files can be found at http://cs.senecac. on.ca/~jordan.anastasiade/articles/WSJ. Once you build your service on your machine, it would be interesting to analyze your WSDL file:

```
http://localhost:8080/hello-generify-jaxrpc/hello-generi-
fy?WSDL
```

What about a client for your first generify Hello Web service? The client could be written following the classic three models defined in JWSDP: a static stub client, a dynamic proxy client, and a dynamic invocation interface (DII) client. For test purposes, I developed only a simple static stub client (see Listing 3). After I obtained the stub, I invoked the single available method. The type variable for my parameterized Web service is the Date class from java.util package:

```
HelloIF service = (HelloIF)stub;
LinkedList<Date> list =
    service.sayHello(new Date());
for (Date d : list)
   System.out.println(d);
```

You could modify the type parameter to any class you wish to use as long as your class is currently a JAX-RPC supported Java type. What the generify Hello Web service is able to give you in return is a string representation of your class. It is a way of saying hello from a class point of view! Even though I also could have written a parameterized client type, a so-called generify client, I reserved the pleasure for the next Web service, which explores more subtle issues related to generic data types.

Human Resources Web Services

Consider a simple class hierarchy containing classes like employees, sales persons, and programmers; you could define it for describing a company work force. Suppose you are a developer and your manager asks you to implement some simple operations for a human resources deparment. Your implementation should be exposed like a Web service. To have the implementation of the class hierarchy complete for a Web service, you are going to implement three value type classes like employees, sales persons, and programmers (see Listing 4). A value type is a class whose state may be passed between a client and a Web service as a method parameter or return value. Suppose that the company has departments, and there is a list of all departments and for each department there is a list of the employees (see Listing 5). In the real world, the data would be kept in databases, but from your point of view as a Web services developer, the way the data is stored and retrieved is only an implementation detail.

As the input data, a list of all departments and a list of employees for each department are given to you (see Listing 5) and you are asked to:

1. Calculate the salary of each department
2. List employees from each department
3. Move some employee from one department to another

4. Calculate the company size by the number of employees

The first thing you would probably do is write the service endpoint interface. Let us consider one method at a time. To find out the salary of a department the parameter of your method should be a List. What kind of List, you might ask? It might be a List of Employees, because every object of your hierarchy is at least an object of type Employee. Let's try to calculate the salary of the sales department. You have to use a List of salespersons as an actual parameter. Is the list of sales employees a subclass of a list of employees? It is if you use as a parameter a covariant generic type: List<? extends T>, where your method uses the list for read-only operations and could be defined as:

```
public int
departmentSalary(
    List<? extends Employee> emp
) throws RemoteException;
```

How would you describe the method signature for showing employees? If you consider that your parameter list could be any object, a bivariant generic parameter List<?> seems to be the best selection and your method will most probably look similar to:

```
public LinkedList<String>
showEmployee(
    List<?> list
) throws RemoteException;
```

Now consider the method that moves employees from one department to another. As you would probably expect, the destination argument should be a covariant generic type, because you are going to write into the destination List, while the source must be a cotravariant generic type, because you read it from the source List.

```
public <T> void
move(
    List<? extends T> source,
    List<? super T> destination
) throws RemoteException;
```

Finally, you have to define the signature of the method that calculates the size of

the company. It is clear that the parameter should be the list that contains the list of departments. The list is a covariant generic type like List<? extends List<?>> on the parameter List<?>. The signature of the method allows it to be called any list of lists, for instance a List<List<Sales>> or a List<List<Programmer>>, while the inner list has a bivariant generic type. Thus, the method definition to calculate the number of employees of a company seems to be somewhat like:

```
public int
companySize(
    List<? extends List<?>> departs
) throws RemoteException;
```

Once you finish the most difficult part, namely the service endpoints interface (see Listing 6), the implementation of the Web service class is defined as:

```
public class HumanImpl<T>
        implements HumanIF<T> {
```

It could be as complicated as you need it to be, eventually connecting with databases to obtain and store the data. The method implementations would use the variant generic types for the arguments to accomplish the task at hand. For instance, the implementation method for calculating the salary per department would come across as:

```
public int
departmentSalary(
    List<? extends Employee> emp
) {
    int salary = 0;
    for (Employee e: emp)
      salary += e.getSalary();
    return salary;
}
```

The simplified, although fully functional, class implementation for the human resources Web service is in Listing 7. What remains to be defined are the configuration files and the XML file for building the Web service using the Ant tool. Those files are normal, standard files, and the code for the client (see Listing 8) and the whole project, including a static stub client can be found at http://cs.senecac.on.ca/~jordan.anastasi-ade/articles/WSJ.

The Web Services Interoperability Organization (WS-I) was created to promote Web services interoperability across platforms, operating systems, and programming languages. In the days to come, the challenges for using generic variant Java types in developing Web services would be measured by the degrees on which, following WS-I specifications, we would be able to improve the mapping of generic types to and from WSDL files. The JAX-RPC specification allows APIs to support an extensible type-mapping framework.

Conclusion

The Web services built in this article showed that the variant generic types offer increased flexibility, reusability, and expressive power to the Java programming environment for Web services.

Yet the ability to use parametric polymorphism has not made the transition from single-environment programming languages to distributed-programming environments like Web services because of the need for tools to map to and from variant generic types and XML files.

If a Grid is a collection of distributed computing resources available over local- or wide-area networks, which appear to an end user or application as one large virtual computing system, then a generic programming model transition could help tremendously such a distributed computing technology. ℮

References
- *Java API for XML-Based RPC (JAX-RPC):* http://java.sun.com/xml/jaxrpc
- *Prototype for JSR014: Adding Generics to the Java Programming Language v. 2.2:* http://developer.java.sun.com/developer/earlyAccess/adding_generics/index.html
- *The Java Web Services Tutorial for JWSDP v1.3*: http://java.sun.com/webservices/tutorial.html
- Anastasiade, Jordan. (2003). "Optimizing Web Services Using Java, Part 1." *Web Services Journal* (Vol. 3, Issue 12)

■ About the Author
Jordan Anastasiade is professor of computer studies at Seneca College's Campus at York University, in Toronto. He has created industry-leading software products designing and developing distributed applications. Currently he is teaching Java technology and Web services.
■ ■ ■ jordan.anastasiade@senecac.on.ca

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

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

Real-time monitoring
No additional coding required, turn on instrumentation for a Web service, view all conversations as they execute, and drill down to views of individual operations and controls

Policy specification and enforcement
Define operational policies including security, logging and monitoring for Web services within BEA WebLogic Workshop 8.1

Explicit instrumentation with custom controls
Use Confluent Control to explicitly call operations such as monitoring and security from Web services

BENEFITS:

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

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

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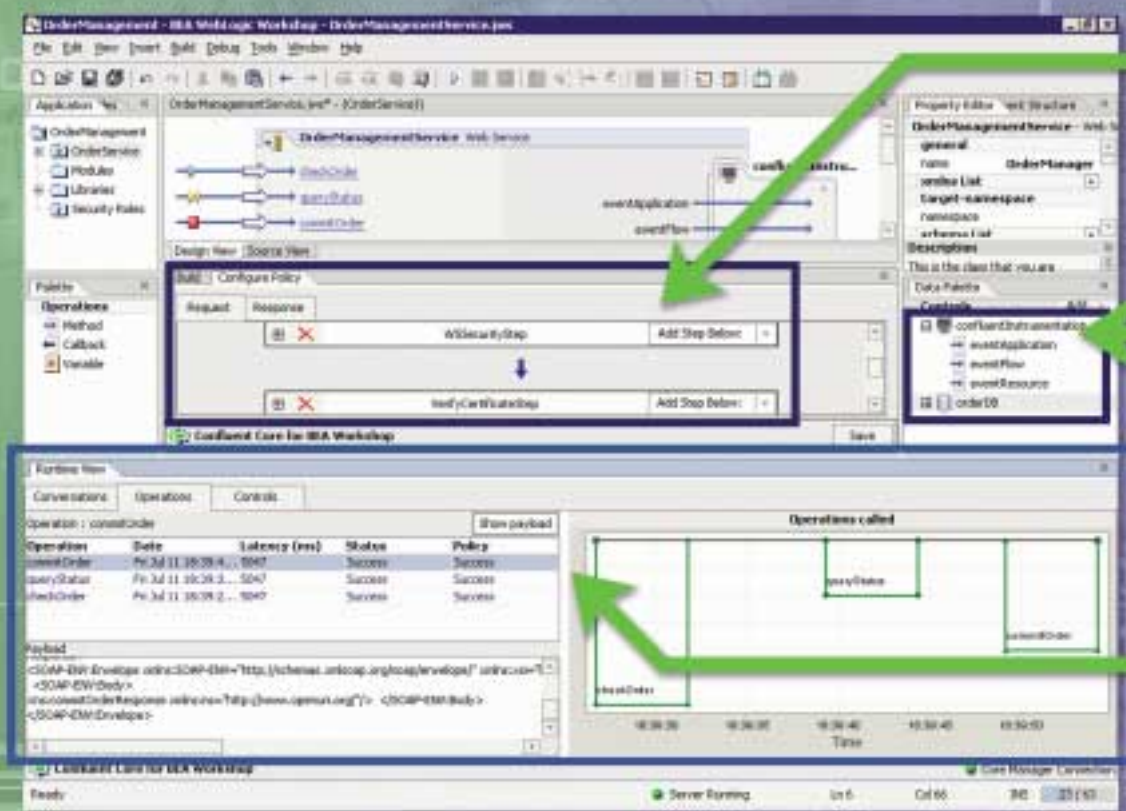
Policy specification and enforcements

Explicit instrumentation with custom control

Real-time monitoring

Confluent Software

Horses for Courses: Services, Objects, and Loose Coupling

Integration without compromise

■ Object-oriented technologies are used today in the design and development processes for many computer systems; it is a proven paradigm and has made possible the development of large and complex software systems. Enabling platforms and tools for building and consuming Web services will not be an exception. However, how a service is implemented using objects and the way in which it interacts with other services via message exchanges require very different approaches.

Today, most tools represent Web services to application developers as objects. Such an approach carries the danger of underperformance and fragile applications with too tight a level of coupling, which loses the benefits of service orientation. This article shows you how objects and services should be integrated to build loosely coupled, performant, and reusable services without compromising either paradigm.

Anatomy of a Web Service

To frame this discussion, we must first reach a common understanding of what exactly a Web service is. Our deliberately simple view of a service

WRITTEN BY

JIM WEBBER &

SAVAS PARASTATIDIS

is as an entity that exchanges messages (usually with other services), with well-defined boundaries. A Web service shares those characteristics, but adds the constraint that the messages exchanged between Web services are in fact SOAP messages carried over some suitable transport protocol (which may include application protocols like HTTP).

Those messages may contain out-of-band data for quality-of-service purposes, and are produced and consumed by a piece of middleware, usually called a SOAP server, SOAP stack, or SOAP message processor. It is this middleware that performs the translation of SOAP messages into a form more suitable for processing at the application level, which it typically does by converting the SOAP XML message into

application objects. This canonical Web service is shown in Figure 1, and will form the basis for later discussion on how objects and services interact. Please note that Figure 1 is only a conceptual view of a canonical Web service and there is no suggestion of a one-to-one association between a Web service and a particular host. Indeed, an entire infrastructure with computational, data, human, etc., resources could make a single Web service.

An Object-Oriented System

Before we explore the interaction between objects and services, it is useful to recall how we would implement a purely object-based system. Consider a simple purchase-order system that is used by a (potentially remote) purchase-order application, composed from a number of purchase-order objects.

In Figure 2, we see that the purchasing application holds a reference to (potentially remote) purchase order objects. Through those references, the application can obtain information about the purchase order, such as the order number, the date, the cost of the order, and the items requested.

While this approach is appealing because of its simplicity, it suffers from a number of drawbacks if we try to scale it to Internet-level computing. First, while the interface of the purchase order object is defined at the right level for a closed system (either a

stand-alone application or an intranet-based distributed object environment), it is far too fine grained to use efficiently over a wide area network. The network overhead of calling a method like getPurchase OrderNo() outside of a single domain is prohibitively high.

Second, we see that the level of coupling with this approach is very high. If, for example, the purchasing application and the purchase order object physically belong in separate organizations (which is the normal case in CORBA, where object references are propagated around an application and the underlying infrastructure handles location transparency), then the organization hosting the purchase order object can legitimately delete that object but, as a consequence, breaks the other organization's purchasing application, which relied on a reference to that object. Of course, technologies for maintaining referential integrity have been devised (e.g., distributed reference counting and garbage collection, leasing, etc.), but in most cases they don't scale very well either.

Finally, and most crucially, this approach assumes that there is a type system shared by the purchasing application and the purchase order object. If two application domains are unable to share a type system, for instance if one is built on .NET and the other on J2EE, then direct invocation of objects from one domain into the other is all but precluded, which means that specialist adapters then have to be deployed to solve the problem piecemeal.

While middleware like CORBA or DCOM can hide the differences between individual language and platform technologies and allow, for example, a C++ purchasing application to invoke a Java purchase order object, it does so because it uses its own metalevel type system, and does not solve the problem of sharing types but moves it to the metalevel. It also throws up the barrier that if there is no IDL mapping to the language that a specific component is written in, then there is no way that component can be used in the application.

