SYS-CON
NOW INCORPORATING XML JOURNAL

SOA Web Services JOURNAL

May 2006 Volume 6 Issue 5

Identity Propagation in a SOA

The shortcomings of current solutions

pg.10

Best Practices for Securing Web Services

Ensuring complete Web Service security

pg.38

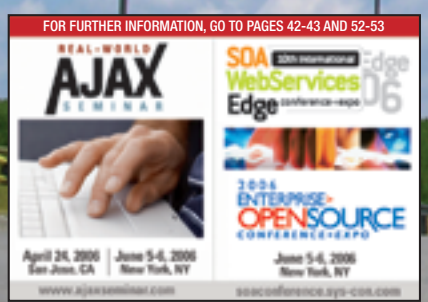
SOA: Focus Is on Approaches Not Technology

pg.50

AJAX COMPOSITE APPS

The last mile between your users and your SOA

FOR FURTHER INFORMATION, GO TO PAGES 42-43 AND 52-53



RETAILERS PLEASE DISPLAY
UNTIL JULY 31, 2006
$6.99US $7.99CAN
05>
71486 03420 9

Altova® XMLSpy® 2006 – The industry standard XML development environment.

ALTOVA® xmlspy® 2006

Bring your development plans to light

Sneak a peek at XMLSpy® 2006, and see how vital it is to master XML.

New in XMLSpy 2006:

- Schema-aware XSLT 2.0 support
- Schema-aware XQuery support
- Updated platform integration for Microsoft® Visual Studio® .NET 2005
- Updated platform integration for Eclipse 3.1

Altova® XMLSpy, the industry standard XML development environment, is indispensable for modeling, editing, debugging and transforming all XML-related technologies. Illuminate your strategy with advanced standards compliance, extended platform integration, and enlightened usability aides. Use XMLSpy to structure XML Schemas and devise XML documents, then automatically generate runtime code from schemas in multiple programming languages. Become a markup mastermind!

Download XMLSpy® 2006 today: www.altova.com

ALTOVA®

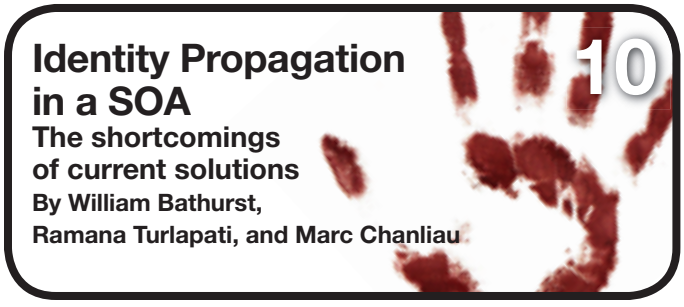
Also available in the Altova XML Suite.

SOA Web Services Journal

May 2006 Volume 6 Issue 5

XML Journal

Inside WSJ



From the Editor
The Only Thing Worse
Sean Rhody 7

SOA Web Services
Industry News
..... 8

WSJ: SOA
Does a Web Service Make a Service for SOA?
SOA Service & SOA SLA as drivers for renovating legacy applications for SOA
Michael Poulin 18

WSJ: Frameworks
WSDL 2.0: A Pragmatic Analysis and an Inter-operation Framework
Minimizing interoperation issues with a WSDL version management framework
Srinivas Padmanabhuni, et al. 24

WSJ: Patterns
WEB 2.0
Its Component Model and Message Exchange Patterns
Chris Madrid 30

WSJ: Product Review
ActiveBPEL 2.0 from Active Endpoints Excels at BPEL
Paul T. Maurer 36

Bringing Application Awareness to the IP/MPLS Service Provider Cloud
JONATHAN BOSLOY
56

_INFRASTRUCTURE LOG

_DAY 8: I give up. Our infrastructure is so inflexible. Our apps and processes don't work together. We can't respond quickly to change. It's out of control.

_Gil had an epiphany. Duct tape. A few dozen rolls later and he's integrated everything, and everyone, by hand.

_DAY 10: Duct tape can fix many things. Basketballs. Sofas. Doorknobs. But not widespread app and process inflexibility.

_DAY 13: I've found something better: IBM WebSphere middleware. It'll make our infrastructure more flexible by seamlessly integrating our apps. We can change processes in a snap and use what we already have—even apps from SAP and Oracle. And with IBM's industry-specific expertise, we're on our way to enabling a service oriented architecture.