Don Box of Microsoft sums this up very neatly when he notes that "shared abstractions are expensive." Box convincingly argues that for a distributed object approach to have a hope of working across enterprise boundaries, all parties involved in the construction of the application have to agree on the set of shared abstractions. As the num-ber of parties involved increases, there is a combined explosion of different systems, different people, and different corporate politics to navigate, and eventually the rewards for deploying such a system are outweighed simply by the effort required to deploy it.

An Object-like System Using SOAP and WSDL

If there are problems in scaling object-based systems, it seems an attractive idea to use technologies like SOAP and WSDL to alleviate that problem. One such common approach is shown in Figure 3.

The example shown in Figure 3 at first appears to solve the issues that we previously identified with integrating multiple enterprises' object-based systems. Certainly, it seems to solve the problem about having shared types, since now all type information is exposed via a WSDL interface that allows any software agent that understands the WSDL to invoke the object behind the service. It also means that specialist adapters can be replaced with commodity SOAP stacks in order to plumb various application components together. The ability to understand XML, XML Schema, and SOAP are the only assumptions being shared.
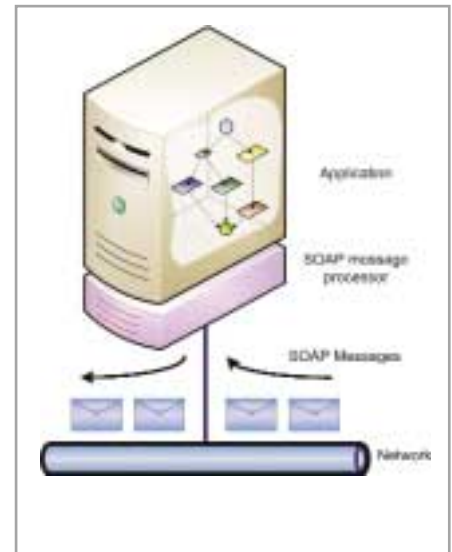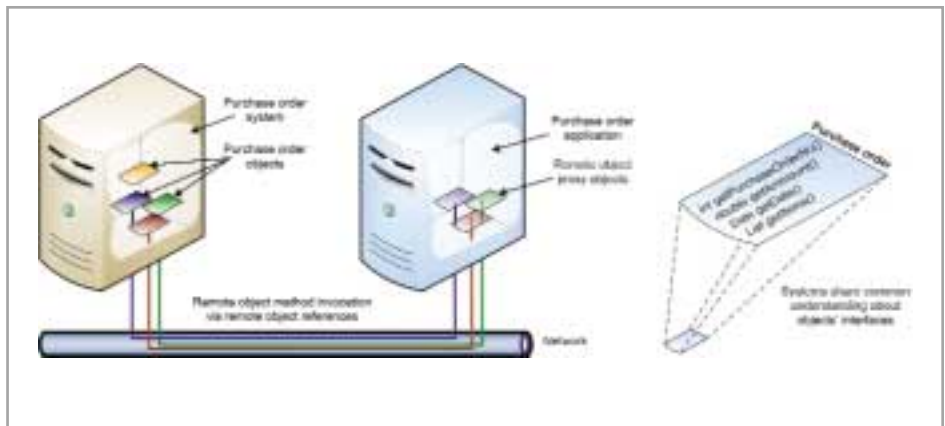

FIGURE 1 | A canonical Web service

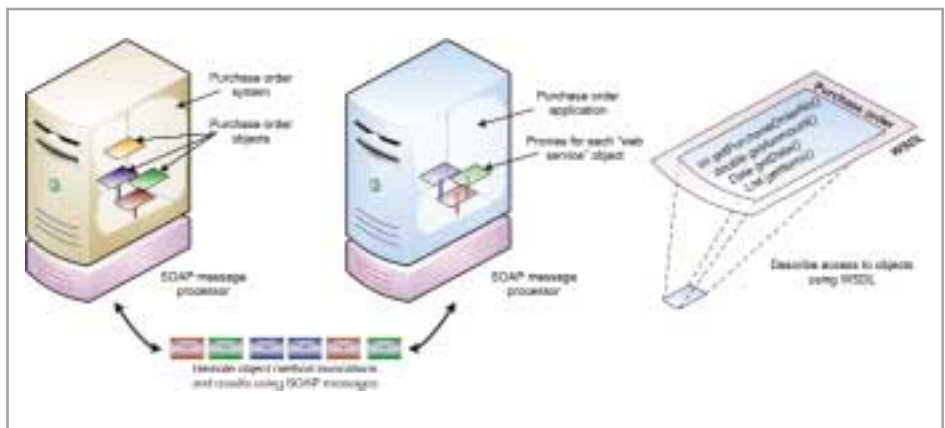

FIGURE 2 | Exemplifying the object-oriented approach

FIGURE 3 | Using SOAP and WSDL to Support an Object-Oriented Approach

However, there are significant flaws in this approach. While this approach may be platform neutral because of the XML payloads that we move across the network, applications built this way may not be performant because we have exposed the same fine-grained object interface for consumption, and we still have to bear the cost of translating to and from XML whenever we want to invoke such a method.

Worse, applications built this way are brittle. For example, imagine that the purchase order object is made available at the URL http://example.com/wsj/purchase Order/14529 and that the client binds to that URL and begins to invoke methods (via SOAP) on the purchase order object. Given that the purchase order object is a resource that belongs to an enterprise, that enterprise is perfectly within its rights to manage that resource as it sees fit, including to move or delete it. If the URL of the purchase order object changes or disappears, this has the knock-on effect of breaking the purchasing application.

While we have schemes such as that used in OGSI, which address this brittleness by providing a mechanism to re-resolve an endpoint in case a resource is moved, this patch fails to address the fundamental problem with this approach – that an enterprise's private resources (objects) are being exposed to the network. In short, this approach is nothing more than CORBA or DCOM with angle brackets, and while CORBA and DCOM were extremely useful technologies in their specific domain, in this context such an architecture is, at best, ungainly.

Given these drawbacks, we must ask what would drive developers and architects to propose such solutions. While one motivation is the (completely inaccurate) view that Web services are a yet another distributed object system, we feel the primary reason for this model having permeated so far into the developer mindset is tool support.

For instance, the WebMethod attribute in .NET makes it easy to expose an object as a Web service, even though such objects may well have fine-grained interfaces that are wholly unsuitable for exposure as a Web service. On the client side, tools like Axis WSDL2Java present the abstraction of a Web service as an object with methods (which is ironic given that at least some of the time it, lamentably, is).

A Service-Oriented View

So far we have seen the pitfalls of mixing service and object orientation. However, both are extremely valuable techniques and in order to derive maximum benefit from them we simply need to understand where it is best to utilize each, and how the object-service interface should look. To put things into perspective, let's revisit the purchase order system from a service-oriented architecture point of view.

The diagram shown in Figure 4 fixes all of the problems we encountered in examining previous approaches. There is much looser coupling in this system since no back-end resources are exposed across administrative boundaries. Instead, a (logically, though not necessarily physically) single, fixed endpoint is used as the entry point to a process that deals with the creation of a purchase order. When that process is invoked, some business-specific activity is executed and a purchase order document is passed back to the purchasing application. The purchase order service interface is now much more coarse grained as it captures an entire business process rather than individual method calls, and is likely to be performant and scalable.

However, this is only half the story as far as we are concerned. While this architecture is the correct choice for this application, we still need to see how it can be mapped into code in the service implementation. Figure 5 presents the details of the implementation of the purchasing order service and shows in greater detail how a service could be implemented using objects without actually exposing them.

The fundamental issue shown in Figure 5 is that it is the messages exchanged with Web services that should be consumed by applications, and not the Web services them-
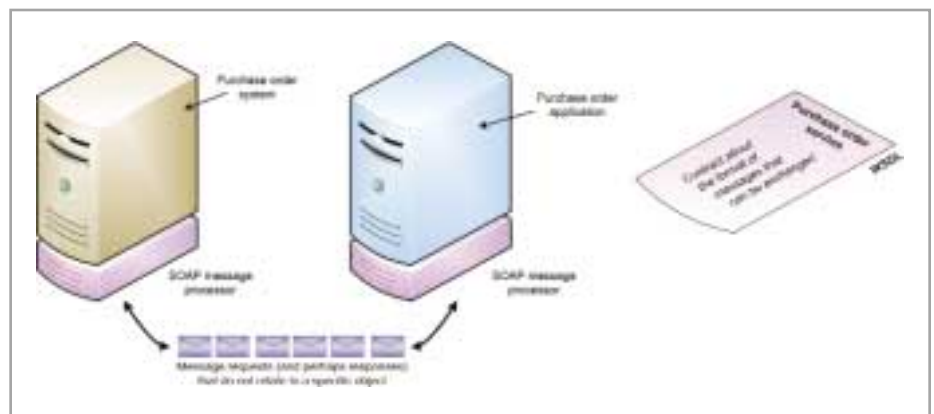

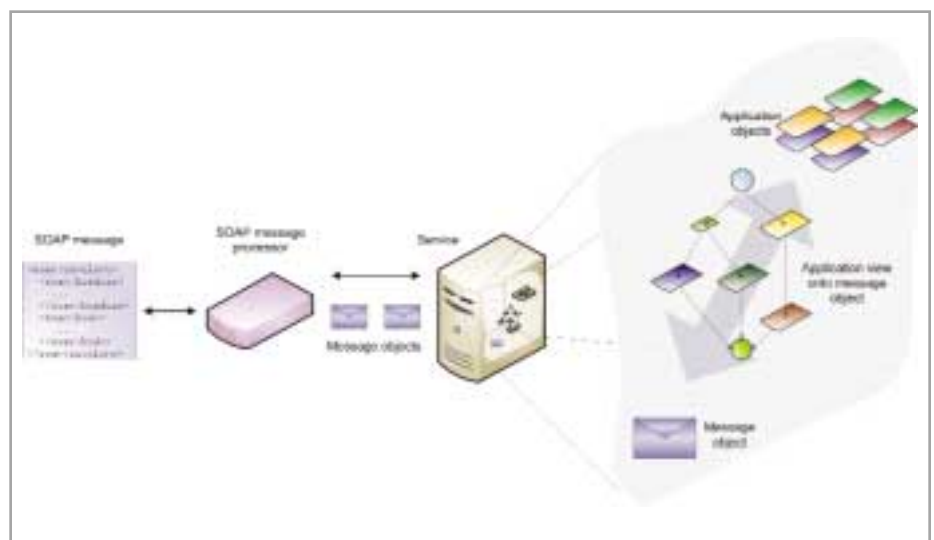FIGURE 4 | Restructuring the application in terms of business processes and SOA

FIGURE 5 | Viewing messages (not services) as application-level objects

selves. Where Web services are seen as application-level objects, there is the constant danger we'll fall into the trap of treating them as objects.

However, with this approach we use fewer but richer message exchanges, we increase the granularity of a Web service, and reduce the performance penalty. This strategy also ensures that loose coupling is the norm since applications never bind directly to services, only to messages. Since those messages can come from any source as far as the application is concerned, migrating the service provider to another service is not invasive to the application. Finally, since the architecture is concerned only with messages and message exchanges, the barrier to entry for any enterprise is low. There is no need to argue about whether a system should be based on J2EE, .NET, or some other technology since most platforms have Web services support nowadays.

Conclusion

Building object-like systems on top of Web services specifications is a strategy that leads to tightly coupled, brittle, and underperforming systems. While the current crop of tool support implicitly advocates this approach to the unwary, a far better solution is to think of an application in terms of services and the messages that those services exchange. When building code for those service implementations don't be tempted to bind directly to a Web service, but instead bind to the messages that the service produces and consumes. Such an approach promotes loose coupling; your applications will be able to transcend changes in the service they consume and will be much easier to maintain.

References

- Box, Don. (2003). "Service-Oriented Architecture and Programming (SOAP), Parts 1 and 2." Talks from the MSDN TV archive (http://msdn.microsoft.com/msdntv/episode.aspx?xml=episodes/en/20030827SOAPDB/manifest.xml, http://msdn.microsoft.com/msdntv/episode.aspx?xml=episodes/en/20030902SOAPDB/manifest.xml).
- Tuecke, Steve., et al. (2003) "Open Grid Services Infrastructure (OGSI) - Version 1.0". https://forge.gridforum.org/projects/ogsi-wg. ℮

■ About the Authors

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

■■■ jimwebber@hotmail.com

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

■■■ savas@parastatidis.name

XMLSPY 2004 from Altova

An excellent tool gets a powerful addition

■ Altova recently released the 2004 version of XMLSPY and introduced a new product, MAPFORCE 2004. This new version includes several updates and features that will assist with Web services development while also providing a graphical XML mapping tool for data transformation.

XMLSPY Visual Studio .NET Integration

One of the biggest new features in XMLSPY is an add-on that integrates into Microsoft Visual Studio .NET. This utility provides full XMLSPY functionality within the Visual Studio .NET environment. XML Schemas, XSL Stylesheets, WSDL files, among others, are all fully editable using the familiar XMLSPY tools from within the .NET IDE. Figure 1 shows a screenshot of the WSDL editor from within the Visual Studio .NET IDE.

XML Differencing

A powerful new feature of XMLSPY is the differencing engine. Useful in a wide array of scenarios, the differencing engine extends the concept of file comparison utilities to incorporate awareness of the unique characteristics of XML. For example, namespace prefixes may be changed within XML documents. The XMLSPY differencing engine, when configured to do so, will ignore prefixes. Therefore, two documents with different namespace pre-

WRITTEN BY
BRIAN BARBASH

fixes representing an instance of the same XML Schema will be shown to match if the other characteristics of the document match.

XMLSPY IDE Enhancements

Several enhancements to the XMLSPY IDE have also been included to help the Web services developer. Use of XPath often factors heavily into Web services development; XMLSPY provides a beta implementation of the W3C's XPath 2.0 specification based on the May 2003 working draft. When testing XPath statements, the developer may select between the 2.0 beta and 1.0 specifications.

When developing WSDL specifications for Web services, the data passed in and out of the service is often described as an XML Schema. XMLSPY provides a code-generation utility that builds code to manipulate instance documents of XML Schemas. The utility will generate Java code based on JAXP, C++ code based on MSXML4 or Xerces, and C# code.

The user interface of XMLSPY has changed slightly to make it easier to navigate between the various views of documents. As seen in Figure 1, below the main document window are buttons that correspond to the five views in which a document may be shown. These document buttons replace the toolbar items in previous versions.

XML Database Support

XMLSPY 2004 provides full support for Microsoft SQL Server XML extensions, Oracle XML DB extensions, and the Tamino Schema extensions for the Tamino Server. Each option

adds namespace declarations for the respective server vendors to the XML Schema document currently loaded. Once enabled, additional tabs are visible in the Details Entry helper component of the GUI that includes the available database-specific elements.

MAPFORCE 2004

A new sister product of XMLSPY 2004 is MAPFORCE 2004. MAPFORCE, as the name implies, is a graphical mapping tool for transforming XML documents from one schema definition to another. It provides a graphical interface in which transformations are built using drag-and-drop operations. Once the mapping is complete, MAPFORCE may be used to output an XSL Stylesheet, Java source code, C# source code, or C++ source code that in turn may be used in custom applications to perform data transformation.

Summary

XMLSPY continues to be an excellent tool for the XML and Web services developer. The new features and enhancements of the 2004 version release add to its extensive portfolio of functionality. Complementing the 2004 release is the MAPFORCE product. Its ability to create data transformation logic graphically and produce code in multiple languages is a powerful addition to the Altova product suite.

■ **About the Author**

Brian R. Barbash is the product review editor for *Web Services Journal*. He is a consultant for the Consulting Group of Computer Sciences Corporation, where he specializes in application architecture and development, business and technical analysis, and Web design.

■ ■ ■ bbarbash@sys-con.com

ALTOVA

FIGURE 1 | **XMLSPY in Visual Studio .NET**

edge 2004 EAST

DEVELOPMENT TECHNOLOGIES EXCHANGE

February 24–26, 2004
Hynes Convention Center, Boston, MA
ARCHITECTING JAVA, .NET, WEB SERVICES, MX, XML, AND OPEN SOURCE

JOIN THE EXCHANGE!

Become a part of the Development Technologies Exchange in Boston this February. Learn from the leading voices in the i-Technology world, see the latest developments from the companies that push the envelope of innovation, and network with the movers and shakers of IT.

The Edge 2004 East Conference Program holds the information you need to excel in today's competitive environment.

Who Should Attend

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

2 Features & Attractions

- 3 Days Packed with Education and Training
- Keynotes & Panel Discussions from Industry Leaders
- FREE Web Services Security Tutorial
- FREE .NET Tutorial "The Smart Client Perspective"
- Industry-Leading Certification Programs
- "Birds of a Feather" Discussions
- Round Table Discussions
- Opening Day Welcome Reception
- Compelling Case Studies & Best Practices
- Hands-On Labs
- Featured Product Demonstrations
- Exhibit Floor Featuring More than 40 Companies and Hundreds of Products
- Real-time SYS-CON Radio Interviews

Register By JANUARY 24, 2004 SAVE Up To $300

VISIT www.sys-con.com/edge FOR TIMES and SCHEDULE

Keynotes

Chris Hjelm
CHIEF TECHNOLOGY OFFICER, ORBITZ
Tuesday, February 24, 2004 – 10:00 a.m.

ORBITZ

Chris Hjelm is chief technology officer for Orbitz and leads the company's technology strategies, engineering, site architecture, and operations. In this role, he is responsible for all aspects of technology, ensuring that Orbitz continues its industry-leading role in providing the lowest cost and most consumer-friendly technology platform in online travel. Before joining Orbitz, Hjelm served as SVP for technology at eBay Inc., overseeing all aspects of its technology. Under his leadership, eBay achieved record levels of site quality and reliability during a period of accelerating global business growth and new product introductions. Chris is also known for his work in creating and advancing world-class technology platforms at high-growth Internet companies as well as in Fortune 500 environments.

David Mendels
GENERAL MANAGER, MACROMEDIA
Wednesday, February 25, 2004 – 10:00 a.m.

macromedia

SOA+RIA=ROI

Service-oriented architectures are imperative to drive systems and data integration, but for the integration to matter it needs to reach users in meaningful ways. Rich Internet applications (RIAs) combine the richness of the desktop with the reach of the web. Bringing web services and rich Internet applications together fulfills the promise and value proposition of service-oriented architectures, as users benefit from actionable, intuitive, and effective data and information. This next generation enterprise application architecture will deliver solutions that capitalize on existing systems and increase return on investment.

David Mendels is general manager for designer and developer products for Macromedia. Since joining Macromedia in 1992 he's played a major role in the development of Macromedia's product and business strategies and in his current role supervises the gamut of Macromedia's designer and developer products, including Flex, Flash, Dreamweaver, and ColdFusion. In that capacity, he has overall responsibility for the business unit and directly manages product marketing, product management, and product development. Mendels began his career at Macromedia in the international department, moving to Japan to lead the company's field operations, sales, and marketing there. After returning to the United States in 1995, Mendels held positions running business development, alliances and, general management of a number of Macromedia product groups.

Dr. Robert Sutor
**DIRECTOR OF WEBSPHERE INFRASTRUCTURE
SOFTWARE, IBM**
Thursday, February 26, 2004 – 10:00 a.m.

IBM

Web Services: Surviving the Mid-Life Crisis

Mid-life can be tough on those who had fast-paced, high-profile youths but suddenly find themselves with dependents and responsibilities put upon them by others. So it is with Web services as it enters the middle phase of its standardization process and moves from being a favorite of the early adopter crowd to being a dependable citizen for the IT mainstream. How is it adapting to the challenge? Have companies like IBM delivered on the early promises of Web services? Do we need to adjust expectations now for Web services so we're not faced with major disappointments later? Dr. Sutor will discuss these and other issues, as well as explain how Web services is a critical part of how companies can become on-demand e-businesses.

Dr. Bob Sutor is IBM's director of WebSphere Infrastructure Software. He is responsible for business and product leadership for the WebSphere Application Server and WebSphere Studio software offerings. Previously, Bob was IBM's director of Web Services Technology, responsible for driving the cross-IBM Web services initiative to continue and advance IBM's leadership in providing Web services solutions, products, and services.

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

DAY 1 — FEBRUARY 24

	Java	.NET	Web Services
8:00	Registration		
9:00	Empowering Java and RSS for Blogging	NET Compact Framework Performance Tips and Tricks	WS-CAF: Standardized Web Services Transactions and Composite Applications
10:00	Opening Keynote – Chris Hjelm, CTO, Orbitz		
11:00	**EXPO OPEN**		
1:00	Keynote Panel Discussion - 21st Century Computing: Is IT Fulfilling its Promise - or Breaking It?		
2:00	Enterprise Architecture & Open Source	Using the Enterprise Instrumentation Framework	Web Services Orchestration, Management, and Security - Will They Play Together?
3:00	Squeezing Java	J2EE/.NET Interoperability	Session TBA
4:00	Opening Night Reception		

DAY 2 — FEBRUARY 25

	Java	.NET	Web Services
8:00	Registration		
9:00	Desktop Java	Best Practices and Techniques for Building Secure ASP.NET Applications	Building Interoperable Web Services
10:00	Keynote Presentation - David Mendels, General Manager of Products, Macromedia		
11:00	**EXPO OPEN**		
1:00	Keynote Panel Discussion - The Open Source Debate		
2:30	Next Phase in Evolution of J2EE	Microsoft Office 2003: A Solutions Platform	Service-Oriented Integration: Making the Right Choices To Support The Next-Generation of Integration
4:00	Aspect Oriented Programming & Java	.NET Framework: Exploring What's New in the Base Class Library for "Whidbey"	Impress the Boss: Roll Your Own Web Services Initiative

DAY 3 — FEBRUARY 26

	Java	.NET	Web Services
8:00	Registration		
9:00	Simplifying J2EE Applications	Session TBA	Government Real-Time Fraud Detection Using Web Services
10:00	Keynote Presentation - "Web Services: Surviving the Mid-Life Crisis", Dr. Robert Sutor, IBM		
11:00	Apache Axis	Introduction to BizTalk Server 2004 and Web Services	Exploring the Dark Side
12:00	**BREAK**		
1:00	ANT/JUnit	BizTalk Server 2004 Technical Drilldown	ID, Please. The Case for Giving Web Services an Identity
3:00	Session TBA	Moving your Architecture to .NET	Session TBA

XML	MX
XML Security Integration Challenges	Enterprise Infrastructure for Rich Internet Applications with Macromedia Flex
SOA Foundation Components: Building an XML Content Router	Code Base RIA's
Real Best Practices for XML Web Services Management and Security	Using Flash with Web Services

Microsoft

Full Day
.NET Tutorial
Presented by
Microsoft
The Smart Client Perspective

XML	MX
XForms - Simplifying the Development of Transactional Web Forms	Using Web Services with ColdFusion
Session TBA	ColdFusion Components
Using XML Schemas Effectively in WSDL Design	Rapidly Build Web Services Applications with ColdFusion and Studio

westbridge

Full Day
Security Tutorial
Presented by
Westbridge
Technology
Strategies for Web Services Security Success

XML	MX
What's New in XSLT 2.0?	Leveraging Web Services
Rich Internet Applications: An XML-Based Approach	Session TBA
Cutting the Crap: Using Rules to Clean Up XML	Code-based Rich Internet Applications with Macromedia Flex
Session TBA	Session TBA

FOR COMPLETE SESSION DETAILS & INFORMATION VISIT
www.sys-con.com/edge

JAVA SESSIONS

Aspect-Oriented Programming & Java
RON BODKIN, NEW ASPECTS OF SOFTWARE

Aspect-Oriented Programming (AOP) has become a hot topic of discussion for enterprise development, with the recent news of support by IBM, JBoss, BEA, Eclipse, and IntelliJ.
- What's real?
- What is AOP?
- What problems can it solve today?
- How can you make an informed decision about using AOP?
- What is the best adoption strategy?
- What are the long-term possibilities for AOP?

This session answers these questions and gives an introduction to AOP for enterprise Java development. AspectJ is a seamless AOP extension to Java. It allows the consistent and flexible implementation of crosscutting concerns such as security, exception handling, testing, logging, caching, and business rules. This is a major improvement on scattered and tangled traditional implementations.

This tutorial introduces AOP and shows how to use AspectJ effectively in an enterprise context, including examples of tools support. Attendees should have experience with OO design and Java development. AspectJ is an open source project available from the Eclipse consortium.

BIO: Ron Bodkin is the founder of New Aspects of Software, which consults and trains on enterprise Java applications with an emphasis on Aspect-Oriented Programming. Ron is also a member of AspectMentor, a consortium of AOP experts. Ron previously worked for the AspectJ group at Xerox PARC, where he led the first AOP implementation projects and training for customers. Prior to that, Ron was founder and CTO at C-bridge, a consultancy that delivered enterprise applications using Java and XML frameworks.

Squeezing Java
ALAN WILLIAMSON, JAVA DEVELOPER'S JOURNAL

Java is a very powerful language, and while it offers the developer a rich array of tools, the fundamentals mustn't be overlooked. Improving your code at the core layer will result in great improvements in efficiency and produce (hopefully) fewer bugs. We'll look at the dos and don'ts of programming and learn lots of hints and tips that will accelerate your Java coding.

BIO: Alan Williamson is SYS-CON Media's technical director, and editor-at-large of *Java Developer's Journal*. In addition, he holds the CTO office at n-ary (consulting) Ltd, one of the first companies in the UK to specialize in Java at the server side.

Enterprise Architecture & Open Source
JAMES MCGOVERN, THE HARTFORD FINANCIAL SERVICES GROUP, INC.

Use of open source software within the enterprise is gaining momentum. The vast majority of organizations use some form of open source software, such as Linux, Apache, and JBoss, in production environments. The enterprise architecture, however, needs to incorporate the best thinking of the industry that not only includes using open source but contributing to it. The model in which open source software gets developed has practices that could assist an organization in becoming agile in their software development practices and

allow them to develop software faster, with lower costs and better quality. In this session you will learn:
- Two models of development: the cathedral and the bazaar
- Value proposition of using open source
- Harnessing the power of the mob: the value proposition of contributing to open source
- Making the build versus buy decision: additional thoughts

BIO: James McGovern is an industry thought leader and the author of the best-selling book, *A Practical Guide to Enterprise Architecture* (Prentice Hall). He is working on two upcoming books entitled *Agile Enterprise Architecture* and *Enterprise SOA*. He is employed as an enterprise architect for The Hartford Financial Services Group, Inc. He holds industry certifications from Microsoft, Cisco, and Sun. He is member of the Java Community Process and of the Worldwide Institute of Software Architects.