_Hmmmm...WebSphere. More powerful than duct tape.

WebSphere®

Download our IBM SOA Assessment Tool at:
IBM.COM/**TAKEBACKCONTROL**/SOA

SOA Web Services Journal

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

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

EDITORIAL
EDITOR-IN-CHIEF
Sean Rhody sean@sys-con.com
XML EDITOR
Hitesh Seth
INDUSTRY EDITOR
Norbert Mikula norbert@sys-con.com
PRODUCT REVIEW EDITOR
Brian Barbash bbarbash@sys-con.com
.NET EDITOR
Dave Rader davidr@fusiontech.com
SECURITY EDITOR
Michael Mosher wsjsecurity@sys-con.com
RESEARCH EDITOR
Bahadir Karuv, Ph.D Bahadir@sys-con.com
TECHNICAL EDITORS
Andrew Astor andy@enterprisedb.com
David Chappell chappell@sonicsoftware.com
Anne Thomas Manes anne@manes.net
Mike Sick msick@sys-con.com
Michael Wacey mwacey@csc.com

INTERNATIONAL TECHNICAL EDITOR
Ajit Sagar ajitsagar@sys-con.com

EXECUTIVE EDITOR
Nancy Valentine nancy@sys-con.com

ONLINE EDITOR
Roger Strukhoff roger@sys-con.com

PRODUCTION
LEAD DESIGNER
Andrea Boden andrea@sys-con.com
ART DIRECTOR
Alex Botero alex@sys-con.com
ASSOCIATE ART DIRECTORS
Abraham Addo abraham@sys-con.com
Louis F. Cuffari louis@sys-con.com
Tami Beatty tami @sys-con.com

CONTRIBUTORS TO THIS ISSUE
Abhishek Malay Chatterjee, Terance Dias, Aaron Flint, Dan Hynes, Matjaz B. Juric, Sean Kline, Geo Philips Kuravakal, Francois Lascelles, Andrew Lawlor, David Linthicum, Roy Mitchell, Srinivas Padmanabhuni, James Pasley, Varun Poddar, Sean Rhody, Vivek Singhal, Doug Todd, Ajit Sagar, Brian Wilson

EDITORIAL OFFICES
SYS-CON MEDIA
135 CHESTNUT RIDGE ROAD, MONTVALE, NJ 07645
TELEPHONE: 201 802-3000 FAX: 201 782-9637
WEB SERVICES JOURNAL (ISSN# 1535-6906)
Is published monthly (12 times a year)
By SYS-CON Publications, Inc.
Periodicals postage pending

Montvale, NJ 07645 and additional mailing offices
POSTMASTER: Send address changes to:
WEB SERVICES JOURNAL, SYS-CON Publications, Inc.
135 Chestnut Ridge Road, Montvale, NJ 07645

The Only Thing Worse

WRITTEN BY
SEAN RHODY

If you work in the IT industry long enough, you're bound to hear one particular joke (well, you'll hear a number, I want to focus on this one) – "What's the only thing worse than no architect on a project?" The answer of course is "Two or more". And of course that's true, since when you put four architects in a room, you get five opinions (at least one is schizophrenic) on anything.

Nevertheless, as the joke does make clear, architecture is important. Without it, projects flounder, or worse yet succeed in a dizzying display of spaghetti code and cowboy heroics. When it does, the mess it creates comes back to roost in the future. I've visited shops where they can't make changes to their main business processes, because the COBOL code that runs it no longer even exists in source form – so they have to make changes at the compiled code level. In a very real sense, that's an architecture failure – they haven't kept up with the times, assured a realistic software portfolio management program, and maintained good software development life-cycle practices. This is not uncommon.

One thing that often gets missed in the acronym SOA is Architecture. It's fairly easy to service-enable code in this day and age – most modern solutions and platforms give you basic Web services capabilities out of the box. I even know multiple ways to reach the COBOL code (although I shudder at the thought of people writing Web services in COBOL, but even that is happening) and make it play nice in a services environment.

The point though is that service enablement is not service-oriented architecture. Architecture discusses various dimensions of software, such as how to build it, how to reuse it, how fast it needs to be, how much downtime it can have, and a number of other topics. Some folks divide architecture into business, application, and technical (or infrastructure). Others look at different levels – conceptual, logical, and physical. People with specialties (such as security) see the world as a combination of both sets, all slanted toward what is important to them. Enabling services is a good thing for many environments, but for each Service, the management, the SLAs and QoSs, the security and the transactionality of the Service must be considered. And not in a vacuum. That's where Architecture participates in the conversation. A good architecture forms a basis for the development of Services in a structured organized fashion with clean demarcations of responsibility and attention to the various performance characteristics that make a Service really work.

Realistically, that is the broadest definition of Architecture – practices that enable software to work. And practices definitely include process and not just the technical aspects of software development. In any large shop, the process of accomplishing software development is a critical part of architecture. If the enterprise concepts developed by an architecture group aren't driven into the development organization and monitored, the benefits of architecture can't be realized. That's process, not technology, but it's certainly still in the domain or architecture.

Service-oriented architecture emphasizes application delivery via discrete services over monolithic applications. Every type of architecture has its own challenges, but SOA in particular places a great deal of emphasis on layers of services and the ability to apply services discretely. Responsibilities are encapsulated outside of the services – there's usually no need for the service to understand security or service-level agreements as someone else manages that above or around them. Because each service is discrete, and to a certain extent can ignore the impact of other services, the Architecture becomes even more important, as it is the glue that holds the whole thing together and makes it function.

While two or more architects may be the worse thing in the world, remember, no architects (and by extension, no Architecture) is the next worse thing.

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

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

ADVERTISING
SENIOR VP, SALES & MARKETING
Carmen Gonzalez carmen@sys-con.com
VP, SALES & MARKETING
Miles Silverman miles@sys-con.com
ADVERTISING DIRECTOR
Robyn Forma robyn@sys-con.com
NATIONAL SALES & MARKETING MANAGER
Dennis Leavey dennis@sys-con.com
ADVERTISING MANAGER
Megan Mussa megan@sys-con.com
ASSOCIATE SALES MANAGERS
Kerry Mealia kerry@sys-con.com

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

CUSTOMER RELATIONS/JDJ STORE
CIRCULATION SERVICE COORDINATORS
Edna Earle Russell edna@sys-con.com

sys-con.com
CONSULTANT, INFORMATION SYSTEMS
Robert Diamond robert@sys-con.com
WEB DESIGNERS
Stephen Kilmurray stephen@sys-con.com
Wayne Uffleman wayne@sys-con.com

ACCOUNTING
FINANCIAL ANALYST
Joan LaRose joan@sys-con.com
ACCOUNTS PAYABLE
Betty White betty@sys-con.com
ACCOUNTS RECEIVABLE
Gail Naples gailn@sys-con.com

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

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

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

SYS-CON MEDIA

AgilePath Announces New SOA Governance Reference Model

(Newburyport, MA) – AgilePath Corporation, an SOA management and technology consulting firm, announced the release of its Service-Oriented Architecture (SOA) Governance Reference Model Version 1.0 at the Federal CIO Council's Architecture and Infrastructure Committee (AIC) meeting held on February 17, 2006 in Washington, DC. AgilePath's SOA Governance Reference Model is a framework that helps organizations place more emphasis on SOA governance from an organizational, process and cultural perspective while deemphasizing the technology-focused approaches being endorsed by the vendor community.

AgilePath announced its initial SOA governance and organizational dynamics solutions in 2005, and quickly watched SOA governance become oversimplified into vendor-and product-centric themes rather than placing proper emphasis on the true drivers of organizational behavior. As a key facet of AgilePath's patent-pending SOA Playbook methodology, AgilePath's SOA Governance Reference Model will help organizations realize their business and SOA strategies through robust governance, policy enforcement, and attention to organizational, process, and behavioral factors. www.agile-path.com

ZapThink Report: DataDirect XQuery Simplifies Data Integration

(Bedford, MA) – DataDirect Technologies, a provider of data connectivity and mainframe integration, and an operating company of Progress Software Corporation has announced the findings of an independent report on DataDirect XQuery, a Java component for integrating relational and XML data using XQuery. The new study was written by ZapThink, an IT research and advisory firm specializing in XML, Web services, and service orientation. The report concludes that DataDirect XQuery offers companies the means to win out at long-lived data integration challenges because it leverages the power of standards, allowing the user to determine the runtime infrastructure. www.datadirect.com/zapthink_report

IBM Launches Open Test Drive of SOA-Tuned DB2 'Viper' Data Server

(Jaipur, India) – IBM has launched an open test drive of the next-generation DB2 data server – code-named "Viper" – which is designed to help customers manage and access data via an information-centric approach to service0oriented architecture (SOA) with flexibility, speed, and security.