J2EE v1.4
BILL ROTH, E.PIPHANY

Day-to-day work with deadlines makes it difficult to keep abreast of the rapidly evolving landscape of J2EE, especially given the numerous constituent J2EE technologies. J2EE v1.4 is chockfull of new services that affect and benefit a wide range of enterprise development tasks. This talk will extract core material from the speaker's new *J2EE Developer's Handbook* and describe what's embodied by J2EE v1.4. The new Web services features provided by J2EE v1.4 in particular will be highlighted. The talk will also briefly address those services missing from the current J2EE standards but still needed when building enterprise applications.

BIO: Bill Roth is senior technical evangelist for E.piphany. With over 15 years in this industry, he has played numerous roles in engineering and product marketing including program manager for Java IDL, and managing product marketing and product management for J2EE 1.2 at Sun.

Apache Axis
CHRIS HADDAD, BURTON GROUP

Apache Axis is the very popular SOAP engine, which includes everything you need to start producing Web services. Discover just what Axis is and how you can utilize the power of this free engine to kick-start your Web services.

BIO: Chris Haddad has 13 years experience in Web services, protocols and standards (SOAP, UDDI, WSDL), application framework, development tools, key development languages (Java, C++, Perl, JavaScript), and enterprise application integration. His background includes Grand Central, Flamenco Networks, Adjoin, Securant, Employease, Jamcracker, TRX, and Raygun.

VISIT **www.sys-con.com/edge** **FOR TIMES** and **SCHEDULE**

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

Empowering Java and RSS for Blogging

JASON BELL

One of the fastest growing areas over the last few years is that of the blogging community. The ease at which you can post and publish information has enabled everyone to become his or her own publisher. One power of blogs has been the syndication of data via the RSS (XML) protocol. Discover how you can easily produce and consume RSS feeds within your Java applications for wider appeal and hook into the likes of JavaBlogs.

BIO: Jason Bell is a Java developer and IT development manager for a B2B portal in the UK. He also contributed to a number of open source projects and is an advocate of everyone reading the API docs.

Ant/JUnit

KYLE GABHART

A defined and easily repeatable process is one of the most necessary but often least-used aspects of good software development. A defined build process ensures that your project's software is built, deployed, and tested identically each time. Without this type of control and predictability, valuable time is often lost chasing down bugs that don't exist or rejecting solutions that were only partially implemented.

A critical measure of the success of software can be found in whether or not it executes successfully. Equally important, however, is whether or not that software does what it was intended to do. JUnit is an open source–testing framework that provides a simple means for developers to define their intentions for how their software should work. JUnit then provides test runners that process your intentions and verify that your code performs as intended. The result is software that not only works, but works in the correct way.

Apache's Ant is a powerful scripting tool that enables developers to define and execute routine software development tasks using the simplicity and extensibility of XML. Ant provides a comprehensive mechanism for managing software development projects, including compilation, deployment, testing, and execution. In addition, it is compatible with any IDE or operating system.

BIO: Kyle Gabhart is an independent consultant, mentor, and published author, specializing in J2EE, XML, and Web services technologies. He is a prolific writer, with his most recent work displayed on IBM's developerWorks Web site in the J2EE Pathfinder column. You can find him on the Web at www.Gabhart.com.

Desktop Java

JOE WINCHESTER, IBM

Java is enjoying a renaissance on the desktop, and there are several reasons for this:

- The issues that plagued early client/server projects or Java desktop applications have largely been solved. Swing 1.4.2 delivered great performance improvements, good fidelity XP, and GTK look-and-feel classes. Java Web Start now exists as a way to deploy to a client PC the programs that run in a local JRE and enjoy the benefits of local caching, lazy update, and execution within Java's security model
- Java is becoming part of the default installation setup for many PC vendors such as Dell, HP, and Gateway.
- SWT provides a set of rich native controls over and above those offered by AWT, and many Java developers are finding that it's a useful GUI toolkit to use in their end-user applications.

FULL DAY JAVA TRAINING

CREATE SERVER-BASED SOLUTIONS USING JAVA 2 ENTERPRISE EDITION

Developing J2EE Applications Using WebSphere Studio Application Developer*

February 25th & February 26th

J2EE is a standard coherent Java API for creating server-based applications, and WebSphere is an industry-leading enterprise application server. Students learn the steps necessary to develop J2EE applications (built using database, servlets, JavaServer Pages, and Enterprise JavaBeans) using WebSphere Studio Application Developer. Hands-on labs reinforce the lecture topics while providing practical experience.

BIO: Ken Greenlee is the chief technology officer of Kenetiks, Inc., a Java-WebSphere training and consulting company he founded in 1997. He has written many courses covering J2SE, J2EE, WebSphere Studio Application Developer, WebSphere Application Server, and WebSphere Portal Server. Prior to founding Kenetiks, he worked for IBM in North Carolina where he worked on development teams responsible for such products as VisualAge Smalltalk and OS/2. He holds a BS degree in computer science from Indiana University along with numerous professional certifications in Java and WebSphere.

*Available for Gold Pass + Training Pass registrants only.

- HTML is no longer being viewed as the only viable client for application development. A number of hybrid technologies that combine the benefits of server-side computing with a client GUI have arisen, including Ultra Lightweight Client, Droplets, JFlash, Macromedia Royale, Asperon, and Thinlets.

This session will look at the Java desktop space and discuss the issues and technologies, as well as what is at stake if Java can or can't recapture its lost pride as a client platform.

BIO: Joe Winchester is a software developer working on WebSphere development tools for IBM in Hursley, UK. Joe was part of the expert group for JSR 57.

VISIT www.sys-con.com/edge FOR TIMES and SCHEDULE

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

.NET SESSIONS

.NET Compact Framework Performance Tips and Tricks
JIM WILSON, JW HEDGEHOG, INC.

Learn the techniques that can be used to increase the responsiveness of user inter-face and network operations for users of applications built on the .NET Compact Framework. Look under the covers at advances and changes in the "Whidbey" release that significantly improve performance. Get a general overview of how the .NET Compact Framework works under the hood at runtime, with specific focus on performance implications. Next, we cover general user interface tips to increase per-formance. Explore how asynchronous infrastructure, such as threading, in the .NET Compact Framework can be leveraged to optimize both user interface and network operations. Learn about the architectural guidelines for creating applications that perform well under frequently changing network conditions.

BIO: Jim Wilson is president and cofounder of JW Hedgehog, Inc., a consulting firm specializing in the integration of smart devices as part of enterprise solutions with a focus on the .NET Compact Framework, .NET Framework, Web services, and other XML-related technologies. He develops and teaches DevelopMentor's Mobility training course, was one of Microsoft's original five MVPs for the .NET Compact Framework, has written articles for numerous pub-lications, and has spoken at a myriad of events, including Microsoft's Tech-Ed and Mobility Developers conferences.

Best Practices and Techniques for Building Secure ASP.NET Applications
PATRICK HYNDS, CRITICALSITES

When the enterprise depends on your application, careful attention to security is essential. This session provides specific recommendations to follow when develop-ing secure ASP.NET Web applications and services, and focuses on the details of configuring IIS for security. Understand how to use authentication, authorization, threat modeling, configuration settings, and secure database access to create secure systems, plus learn common coding techniques for storing secrets, error handling, data validation, and code access security.

BIO: Patrick Hynds, MCSD, MCSE+I, MCDBA, MCSA, MCP+Site Builder, MCT, is the Microsoft Regional Director for Boston and the CTO for CriticalSites. Named by Microsoft as the Regional Director for Boston, he has been recognized as a leader in the technology field. An expert on Microsoft technology (with at last count 54 Microsoft certifications) and experienced with other technologies as well (WebSphere, Sybase, Perl, Java, Unix, Netware, C++, etc.), Patrick previously taught freelance software development and network architecture. He has been a successful contractor who enjoyed mastering difficult troubleshooting assignments. A graduate of West Point and a Gulf War veteran, Patrick brings an uncommon level of dedication to his leadership role at CriticalSites. He has experi-ence in addressing business challenges with blended IT solutions involving leading-edge database, Web, and hardware systems. In spite of the demands of his man-agement role at CriticalSites, Patrick stays technical and in the trenches, acting as project manager and/or developer/engineer on selected projects throughout the year.

Using the Enterprise Instrumentation Framework
DEREK FERGUSON, EXPAND BEYOND CORPORATION

The Microsoft .NET Framework 1.1 and Windows Server 2003 offer a number of new features to help developers instrument their code. In this session, you'll learn about the challenges facing application management in today's distributed world. We will examine the new unified instrumentation API in the Enterprise Instrumentation Framework (EIF), including the new Windows Event Trace available in Windows Server 2003, configurable at-source event filtering, and how request-based event tracing using EIF allows you to put a request context around the trace messages that map to a business process flow in your application. We will also dis-cuss the benefits to both the developer and the application administrator for using EIF in your application.

BIO: Derek Ferguson is chief technology evangelist for Expand Beyond Corporation (www.xb.com), the worldwide leader in mobile software for enterprise management. He is also editor-in-chief of *.NET Developer's Journal* and author of the book Mobile .NET (Apress).

.NET Framework: Exploring What's New in the Base Class Library for "Whidbey"
JULIA LERMAN

The base classes serve as the essential libraries for any developer. Continued evolution of the base classes provides numerous benefits, including the ability to write more reliable, faster solutions, easier-to-write code, and more solutions entirely in managed code. Take a look at the many features that are a part of that evolution, including features in IO, event-logging, and various features in System.

BIO: Julia Lerman is the owner of Julia Lerman, Inc (dba The Data Farm). and has been developing software applications and consult-ing to businesses since 1984. Julia is the founder and leader of VTdotNet, Vermont's .NET user group (www.vtdotnet.org), as well as a mem-ber of the User Group Relations Committee for INETA (International .NET Association) www.ineta.org. She writes on Visual Studios and .NET and in her pre-Visual Basic days, FoxPro Advisor. Having spent many years as a FoxPro. Julia is working very hard to help bring the same community spirit to .NET that is so unique to FoxPro.

Microsoft Office 2003: A Solutions Platform
JOHN HOLLINGER, INTERNOSIS

For all developers who would like to integrate custom business solutions with Microsoft Office products, this session will introduce you to the expanded developer features that have been included in the newest version of Microsoft Office. Come explore new XML-based programmability in everything from Word 2003 and Excel 2003 to FrontPage 2003 and SharePoint. Build powerful, modular solutions with Web services. Learn about InfoPath 2003 support for XML standards. Discover how to use the Microsoft Visual Studio Tools for the Microsoft Office System to automate and extend Microsoft Office Word 2003 and Microsoft Office Excel 2003 using Visual Basic .NET and Visual C# .NET. More than ever, Office has a solution for you.

BIO: John V. Hollinger serves as CTO of Internosis clients are able to leverage the latest technologies while planning for tomorrow's challenges. Mr. Hollinger is responsible for the performance of the Northeast/Commercial business unit, including all aspects of business development, client care, marketing, resource management, and financial management. Prior to joining Internosis, he played an instrumental role in the creation of New Technology Partners (NTP) and then went on to run a highly successful division of Xpedior, where his team was a four-time Microsoft Solution Provider Partner of the Year in New England and the recipient of the award on a worldwide basis in 1997. Previously, John was vice president of Advanced Technologies at AIG.

VISIT www.sys-con.com/edge FOR TIMES and SCHEDULE

Advanced BizTalk Server 2004
BRUCE BACKA, CRITICALSITES

BizTalk Server 2004 is designed to provide Enterprise Application Integration (EAI), Business Process Automation (BPA), and Information Worker Integration solutions. Come join us for a technical drilldown on the new features and toolsets available.

BIO: Bruce Backa is the founder of CriticalSites. A noted business leader and consultant in the IT industry, he has acted as chief architect, technologist, and project manager for assignments involving large-scale technology and implementation strategies. He has held the positions of director of Technology and Business Research for the American Stock Exchange (AMEX) and director of Technology for American International Group. Bruce has been responsible for the architecture, implementation, and management of a worldwide client/server networking infrastructure for a Fortune 10 company with a platform of over 600 servers connecting 10,000 users across 50 cities throughout North America and offshore. In 1994 he founded NTP Software, a provider of business solutions for Windows NT and Windows 2000. At the recent World Economic Forum in Switzerland, Bruce was recognized as a technology pioneer. This follows a similar award from the National Computer Conference in 1974, where he was honored as a part of the Dartmouth College team that invented computer timesharing.

Moving your Architecture to .NET
CARL FRANKLIN, FRANKLINS.NET

This session is focus on how to migrate existing business components from VB6 COM Objects to VB.NET assembly components. We'll spend time discovering how to best move different tiers of a multi-tiered application from COM to .NET, as well as effective strategies on how to wrap existing COM components for interoperability. We'll also examine best practices for moving your application from a COM-based architecture to a .NET-based architecture.

BIO: Carl Franklin has been a luminary in the VB community since the very early days. Since then he has written extensively on the subject of VB. He has authored two books for John Wiley & Sons on sockets programming in VB, and in 1994 he helped create the very first Web site for VB developers, Carl & Gary's VB Home Page. He now teaches hands-on VB.NET classes for his company, Franklins.NET. He has taught developers from Citigroup, Aetna, Fidelity Investments, Fleet Bank, Foxwoods Casino, UTC, Hubbell, Microsoft, Mohegan Sun Casino, and Northeast Utilities, to name a few. Carl is cohost of a weekly talk show on his Web site for .NET programmers, called ".NET Rocks!" Carl is MSDN Regional Director for Connecticut.