Among the new SOA-tuned security features introduced within Viper will be DB2 Label Based Access Control (LBAC), a data access capability that allows users to define structures within the database in ways they have never been able to before. In addition to row-level access control, the new column-level labeling capability provides new ways to control access to sensitive data stored within the database. www.ibm.com/db2/viper

Skyway SOA Platform Now Available in Developer Edition

(Tampa, FL) – Skyway Software, a provider of service-oriented architecture (SOA) design and delivery solutions, has announced that the Skyway SOA Platform is now available in a Developer Edition. The Developer Edition of the Skyway SOA Platform accelerates service solution delivery for small businesses or can act as a "testing" application for larger companies evaluating service development strategies.

This integrated design and delivery platform reduces application backlog and puts developers on the SOA fast track.

- Creating Web services and UIs in seconds
- Delivering consistent standards-based J2EE code
- Requiring minimal training
- Ensuring service compliance and reuse
- Controlling application and service deployments, even in highly distributed environments

Skyway Software's claim that its SOA Platform can build Web services in seconds comes from the cutting-edge nature of their solution framework, an SOA technology that represents the next generation of application development. The Skyway SOA Platform includes an SOA architecture model, an application model (Web 2.0) and a model-driven development environment that together create an agile and service-centric development solution. A key product differentiator is that the Skyway SOA Platform builds new and assembles existing services while many competitive solutions only assemble composite services. The Platform also manages and governs reuse across multiple platforms from any type of data source. www.skywaysoftware.com

Ah, remember when EDI was young and full of promise?

The future of EDI is B2B over IP

EDI sure had promise in the sixties – but its complexity, inflexibility, and the bottleneck of the VAN make it very costly today.

Simple to integrate, easy to manage, and blazingly fast, **terminalONE** is *the* B2B over IP platform. It intelligently transports, transforms, and routes all your data transactions.

We're about ebusiness adaptability: your business world changes fast – wouldn't it be nice if your data interchange adapted quickly and painlessly along with it?

terminalONE™

Secure, intelligent ebusiness transactions

*high-**velocity***
*high-**volume***
*high-**availability***

Take **terminalONE** for a test drive – ***today***
www.xenos.com/VAN

1 888 242 0695
1 905 763 4468
terminalONE@xenos.com

xenos

Identity Propagation in a SOA

The shortcomings of current solutions

One of the challenges IT organizations face is how to propagate identities in complex business processes that are commonly found in Service Oriented Architectures (SOAs). Identities, which are passed from one service invocation to the next in a business process, give the process a user context. Identities can be used to determine access rights to SOA services and for audit and compliance purposes.

WRITTEN BY

WILLIAM BATHURST

RAMANA TURLAPATI

MARC CHANLIAU

For example, consider a procurement business process for an application that's used by a number of purchasing agents. Each agent has a different purchasing privilege. Say a senior agent can purchase up to $50,000 in a transaction, while a junior agent can buy only $25,000. If the business process that enables the purchase is composed of a number of SOA services, each service must know the user's identity to enforce purchasing privileges.

This article shows the need for identities in an SOA, provides examples of SOAs, and reviews the status and shortcomings of current solutions.

Introduction to Identity Propagation

Before we look at identity propagation in an SOA, let's look at it in a three-tier environment, where it's easier to illustrate the basic concepts. Once again, we'll use the procurement application scenario — except this time, the application resides in a Web-based portal instead of being loosely coupled in an SOA.

The presentation and logic tiers exist in the portal application server and the data tier resides in the database. The identity of a procurement agent is established when the user starts accessing the portal application from the Web browser and the identity spans all three tiers of the portal. This identity is used for authentication and authorization purposes throughout the business processes tha span the portal. Identity propagation in this case spans from the Web browser, to the portal, to the backend database (see Figure 1).

To fully illustrate identity propagation, let's dig deeper into this scenario and see how the identity is propagated. The procurement application, which sits in the portal, requires the user to log in to gain access. When the agent initially accesses the portal, the portal presents a JSP- or HTML-based form that requires a username and password. These credentials are sent over an encrypted SSL channel to prevent anyone from sniffing the password over the wire.