Introduction to BizTalk Server 2004
BRIAN LOESGEN, NEUDESIC

BizTalk Server 2004, the latest and third version of BizTalk, is an exciting tool that all .NET application developers should know about. BizTalk fills two key business needs: enterprise application integration and business process automation. The powerful messaging platform allows for easy integration between applications, and also between enterprises. The business process automation engine, based on the new BPEL XML specification, allows for automation of even the most complex business processes. Throw in InfoPath for a user interface into BizTalk, tight integration with Visual Studio, Sharepoint, and Office-based monitoring and reporting tools, and you get an incredibly powerful tool.

This session will be an overview of BizTalk Server 2004. Concepts and tools will be demonstrated by walking through several real-world, practical examples of BizTalk usage. Particular attention will be paid to Web service integration and orchestration,

showing why BizTalk Server 2004 deserves consideration as the integration "glue" in a services-oriented architecture.

BIO: Based in San Diego, Brian Loesgen is a principal consultant with Neudesic, a premier southern California solution provider and system integrator. At Neudesic, Brian uses his expertise to architect and deliver advanced Microsoft technology solutions, and to translate new, leading-edge technologies into real-world value. He is a coauthor of the *Professional XML*, *Professional ASP/XML*, *Professional Windows DNA*, *Professional ASP.NET Web Services*, and *Professional VB.NET Web Services* books from Wrox, and is currently working on the *BizTalk Server 2004 Developers Guide*, due in 2004. In addition, Brian has written technical white papers for Intel, Microsoft, and others. Brian has spoken at numerous major technical conferences worldwide and is a cofounder and president of the International .NET Association (ineta.org). He coleads the San Diego .NET user group, and leads the San Diego Software Industry Council Web Services SIG.

In his spare moments, Brian enjoys outdoor activities such as cycling, hiking in the mountains, kayaking, camping in the desert, or going to the beach with his wife Miriam and children Steven and Melissa.
Brian can be reached at brian.loesgen@ineta.org.

VISIT www.sys-con.com/edge **FOR TIMES** and **SCHEDULE**

10

WEB SERVICES SESSIONS

WS-CAF: Standardized Web Services Transactions and Composite Applications
ERIC NEWCOMER, IONA TECHNOLOGIES

The Web Services Composite Application Framework is a collection of three specifications – Web Service Context (WS-CTX), Web Service Coordination Framework (WS-CF), and Web Service Transaction Management (WS-TXM) – designed to solve problems that arise when multiple Web services are used in combination ("composite applications") to support information sharing and transaction processing. As coauthor of the specification, Eric will discuss how WS-CAF addresses the underlying issues of Web service context propagation and transaction management to expand the scope, usability, and reliability of Web services for business process automation.

BIO: In the role of chief technology officer at IONA, Eric 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.

Government Real-Time Fraud Detection Using Web Services
NEIL MCGOVERN, SYBASE, INC.

Government agencies are faced with increasing amounts of data and are challenged to make sense of, and act on, that data in real time. Failure to interpret and execute on data can result in security threats and, potentially, loss of life. Government agencies are increasingly investing in Web services solutions to address their need for real-time access to information.

The Canadian Passport Office is an example of a government agency leveraging Web services to exchange information in real time to combat terrorism and other illicit uses of fraudulently obtained passports. The agency selected IT consulting firm Pentelar and Sybase, Inc., technologies to electronically authenticate identity document data through the use of Web services and ebXML. Identification tasks performed across 10 provinces and 3 territories previously handled by physical document inspection are now conducted in real time with absolute security and reliability.

In this session, Sybase and Pentelar will discuss this successful pilot project and highlight the ebXML capabilities that enable the Canadian Passport Office to address real-time information exchange.

BIO: Neil McGovern has over 15 years of experience in the software industry. Currently, he is responsible for platform strategy for Sybase's infrastructure product line, including its enterprise-class RDBMS; Sybase ASE; and integration, portal, and tools product lines. Prior to that, Neil was responsible for building the Sybase (New Era of Networks) Adapter suites, commonly recognized as one of the most advanced and complete sets of adapters in the EAI marketplace. Neil developed ERP package expertise as CTO of Convoy Corporation, which specialized in PeopleSoft integration. He led the Development and Professional Services teams for Convoy Corporation, which was purchased by Sybase (New Era of Networks) in 1999 after recording triple-digit growth for three consecutive years.

Service-Oriented Integration: Making the Right Choices To Support The Next-Generation of Integration
DAVID CHAPPELL, SONIC SOFTWARE

Applications are increasingly being developed "built-to-integrate," providing the ability to easily expose key functionality through commonly defined interfaces. Gartner calls this concept SODA, or service-oriented development of applications, fitting into its overall service-oriented architecture landscape. When applied to the ever-present integration challenge, SODA represents a transition to service-oriented integration.

But making the right architectural decisions is absolutely vital to ensuring success with service-oriented integration projects – whether applications were built to integrate or not. Choices at all levels – from application-interface style to overall system architecture – can seriously affect the long-term value derived from integration projects. Application infrastructure products on the market today embody architectural directions that should be carefully examined before making purchase decisions. Any two products that claim to support a "service-oriented architecture" may, in fact, promote radically different architectures.

This presentation will examine the leading choices for supporting service-oriented integration: enterprise service buses (ESBs), integration brokers, and application suite platforms.

BIO: Dave Chappell, VP and chief technology evangelist for Sonic Software, has over 18 years of experience in the software industry. Dave and Sonic Software are involved in many standards organizations, technical committees, and expert groups, including the W3C, OASIS, WS-I, and the Java Community Process. Dave is an active contributor to the Apache Axis project and SOAPBuilders.

Dave also has extensive experience in distributed computing, including message-oriented middleware, CORBA, COM, and Web application server infrastructure. In 2002, Dave received the "Most Outstanding Individual Contributor to the Java Community" award from the editors of JavaPro Magazine.

Securing the Web: What Can Be Done Today
MARK SECRIST, HEWLETT-PACKARD CO.

Security is considered one of the main barriers to the adoption of Web services today. With the proliferation of emerging security standards, there is a lot of confusion over which ones are mature enough to use and how they might fit together. This session will present cover current and emerging security standards for Web services and show how they can be fit together architecturally to address various security concerns. At the conclusion, the audience will have a better understanding of these standards and how to begin securing their own Web services.

BIO: Mark Secrist is a senior software consultant for HP's Developer Resources Organization, with over 15 years of experience working in the software development industry. He provides technical consulting to HP and its customers in the areas of J2EE and Web services development, as well as development strategies for "management-ready" applications. In this capacity, he has written a number of white papers and trade publication articles, presented at a number of industry events, and provided training and consulting to many of our enterprise customers.

VISIT www.sys-con.com/edge **FOR TIMES** and **SCHEDULE**

Exploring the Dark Side
PETER VARHOL, COMPUWARE CORPORATION

The growing use of services-oriented architectures puts pressure on application developers relying on Web services for key features of their applications. Performance, scalability, and reliability of these components affect the ability of applications to meet service-level agreements, yet can't easily be analyzed as a part of the application when developers have a problem. In fact, the Web service may be on a different software platform than the rest of the application. This session describes how developers can shed light on memory use in Web services written in either .NET or Java, even if they didn't write the code and wrote their own applications on the other platform.

BIO: Peter Varhol is a product manager at Compuware Corporation, where he provides direction for developer tools targeting enterprise applications and Web services. He holds graduate degrees in computer science and mathematics.

Building Interoperable Web Services Using WS-I Basic Profile
KEVIN LIU, SAP LABS, LLC

The promise of Web services interoperability is based on a set of standards, including XSD, SOAP, WSDL, and UDDI. However, early interoperability experiments such as the SOAPBuilder activity has proven that making these technologies seamlessly work together, inter- or intra-enterprise, is a bigger challenge than most of us expected. Most of the Web services technologies are still in the standardization process, and Web services platforms have to be based on pre-standard draft specifications. The inconsistencies and ambiguities left in these initial drafts can easily lead to different interpretations for the same features and substantially compromise the promised interoperability. This presentation explains how WS-I Basic Profile addresses these challenges, and what are the key advises you can get from the profile that can make your Web services interoperable. It explains why the profile disallows SOAP encoding and why RPC style should give way to document style.

BIO: Kevin Liu is an architect of XML Web services technologies at SAP Labs, California. Kevin helps to drive the adoption of Web services technologies in SAP's strategic products, as well as representing SAP in various standards bodies. He has over 10 years of experience in software engineering and in financial industries and holds multiple master's degrees in information management, finance, and economics.

ID, Please. The Case for Giving Web Services an Identity
ASHISH LARIVEE, NOVELL

Without identity management, Web services can be consumed by anyone. The challenge for Web services developers is to provide appropriate access based on the user's identity. As identity management moves into the forefront of technology, directory services will evolve from simple LDAP repositories used for authentication and storage to robust engines that provide identity integration, access management, and policy enforcement. This presentation will discuss how identity management and directory services provide a robust solution for Web services authentication, authorization, and single sign-on.

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.

Web Services Orchestration, Management, and Security - Will They Play Together?
PAUL LIPTON, COMPUTER ASSOCIATES

Web Services orchestration, management, and security are among the principal challenges facing implementers of service-oriented architectures today. There is still much confusion in the IT community about the standards themselves, which are at various stages of maturity. Also, their relevance to enterprise IT and how they might someday be able to effectively work together is often unclear. This session provides a useful overview of standards in these three critical areas of Web services – orchestration/choreography, management, and security – and more importantly, how each affects the other. Attendees will gain practical knowledge and a deeper understanding of future trends and the need to address certain real-world issues in order to create a more cost-effective and agile IT infrastructure.

BIO: Paul Lipton is the Web services technology leader for the field services organization and a technology strategist in the Office of the CTO at Computer Associates. He has been an architect and developer of enterprise systems for more than 20 years, and has worked closely with key CA customers to architect distributed solutions using J2EE, .NET, wireless, and Web services technology. Paul has represented CA in various standards organizations, and has participated in the Java Community Process. He has published magazine articles on many technologies including Web services, Java, .NET, EAI, wireless technology, and distributed systems.

Impress the Boss: Roll Your Own Web Services Initiative
BOB ZUREK, ASCENTIAL SOFTWARE

Some companies are leveraging open source software like Linux, JBoss, PHP, MySQL, and Apache SOAP to develop and deploy Web services. As this open source Web services stack continues to mature, companies will benefit beyond simply saving money on license fees; they will be able to assemble the stack and get a Web service up and running quickly as well.

Developers frustrated by their enterprise's lack of Web services interest can immediately master and build Web services at home with open source stack components with little investment except time and a broadband connection. Imagine impressing your company with a demo of wrapping that chunk of code as an open source Web service and calling it from a legacy Visual Basic application.

BIO: Bob Zurek, VP of Advanced Technology, Ascential Software, is responsible for Ascential's overall product strategy. He is instrumental in developing and driving its enterprise integration strategy, including its parallel processing framework, data quality, and Web services strategies. Previously, he was a senior analyst with Forrester Research; prior to Forrester, he held senior management positions at LumaPath, Centive Systems, and Sybase.

VISIT www.sys-con.com/edge **FOR TIMES** and **SCHEDULE**

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

XML SESSIONS

Real Best Practices for XML Web Services Management and Security
KERRY CHAMPION, WESTBRIDGE TECHNOLOGY

Companies deploying Web services in a meaningful way are increasingly finding they need to address Web services management and security early in the architectural phase. Basic Web services connections are easy to do, but managing the security, performance, scalability, and the inevitable changes to the production environment requires some knowledge, expertise, and planning. This session cuts through the hype and outlines real-world mistakes many companies make when deploying Web services and the real best practices from companies who have successfully captured the value of XML Web services. It provides practical advice on how to successfully manage and secure your XML Web service environment.

BIO: Kerry Champion is president and founder of Westbridge Technology, a leading provider of XML Web services management and security solutions. A frequent writer and speaker on XML Web services, Kerry is a renowned technologist who has served in various senior technical and product strategy roles at enterprise security and messaging companies.

SOA Foundation Components: Building an XML Content Router
DAN STIEGLITZ, TIAA-CREF

One of the fundamental components for any burgeoning SOA will be an XML content router. This session explores the concepts, patterns, and open source software available that facilitate building an XML content routing system. The system can be exposed as a Web service or simply as a stand-alone J2EE component for use in your enterprise. The "restaurant" pattern is introduced as the principal design pattern for building the service, and this pattern's applicability to building generic services is discussed. Applying the router as an XML data integration tool is also discussed, as well as its potential for acting as a service orchestrator.

BIO: Visit www.sys-con.com/edge for complete speaker information.

What's New in XSLT 2.0?
STEVE HECKLER, ACCELEBRATE

XSLT 2.0, which may achieve W3C Recommendation status by conference time, offers unparalleled power in conjunction with XPath 2.0 for transforming XML documents. In this engaging, example-rich seminar, Steve Heckler demonstrates the most important new features of XSLT 2.0, including sequences, new data types and XML Schema support, regular expressions, multiple document output, grouping, new control-flow operators, and much more. Current/future support for XSLT 2.0 on the Java and .NET platforms will also be discussed. Most examples will use Saxon, but .NET examples will be included if .NET supports XSLT 2.0 by conference time.

BIO: Steve Heckler has been a leading IT trainer and application developer for nearly a decade. Prior to founding Accelerate in March 2002, he served nearly seven years as vice president and then president of WestLake Internet Training. He holds bachelor's and master's degrees from Stanford University.