Let's assume that the portal is running in a J2EE application server. The application would typically use a Java Authentication and Authorization Service (JAAS) login module to process the username and password, and then authenticate and authorize the user. The username and password credentials are checked against an LDAP directory, or perhaps an identity management infrastructure. If the login is successful, a JAAS *subject* is created in the current execution context of the J2EE portal. This object is used to identify the user in the J2EE container.

The *subject* is used to authorize any subsequent requests from the user to a secured

Fiorano SOA™ 2006

The Quickest Path to an SOA

Fiorano provides the industry's first framework enabling developers to rapidly create SOA applications in well-formed J2EE code.

Key Features:

- JCA-compliant Service-component Framework
- JMS based messaging
- BPEL support for composite components and applications
- Synchronous and Asynchronous Services in a single framework
- Support for multiple languages : Java, C, C++, C# and others

With Fiorano you can:

- Directly deploy Service Components and BPEL processes to any standard J2EE container
- Rapidly deliver ACID transactions across front-end (JSP) and backend processes (Databases, ERPs, etc.)
- SOA enable your existing J2EE application (EJB, JCA, JMS) without reengineering or code-generation
- Easily debug complex flows spanning multiple applications and middleware
- Perform runtime service deployment, service updates and application changes

Download your copy of Fiorano today!