Using XML Schemas Effectively in WSDL Design
CHRISTOPHER PELTZ, HP

Developers building Web services today are beginning to see the value of using the document-style approach over RPC. Recent experience shows that to take full advantage of document style Web services requires a strong knowledge of XML Schemas and related XML standards. This session presents a number of important tips and techniques for properly using XML Schemas in the design of a Web services interface, including important XML-based development tools, binding considerations between XML and underlying objects, WSDL reusability through XML Schemas, and XML Schema naming best practices.

BIO: Chris Peltz is a senior software consultant within HP's Developer Resources Organization. He provides technical and architectural consulting to enterprise customers in the areas of J2EE, Web services, and mobile development. Chris has over 10 years of software experience in object-oriented technologies, 4GL development, GIS, and Web applications design.

Using Rules to Clean Up XML
GARY BRUNELL, PARASOFT

Garbage in, garbage out – it's an axiom that applies to many aspects of enterprise development, but none more so than building reliable and robust Web applications and integration projects with XML. Since its inception, XML has been seen as the cure-all for problems related to Web applications and integration projects. However, poorly written XML can slow down an integration project, or worse, cause the integration project to collapse. The key to successfully using XML in an integration project is to first understand the inefficiencies that may cause poorly written XML, and then apply a rule-based system that establishes policies to follow.

BIO: Gary Brunell joined Parasoft in the fall of 2001 to develop and head the company's professional services division. He is responsible for spearheading delivery services, technical support, and training initiatives as well as establishing process-improvement infrastructure.

XML: Getting Started with Minimum Investment
KETAN PATEL, DATAWATCH

Many organizations are evaluating XML as an enabler for sharing and presenting information. Whether part of a portal strategy, an alternative to EDI for billing, or a data conduit to an information repository, XML is core to many future projects. As IT managers evaluate XML's potential, they should understand its capabilities and limitations. XML is not the panacea for all technology issues. However, as an enabler, XML can provide a common language for sharing information inside and outside your organization. The key is to use the technology in projects that have a clear return and minimize the investments required.

BIO: Datawatch director, Technical Strategy, Ketan Patel has over 15 years of experience developing and marketing technology products to customers who are cautious about technology adoption. During his career, Ketan has focused on bridging the gap between technology and business, and is passionate about bringing information technology solutions to market.

VISIT www.sys-con.com/edge **FOR TIMES** and **SCHEDULE**

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

XForms - Simplifying the Development of Transactional Web Forms

ASHISH LARIVEE, NOVELL

XForms is a W3C specification that specifies a declarative language for solving a common requirement for advanced user interaction, data validation, and XML processing. XForms is designed to be integrated into XHTML, but is not restricted to being a part of that language alone. It can be integrated into any suitable markup language. This session gives an introduction to XForms and explains how XForms in the client tier of the J2EE application architecture. In addition, it will cover the benefits of XForms and why it is a perfect fit for interacting with J2EE and Web services. A demonstration of XForms in a J2EE environment using an XForms- compliant browser and a sample application further illustrates the advantages.

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 the J2EE platform. 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.

Rich Internet Applications: An XML-Based Approach

CHRISTOPHE COENRAETS, LASZLO SYSTEMS

Rich Internet applications deliver groundbreaking, highly interactive user interfaces that provide a user experience similar and often superior to desktop applications. In this session, we will discuss the benefits of rich Internet applications, review the market landscape, and explore an XML-centric approach for developing rich Internet applications that can be delivered to any browser today. As a practical example, we will demonstrate how to build a rich Internet application version of an Amazon store using an XML-based presentation server to deliver the rich user interface, and the Amazon Web services API to deliver the content.

BIO: Christophe Coenraets worked at Sybase/Powersoft from 1994 to 2000. He started working with Java in 1996 and became the technical evangelist for the company's Java and Internet Application Division. Christophe then joined Macromedia as the company's JRun and J2EE technical evangelist. He joined Laszlo Systems early in 2003.

XML Security Integration Challenges

PHIL STEITZ, AMERICAN EXPRESS

This session will highlight the key security integration challenges faced by application teams developing enterprise solutions using loosely coupled, XML-based interfaces and service-oriented architectures. We will discuss how emerging standards and technologies address the practical security problems faced by development teams, the gaps that still remain, and the tradeoffs and compromises that architects and developers need to make to implement secure solutions today.

BIO: Phil Steitz is vice president, e-commerce applications development at American Express. Phil has over 20 years of experience as a developer, architect, and technology leader involved in distributed systems development. Before joining American Express, Phil served as a middleware architecture consultant, designing large-scale distributed systems for enterprise customers. Phil holds a PhD in mathematics from the University of Maryland.

Attend a FREE One Day Security Tutorial
Presented By

westbridge

Strategies for Web Services Security Success
Wednesday, February 25, 2004

Are you a developer, software architect, IT operator, or security administrator deploying or planning to deploy XML Web services? If so, this technical seminar is designed to arm you with the practical information and best practices you need to securely deploy XML Web services in your environment. Many questions will be addressed, such as: Why do Web services need special security? What standards are being created and what do they address? How do I leverage my existing environment to secure Web services? What are the different architectural and technical approaches to solving the problem? How can I centrally manage security in a decentralized environment?

Course Highlights/Benefits
- Overview of XML Web services security: Why is it important?
- Discussion of various standards (WS-security, SAML, XML-Enc, XML-Sig, XKMS)
- Architectural considerations
- Malicious Web services attacks
- Strategies for securing XML Web services today and in the future
- "Nuts and bolts" demonstrations of security solutions

As part of the tutorial, we will show you how to secure your entire XML environment without adding any additional code.

Course leaders

Dan Neiman is a Senior Systems Engineer at Westbridge Technology and is responsible for training and working with real world customers in solving their security and management needs for XML and Web Services. Previously, Dan worked for Tumbleweed Communications where he directed the field engineers in designing and implementing secure email transports. He has also held positions at CSC and the Department of Defense. Dan holds a BS degree in Management and Decision Systems from Carnegie Mellon University.

Peter Bostrom is a Federal Technology Director at Westbridge Technology and works closely with federal customers. Previously, Peter was VP of Product Management at Sequation, a security solutions provider and CTO and co-founder of ExactOne, an Internet-scale distributed search engine. Peter also worked in a variety of sales, consulting and product management roles at Legent Corporation. He was awarded Systems Engineer of the Year award in his first year there. Peter can be heard on National Public Radio's WAMU in Washington, DC as a regular guest on iPublic Interesti hosted by Kojo Nnamdi and is president of the DC Association of Internet Professionals. Peter holds a BA in International Affairs from The American University after spending three years as an Army Ranger.

13

VISIT www.sys-con.com/edge **FOR TIMES** and **SCHEDULE**

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

Enterprise Infrastructure for Rich Internet Applications with Macromedia Flex

KEVIN HOYT, MACROMEDIA

Learn how Macromedia's technology initiative, Flex fits seamlessly into today's new service-oriented architectures (SOA). We'll cover design patterns for rich clients, accessing web services and securing your Flex application.

BIO: As a senior sales engineer for Macromedia, Kevin Hoyt plays a primary role in evangelizing the product line, educating the customer base, and supporting the sales organization. Kevin is also a regular speaker at various user group forums throughout the United States.

Leveraging Web Services

TOM JORDHAL, MACROMEDIA

Web Services technology is changing the way we think about designing and building applications. Come and learn what all the fuss is about, find out exactly which problems web services solve, see web services created and used, and even discover how web services expose the world of .NET.

BIO: Tom has been a Principal Engineer at Allaire/Macromedia for over 6 years and has spent the last 2 years on the front lines of Web Services. He is one of the implementers of Apache Axis and is the Macromedia representative on the W3C Web Service Description WSDL 1.2 working group. Before getting involved in Web Services, he was the technical lead for the ColdFusion on UNIX products. Tom is currently 'back home' working on ColdFusion.

ColdFusion Components

GLENDA VIGOREAUX, BUSINESS SYSTEMS CONSULTANTS, INC.

ColdFusion Components combine the power of objects with the simplicity of CFML. This is the way object-based development was intended to be, and in this session, you'll learn about this combination first hand. Starting with a simple data-driven application, you'll gradually convert it into a highly scalable and manageable multitier application, and in the process, will amaze yourself at just how easy ColdFusion makes this process.

BIO: Glenda Vigoreaux has been a Certified Macromedia Instructor and Developer for the last 5 years. She is currently heading the web development division for Business Systems Consultants, Inc. the most successful reseller and services provider of software for associations in the greater Chicago area.

Rapidly Build Web Services Applications with ColdFusion and Studio

SIMON HORWITH, ETRILOGY

The last year has shown that web services are not just another passing fad and their promise of platform-independent distributed applications has been realized. Compared to other application server platforms, ColdFusion makes creating web services easy. This session covers how to create a ColdFusion Component (CFC) in Dreamweaver, as well as how to expose that CFC as a web service by just toggling one attribute of the CFC. That's right: in ColdFusion, it is just that easy.

BIO: Simon Horwith is Chief Technology Officer of eTRILOGY Ltd., a software development company based in London, England. Simon has been using ColdFusion since version 1.5 and is a member of Team Macromedia. He is a Macromedia Certified Advanced ColdFusion and Flash D eveloper and is a Macromedia Certified I nstructor. In addition to administering the CFDJ-List mail list and presenting at CFUGs and conferences around the world, Simon has also been a contributing author of several books and technical papers.

Code-based Rich Internet Applications with Macromedia Flex

CHRISTOPHE COENRAETS, MACROMEDIA

Learn how to use Macromedia's technology initiative Flex to create rich internet applications. This session will cover using components, layouts, and managers to build user interfaces as well as using Flexs XML-based language to create and manipulate client-side data models.

BIO: Please visit www.sys-con.com/edge for a complete speaker bio.

Using Macromedia Flash with Web Services

Web services, a technology that allows developers to execute remote procedures, are emerging as a revolutionary tool for web application development. Macromedia Flash MX 2004 Professional is a powerful tool for building applications that consume web services built in any technology, including Macromedia ColdFusion, Java, ASP.NET and PHP. In this session you will explore the visionary computing model that web services represent as you use Macromedia Flash components to develop a web service based application. You will learn how to discover web services, work with data and UI components, perform data binding, examine security issues and aggregate multiple web services into a cutting edge web service consumer.

For a Complete List of MX Sessions please visit www.sys-con.com/edge

RESERVE YOUR HOTEL ROOM NOW AT THE SHERATON BOSTON HOTEL!
THE OFFICIAL CONFERENCE HOTEL OF EDGE 2004 EAST

THE SHERATON BOSTON HOTEL
PRUDENTIAL CENTER
39 DALTON STREET
BOSTON, MA 02199

Arrangements have been made with the Sheraton Boston Hotel, which is conveniently located adjacent to the Hynes Convention Center. Specially reduced rates have been secured at this luxury, full-service hotel.

To learn more about The Sheraton Boston Hotel you can contact the hotel directly or you can make your reservations by calling Expo Travel International at (800) 829-2281 or (201) 444-0060

(direct). Fax reservations to (201) 444-0062. Credit card information is required to guarantee reservations and expedite confirmation. Confirmations will be mailed directly from the hotel.

To make online reservations visit www.expotravel.com

CONTACT INFORMATION:
The Boston Sheraton Hotel Reservations:
Phone (617) 236-2000
Fax (617) 236-1702

HOTEL ARRANGEMENTS ARE EASIER THAN EVER!
You have your choice –
contact the hotel directly or call us.

The Official Conference travel agent, Expo Travel International.
Official Conference Travel Agent:
Expo Travel International
Toll Free: (800) 829-2281
Tel: (201) 444-0060
Fax reservations to (201) 444-0062

DRIVING DIRECTIONS TO THE SHERATON BOSTON FROM LOGAN INTERNATIONAL AIRPORT
Take the Sumner Tunnel in the right lane to Storrow Drive (Exit 26). Follow for 2 miles and then take the Fenway Exit (left exit). Bear left following "Boylston Street Inbound" and at the fifth light make a right onto Dalton Street.

VISIT www.sys-con.com/edge FOR TIMES and SCHEDULE

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

SPECIAL INSERT:
Edge 2004 East Conference & Expo February 24 – 26, 2004

CONFERENCE: Feb. 24 – 26, 2004   **EXPO:** Feb. 24 – 25, 2004

Hynes 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" 201-782-9651
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.

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 12/19/04	Before 1/24/04	Before 2/21/04	Onsite
☐ **GP: Gold Passport Three-Day Conference** Good for all three days of the conference program, including keynotes, all conference sessions & one day of J2EE training.	$1,395.00	$1,495.00	$1,695.00	$1,795.00
☐ **3D: Three Day Conference** Good for all three days of the conference program, including keynotes, all conference sessions (excludes J2EE training).	$1,295.00	$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 (excludes J2EE training).	$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 (excludes J2EE training).	$595.00	$695.00	$695.00	$795.00
☐ **TRAINING PASS:** Good for Developing J2EE Applications Java Training Session only. Select one: ☐ February 25th ☐ February 26th	$495.00	$595.00	$595.00	$695.00
☐ **VIP PASS FREE with preregistration** Select one: ☐ FREE - The Smart Client Perspective (Feb. 24) ☐ FREE - Strategies for Web Services Security Success (Feb. 25)	FREE	FREE	FREE	$99.00
☐ **EO: Expo Hall Only** Includes keynotes, expo presentations, and workshops.	FREE	FREE	FREE	$50.00

SPECIAL DISCOUNTS AVAILABLE
Take advantage of the Early Bird and Preregistration values available right now, or save even more with a group of 5 or more. For special group discounts contact Michael Lynch at mike@sys-con.com, or by phone at (201) 802-3058.