http://www.fiorano.com/downloads

Fiorano®

Enabling Change at the speed of thought

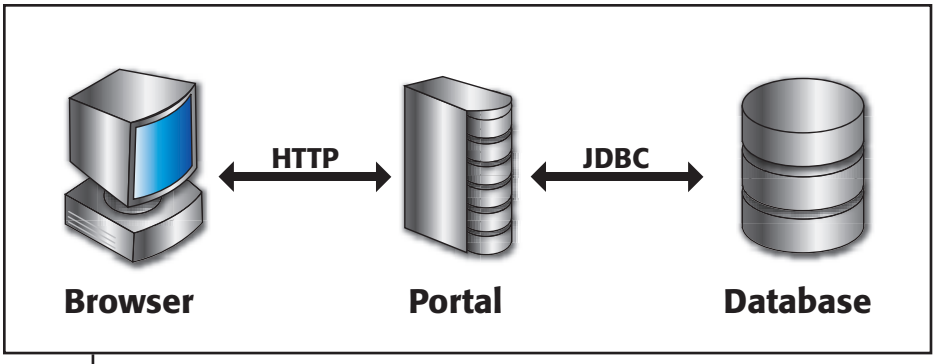


FIGURE 1 | Basic three-tier architecture

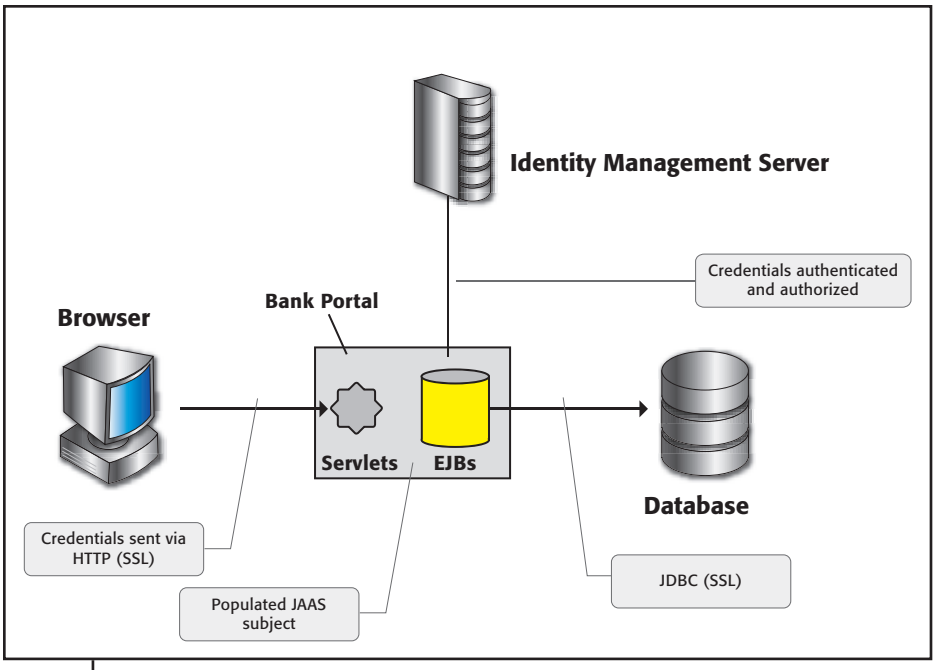


FIGURE 2 | Identity Propagation through a 3-tier architecture

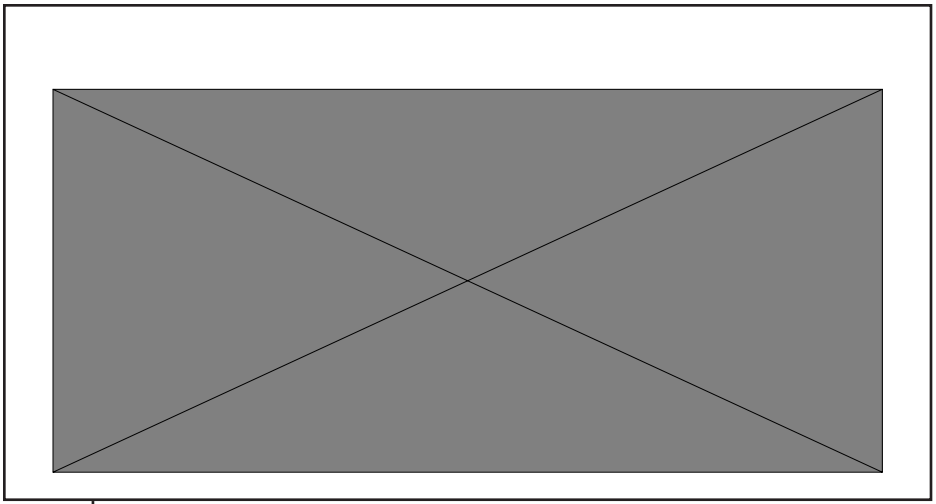


FIGURE 3 | Basic SOA example

resource in the application server. For example, the secured resource may be an Enterprise Java Bean (EJB) that accesses the portal's backend database. The *subject* is used to determine if the user should have access to the EJB. The user identity could also be propagated to the database using proprietary techniques such as impersonation, which could be used to determine if the user should have access to the backend data.

Figure 2 shows how an identity can be passed from the browser to the backend database. The identity is first passed from the browser to the portal application. From there, it can be propagated to EJBs or databases. At each step, the identity is bound to the resource. For example, JAAS is used to bind the identity of the portal user to an executing thread in the portal procurement application. This way, the user's identity can be used to determine access to subsequent resources.

The identity can also be used for audit and compliance purposes. The portal can set alerts for authentication or authorization failures in the banking application or database. Useful data can also be mined based on the user identities passing through the portal. For example, the bank could determine if purchasing agents are trying to exceed their purchasing limits.

Identities in an SOA

In our example, the techniques used to propagate identities are often proprietary or non-standard. This works well in closed environments, but in heterogeneous environments where services must be interoperable, proprietary techniques fall short. SOAs are typically composed of multi-vendor heterogeneous environments.

Think of an SOA as an evolution of the three-tier architecture where applications, like the portal, are loosely coupled applications built as a collection of services. The idea is to expose business logic as services in a reusable and interoperable fashion. For example, a service could:

- Return a list of items that can be purchased
- Return the status of a purchase order
- Submit a purchase order.

SOA services aren't necessarily uniform. For example, they could be exposed through different types of protocols such as JMS, REST,

Altova® DiffDog® 2005 – The developer's dedicated differencing utility.

ALTOVA® diffdog® 2005

Detect differences in files and folders

Depend on DiffDog 2005, the developers' dedicated differencing utility.

Unleashed in DiffDog 2005:

- Source code and text file compare and merge capabilities
- Side-by-side directory synchronization
- On-the-fly file editing
- XML-aware file and directory differencing

Let DiffDog 2005 track down the differences in your development projects. Call on it to quickly compare files and contrast directories. Merge content with just a click, or open and edit files and folders then instantly re-compare. DiffDog 2005 highlights variances side by side, providing cues and controls that make synchronization simple. It even offers advanced XML-aware differencing and editing capabilities based on those standardized by Altova XMLSpy®. Coordinate your code!