4

A. Your Job Title
☐ CTO, CIO, VP, Chief Architect
☐ Software Development Director/Manager/Evangelist
☐ IT Director/Manager
☐ Project Manager/Project Leader/Group Leader
☐ Software Architect/Systems Analyst
☐ Application Programmer/Evangelist
☐ Database Administrator/Programmer
☐ Software Developer/Systems Integrator/Consultant
☐ Web Programmer
☐ CEO/COO/President/Chairman/Owner/Partner
☐ VP/Director/Manager Marketing, Sales
☐ VP/Director/Manager of Product Development
☐ General Division Manager/Department Manager
☐ Other (please specify)_____

B. Business/Industry
☐ Computer Software
☐ Computer Hardware and Electronics
☐ Computer Networking & Telecommunications
☐ Internet/Web/E-commerce
☐ Consulting & Systems Integrator
☐ Financial Services
☐ Manufacturing
☐ Wholesale/Retail/Distribution
☐ Transportation
☐ Travel/Hospitality
☐ Government/Military/Aerospace
☐ Health Care/Medical
☐ Insurance/Legal
☐ Education
☐ Utilities
☐ Architecture/Construction/Real Estate
☐ Agriculture
☐ Nonprofit/Religious
☐ Other (please specify)
_____

C. Total number of employees at your location and entire organization (check all 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 communications and computer 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

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

F. Do you recommend, specify, evaluate, approve or purchase wireless 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

If you require special assistance covered under the Americans with Disabilities Act, please call 201-802-3058 by February 10, 2004.

CANCELLATIONS, SUBSTITUTIONS, REFUNDS
Fax written request to SYS-CON Registration 201-782-9651. Requests for refunds received prior to January 16, 2004 will be honored less a 10% handling charge; requests received after January 16, 2004 and before February 6, 2004, will be honored less a 20% handling charge. No requests for refunds will be honored after February 6, 2004. Requests for substitutions must be made in writing prior to February 20, 2004. 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.

SPONSORS & FEATURED EXHIBITORS

Computer Associates
Gold Sponsor

Computer Associates International, Inc. (CA) offers the most comprehensive portfolio of solutions for managing Web services and Web services platforms. CA's end-to-end solutions, including Unicenter® Web Services Distributed Management, dramatically reduce the cost and complexity of Web services deployments by providing visibility and control, optimized performance and increased enterprise responsiveness. For more information, please visit ca.com/webservices.

EAI Industry Consortium
Association Sponsor

The EAI Industry Consortium, a non-profit global advocacy group, developed to promote Enterprise Application Integration through sponsored research, the establishment of standards and guidelines, best practices and articulation of strategic and measurable benefits. The member driven consortium, designed as an EAI information hub encompasses marketplace education, resource tools and EAI trends, providing members a venue to develop, create and debate.

Ektron

Ektron is a leader in developer-friendly Web content authoring, publishing and management for non-technical end users. Ektron's easy-to-deploy, affordable software enables users to easily update, publish and manage Web content from anywhere without knowing HTML or XML, while working in an environment easily controlled by Web professionals. Nearly one million individuals worldwide use Ektron's award-winning products.

IONA

Artix is a family of Web services based service-oriented integration products that renovate IT assets and consolidate legacy middleware to reduce the complexity and cost of IT operations. Customers use Artix to maximize the value of existing systems through the deployment of best of standards-based middleware and enterprise Web services.

Kenetiks
Education Sponsor

Kenetiks is a worldwide provider of Java-WebSphere training and project mentoring with courses available for every skill level. Our Kenetiks Academy can show you how to build your own J2SE/J2EE applications for IBM WebSphere Application Server and IBM WebSphere Portal Server using WebSphere Studio Application Developer.

Macromedia
Platinum Sponsor

Macromedia is motivated by the belief that great experiences build great business. Macromedia provides the industry-leading software that empowers developers and designers to create and deliver those effective, compelling, and memorable experiences.

Microsoft
Gold Sponsor

Microsoft .NET is a set of Microsoft software technologies for connecting information, people, systems, and devices. It enables a high level of software integration through the use of Web services-small, discrete, building-block applications that connect to each other as well as to other, larger applications over the Internet.

Novell
Silver Sponsor

Novell helps customers dynamically access, provision, manage and deliver secure information services. Customers can flexibly integrate applications and data and ensure unified identity across systems. Novell's Web Services platform, exteNd, is a comprehensive, integrated approach to the visual development and deployment of services-oriented applications.

OASIS
Association Sponsor

OASIS is the non-profit, international consortium that has been providing open solutions for electronic data interchange since 1993. Dedicated from its inception to the technology now known as XML, OASIS is the world's largest independent, vendor-neutral organization for the standardization of XML applications in electronic commerce. The primary mission of OASIS and its members is to identify and resolve interoperability issues that exist between XML applications and technologies.

Sonic Software

Sonic Software is changing the economics of integration. Sonic provides the first comprehensive business integration suite built on an enterprise service bus (ESB). The Sonic product line delivers a distributed, standards-based, cost-effective, easily managed infrastructure that reliably integrates applications and orchestrates business processes across the extended enterprise

Velocitis

Velocitis provides the next level in agile software development tools with Flywheel, a code centric tool for rapidly designing, visualizing and refactoring Visual Studio .NET code. Fine grained synchronization allows the Flywheel and Visual Studio tools to be used in any order, at any time in the development process. With Flywheel developing in code or visually is the same because, the code is the model.

Westbridge Technology
Education Sponsor

Westbridge Technology provides Web Services Management solutions that enable enterprises to secure and manage XML Web Services networks. The Westbridge XML Message Server (XMS), enables organizations like the FTC, Danish Government and Travelers Property Casualty to seamlessly connect, secure and manage XML Web Services networks. For more information, visit www.westbridgetech.com.

WEB SERVICES EDGE CONFERENCE PAST SPONSORS & EXHIBITORS

When to Use Web Services

From Chapter 7

■ There are many compelling reasons to use Web services. It seems as if everyone is at least playing with Web services. Almost every software vendor is building support for Web services into its platforms, languages, and tools. Web services enable any-to-any integration, supporting any programming language, any runtime platform, and any network transport. Technologies such as SOAP and WSDL are simpler to use than traditional integration middleware technologies, and they offer much more flexibility. When combined with domain-specific industry standards, Web services enable unprecedented dynamic interaction. More to the point, they can make it easier for your partners and your customers to do business with you. Best of all, the low cost, pervasiveness, and simplicity of the technology lets your existing staff do more with less.

Web services can be used for many types of applications. Nearly every application requires some integration effort, so I'm sure you can find a way to use Web services in almost any project. After you get a little experience under your belt, you'll very likely adopt Web services as a standard integration technology. When you're just getting started, though, it's always best to limit your scope. Start small. Spend some time learning about the technology. Then you can apply your learning to larger projects. One nice feature of Web services is that you can use them incrementally. There's no need to tackle an enormous project all at once. I recommend that you use Web services in places where they are likely to have a big impact.

Bell Ringers

Which applications would benefit most from Web services? Where should you start? What are the key criteria that should ring a bell in your head and make you think, "This is a job for Web services"?

Heterogeneous Integration

The first and most obvious bell ringer is the need to connect applications from incompatible environments, such as Windows and UNIX, or .NET and J2EE. Web services support heterogeneous integration. They support any programming language on any platform. One

Web Services: A Manager's Guide
by Anne Thomas Manes
Publisher: Addison-Wesley Professional
ISBN: 0321185773
This book excerpt is from Chapter 7 of *Web Services: A Manager's Guide* by Anne Thomas Manes, copyright 2003. All rights reserved. This chapter is posted with permission from Addison-Wesley Professional.

thing that's particularly useful about Web services is that you can use any Web services client environment to talk to any Web services server environment.

For example, JPMorgan uses Web services to connect Excel spreadsheets to UNIX-based financial data. JPMorgan operates the global wholesale businesses for J.P. Morgan Chase. JPMorgan is a leader in investment banking, asset management, private equity, custody and transaction services, middle market financial services, and e-finance. The firm has financial analysts in more than 50 countries around the world. These analysts needed a way to upload and download financial, forecast, and other relevant data used in their spreadsheets to and from various legacy application systems.

Knowing that it's difficult to find a single-vendor solution that would allow it to connect Excel with various UNIX-based systems, JPMorgan decided to use Web services. Web services permit the firm to use the right tool for each side of the equation. JPMorgan created a set of Web services using Systinet's Web Applications and Services Platform (WASP) to enable easy access to the legacy applications. Now the financial analysts can access these services from Excel using Visual Basic for Applications (VBA) macros and the Microsoft SOAP Toolkit, as shown in Figure 7-1.

Unknown Client Environment

The next bell ringer is any situation in which you have little or no knowledge of or control over the client applications that will be used to access the service. Because Web services don't require a specific software environment, you don't need to worry about compatibility issues.

For example, Con-Way Transportation Services uses Web services to support electronic exchange of shipping data with its customers and business partners. Con-Way is a $2 billion transportation company based in Ann Arbor, Michigan. More than two-thirds of its customers are small to medium-sized businesses. Con-Way wanted to provide these customers with a mechanism that would support tight integration with Con-Way's transportation systems. The challenge was that these customers use a variety of transportation applications on a variety of deployment platforms. Con-Way realized that it didn't have the option of asking these customers to install a proprietary API with limited deployment

options to support integrated Con-Way business transactions. Instead Con-Way developed a set of Web APIs using IBM WebSphere. These APIs support invoicing, bill of lading, order pickup, and sales management services. Customers can interface with these services through the Con-Way Web site or use the Web APIs to connect directly from their corporate application systems. The Web APIs support any type of client application–in-house applications as well as packaged applications–as shown in Figure 7-2.

Multichannel Client Formats

A third bell ringer is the need to support many types of client formats, such as browser clients, rich desktop clients, spreadsheets, wireless devices, interactive voice response (IVR) systems, and other business applications. A Web service returns its results in XML, and XML can be transformed into any number of formats to support different client formats.

For example, Wachovia uses Web services to support both browser-based clients and rich desktop clients for Einstein, its customer information system. Wachovia is a leading provider of financial services, with nine million U.S. customer households. Einstein is a GUI application that gives bank staff complete information about a customer, aggregating information from multiple backend systems. Some bank staff use a browser to access Einstein. Others require a richer desktop interface.

As shown in Figure 7-3, Einstein was developed as a multitier Web service application. The backend business functions and data sources are legacy applications implemented in CICS and DB2 on the mainframe. The middle tier, which accesses and aggregates the customer information, is implemented as a set of J2EE Web services using IBM WebSphere. The client environments are implemented using Microsoft .NET. The browser client is implemented using Microsoft .NET WebForms, and the desktop client is implemented using Microsoft .NET WinForms. Einstein's architecture also allows Wachovia to implement other types of client interfaces to support IVR systems, wireless handsets, two-way pagers, and other devices.

Other Web Services Applications

Web services can help you accomplish many types of business goals. You can use Web services to solve immediate tactical problems. You can use them to help you manage your
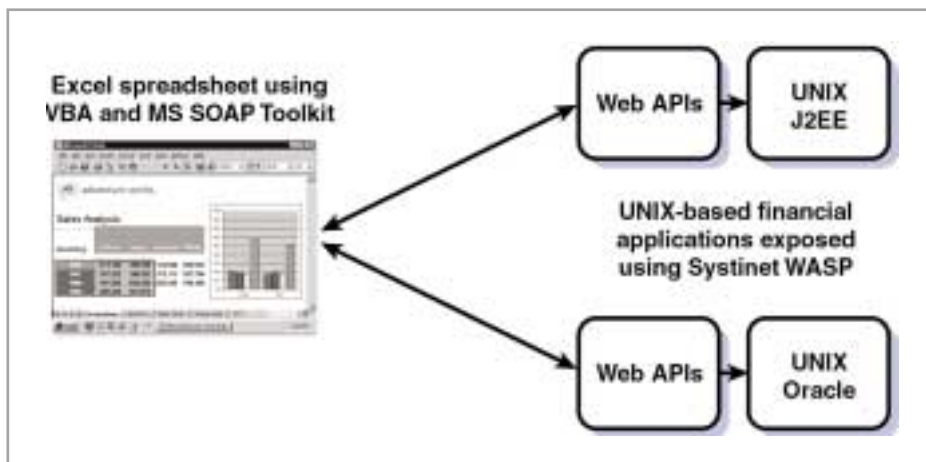

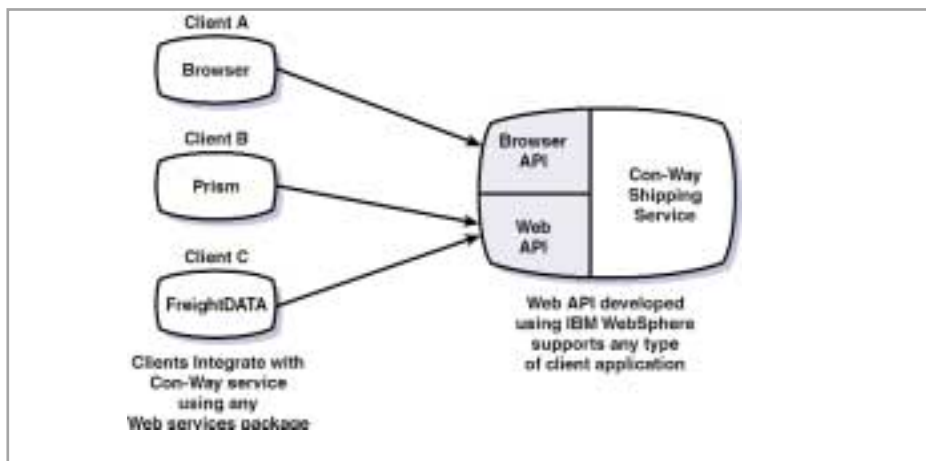FIGURE 7-1 | **JPMorgan's heterogeneous integration project**

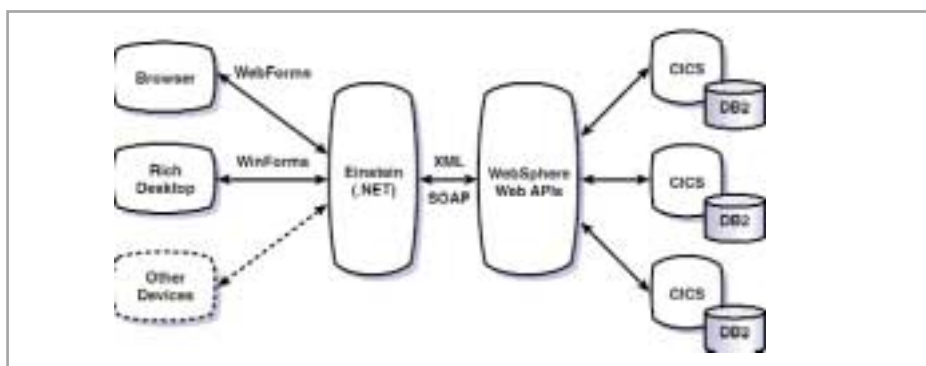FIGURE 7-2 | **Con-Way's unknown client environment project**

FIGURE 7-3 | **Wachovia's multi-channel client project**

software assets, leverage legacy applications, and reduce development costs. Web services can also help you optimize your business process and improve customer relationships.

Point-to-Point Integration

The first and most basic way to use Web services is for simple point-to-point integration. For example, Cape Clear uses Web servic-

es to connect employees' e-mail clients with its CRM solution. Cape Clear is a Web services software startup. It uses Salesforce.com as its CRM solution. Salesforce.com provides a hosted CRM solution using an ASP-style model. Users typically interface with the CRM solution through a browser, recording customer contact information and correspondence.

Like most software startups, Cape Clear

provides e–mail-based customer support. As a result, quite a bit of customer correspondence takes place via e-mail. But Salesforce.com didn't provide a simple, easy way for Cape Clear employees to log this correspondence in the Salesforce.com database. Users had to copy and paste the e-mail from Outlook into the Salesforce.com browser interface. Cape Clear found that lots of correspondence wasn't getting recorded.

Salesforce.com provides a programming API, so Cape Clear decided to eat its own dog food and address this problem using Web services. First Cape Clear used Cape Clear Studio to develop a Web service adapter for

the Salesforce.com native programming API. This adapter accepts a SOAP request and translates it into the Salesforce.com native API. Next Cape Clear developed an Outlook macro using VBA and the Microsoft SOAP Toolkit. This Outlook macro adds a button to the standard Outlook tool bar labeled "Save to Salesforce." As shown in Figure 7-4, when the user clicks on this button, the Outlook macro captures the e-mail message, packages it as a SOAP message, and sends it to the Salesforce.com adapter Web service. The Web service then forwards the e-mail using the native API to Salesforce.com, which logs it.

After seeing the advantages of using SOAP

for integration, Salesforce.com has decided to develop its own set of Web APIs in addition to the native programming APIs. If you are a Salesforce.com customer, you won't need to build your own adapter Web services.

Consolidated View

One of the most popular internal integration projects is enabling a consolidated view of information to make your staff more effective. For example, you probably have many people in your organization who interact with customers. Each time your staffs interact with the customer, you want to let them have access to all aspects of the customer relationship. Unfortunately, the customer relationship information is probably maintained in variety of systems. The good news is that a consolidated customer view provides a single point of access to all these systems.

You can use Web services to implement this type of consolidated view. For example, Coloplast is using Web services to improve its sales and customer support functions. Coloplast is a worldwide provider of specialized healthcare products and services. As part of an initiative to improve customer relationships, Coloplast wanted to set up a state-of-the-art call center system that would give customer representatives real-time access to complete customer histories and product information. The company selected Siebel Call Center as the base application, but it needed to connect this system to its backend AS/400-based ERP systems, which manage the sales, manufacturing, and distribution functions. It did so using Web services. Coloplast used Jacada Integrator to create Web services adapters for the legacy AS/400 application systems. As shown in Figure 7-5, Siebel Call Center uses these Web services to deliver a 360 degree view of customer relationships, including access to backend processes such as open order status, inventory information, customer credit checking, and special pricing. This solution improves efficiency and enhances employee and customer satisfaction.
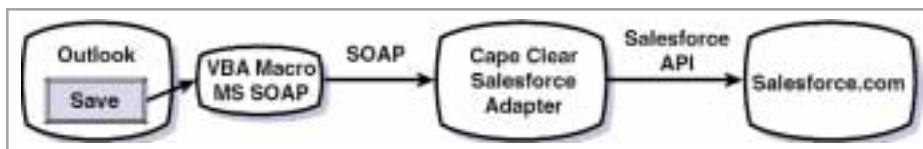
Managing Legacy Assets

Web services can make it easier to manage and maintain your legacy application assets. For example, AT&T estimates that Web services technology has reduced the time it takes to make modifications to some of its oldest application systems by 78 percent.

TIRKS (Trunks Inventory Record Keeping

FIGURE 7-4 | Cape Clear's point-to-point integration project
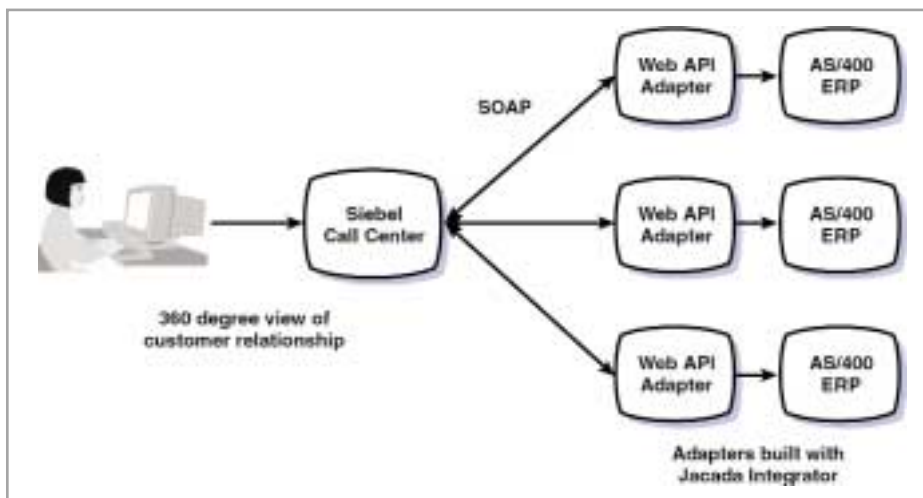
FIGURE 7-5 | Coloplast's consolidated customer view project
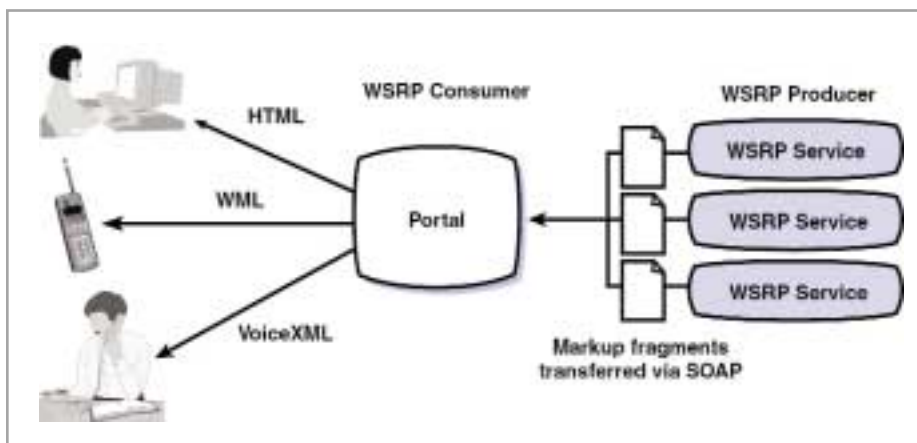
FIGURE 7-6 | Web Services for Remote Portals (WSRP)

System) is a critical application for AT&T. TIRKS was first developed in the 1960s, and it is connected to more than 100 other application systems. Because of the brittleness of these application connections, every time AT&T makes a modification to TIRKS, it must also make a corresponding modification to the other systems. Using Web services, AT&T has developed much more flexible application connections that don't break every time a change is made to the application. AT&T is using IONA XMLBus to replace the more than 100 brittle application connections with a much smaller set of flexible, reusable Web APIs to TIRKS. Now each modification to TIRKS no longer requires the associated changes in all the other application systems.

Reducing Duplicative Applications

One of the more popular ways to use Web services is to reduce redundant applications. A service can support many types of application clients. If you need to perform the same type of function via multiple applications, it makes a lot of sense to develop a single service shared by all these applications rather than duplicate the functionality in each application.

Reduction of redundant applications is a key objective of the U.S. government's E-Gov initiative. The U.S. government encompasses hundreds of federal agencies and bureaus, and there is significant overlap and redundancy of systems across these agencies. A 2001 study by the E-Gov Task Force analyzed the agencies to identify the various business activities performed by the government. The study identified 30 general lines of business, such as economic development, public safety, environmental management, and tax collection. On average, each agency is involved in 17 lines of business, and each line of business is performed by 19 agencies. Some lines of business—such as payroll, travel, HR, procurement, logistics, administration, and finance—are performed by every agency.

The U.S. government spent $48 billion on information technology in 2002 and will spend $52 billion in 2003. The Office of Management and Budget estimates that the government can save more than $1 billion annually in IT expenditures by aligning redundant IT investments across federal agencies. In addition, this alignment will save taxpayers several billion dollars annually by reducing operational inefficiencies, redundant spending, and excessive paperwork.

In October 2001, the President's Management Council approved 24 high-pay-off government-wide initiatives that integrate agency operations and IT investments. One of those initiatives is E-Travel, which is being run by the U.S. General Services Administration (GSA). E-Travel delivers an integrated, government-wide, Web-based travel management service. Federal government employees make approximately four million air and rail trips each year, and until recently each agency and bureau managed its own travel department. Cumulatively, these various departments used four travel charge card providers, six online self-service reservation systems, 25 authorization and voucher processing systems, 40 travel agencies, and a unique payment reimbursement system for almost every bureau.

By consolidating these travel systems into a single, centralized travel management system, the U.S. government expects to save $300 million annually, achieving a 649 percent return on investment. In addition, the consolidated system will deliver a 70 percent reduction in the time it takes to process vouchers and reimbursements.

GSA delivered the first phase of E-Travel in December 2002—an online self-service reservation system. The total end-to-end travel management system is scheduled to be complete by December 2003. The system will use a service-oriented architecture, based on XML and Web services, to ensure easy integration with existing agency systems and future adaptability. The E-Travel team refers to this architecture as "Velcro integration," indicating that modules can be easily replaced when necessary.

Managing Portal Initiatives

Web services can also be very useful as a means to manage and coordinate your portal initiatives. A portal is an integrated, Web-based view into a host of application systems. A portal contains a piece of application code (a portlet) for each backend application. A portlet contains the code that talks to the backend application as well as the code that displays the application in the portal.

Web services technology enhances portals in two ways. First, Web services deliver content to the portal as XML. It's then easy for a portal engine to take this XML content and display the information in a portal frame. It's also easy for the portal engine to reformat the XML content to support other client devices, such as wireless handsets or PDAs. Second,

Web services technology defines a simple, consistent mechanism that portlets can use to access backend applications. This consistency allows you to create a framework to make it quicker and easier to add new content to your portal. Furthermore, as mentioned in Chapter 5, the new OASIS WSRP specification will allow you to add new content to the portal dynamically. Figure 7-6 shows an overview of WSRP.

Another goal of the U.S. government's E-Gov program is to get a handle on government portals. As of February 2003, the U.S. government was managing more than 22,000 Web sites with more then 35 million Web pages. These Web sites have been developed, organized, and managed using the same stovepipe mentality as used in the backend agency applications. Such decentralization and duplication make it difficult for citizens and communities to do business with the government. For example, a community that is attempting to obtain economic development grants must do a tremendous amount of research to learn about federal grants. There's no single source of information. More than 250 agencies administer grants, and you would have to file more than 1,000 forms (most with duplicate information) to apply for all of them. Some of these forms are available online; others aren't. Currently all forms must be filed by postal mail.

The government is working to consolidate this myriad of Web sites into a much more manageable number of portals, each providing a single point of entry to a particular line of business. Each portal will use Web services to access the backend applications that implement the business process. In many cases the government will consolidate backend applications to reduce redundant systems and to ensure a simpler experience for the portal users. For example, the forthcoming E-Grants portal will provide a single point of entry for anyone looking to obtain or administer federal grants. This site will help citizens learn about all available grants and allow them to apply for these grants online. The government expects to save $1 billion by simplifying grant administration as well as saving $20 million in postage.

All government portals will be coordinated through the FirstGov portal at http://www.firstgov.gov. From this one portal, citizens, businesses, and government agencies will have a single point of entry to all other government portals. ◉