Download DiffDog® 2005 today: www.altova.com

Visit Altova at JavaOne **booth # 626**

ALTOVA®

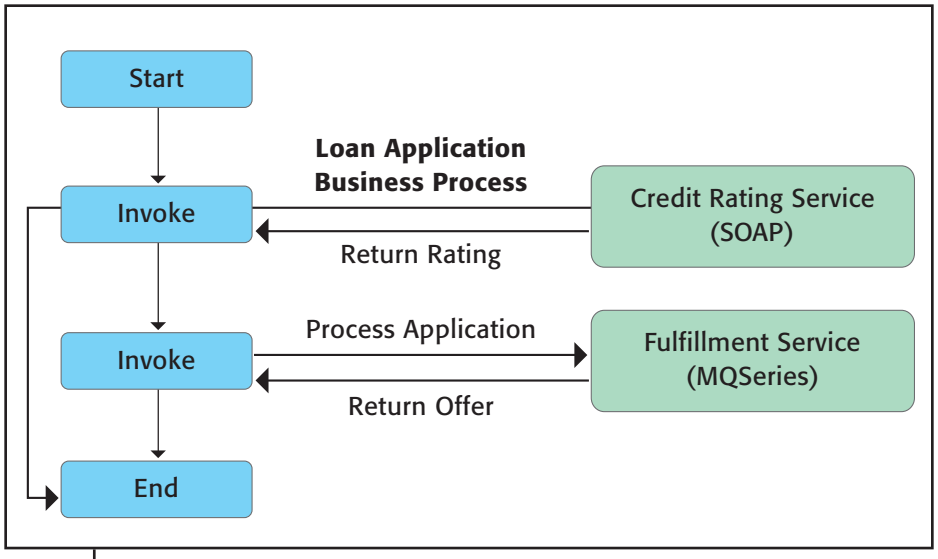


FIGURE 4 Orchestrated loan application business process

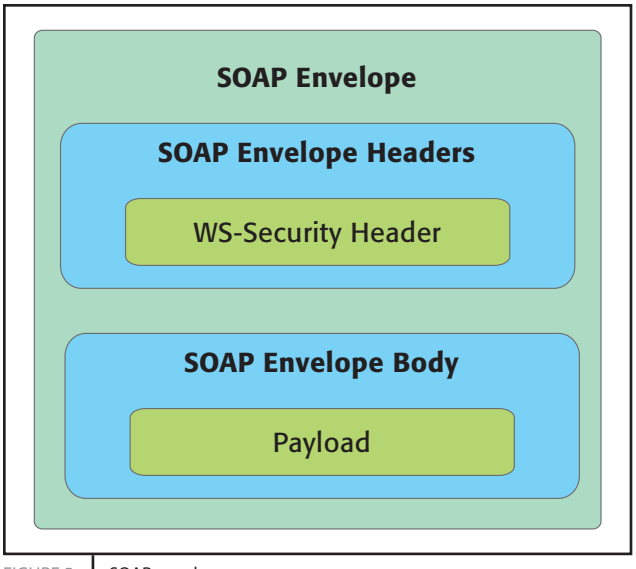


FIGURE 5 SOAP envelope

RMI, .NET Remoting, MQSeries, or SOAP (see Figure 3).

These services can also be orchestrated business processes where services are wired together into business flows and are often orchestrated using open standards such as BPEL. For example, consider an auto loan service, where a bank customer submits an application online. The service processes the application and does a credit check on the applicant. If the applicant's credit meets a certain standard, it's forwarded to the fulfillment service. After the processing is finished, the paperwork is sent to the orchestrating process (see Figure 4).

SOA is quite flexible and powerful, but its decoupled design makes it difficult to propagate an identity across business processes. For example, a transaction may span a multitude of messaging services such as Web Services, MQSeries, and JMS.

Each service has its own way of transporting identities. JMS and MQSeries can pass SOAP-based XML messages in their payload, but these services often aren't XML-based and use different payload types. SOAP-based Web Services have a distinct advantage over other messaging protocols since they can use WS-Security headers in the SOAP envelope to propagate identities (see Figure 5).

The WS-Security header is standardized security metadata located in a SOAP header in the SOAP envelope. WS-Security provides data integrity (XML encryption) and data authenticity (XML signature). In addition, it offers a way to insert standard security tokens such as X.509 certificates, Kerberos tickets, and Security Assertion Markup Language (SAML) assertions in the WS-Security header. For example, SAML was designed to provide a standardized exchange of security information using XML documents referred to as SAML assertions. The following code shows how an identity would be bound to a SOAP message using SAML:

```
<wsse:Security ...>
      . . .
    <saml:assertion= ...>
        . . .
        <saml:Subject>
            <saml:NameIdentifier ...>
                CN=Joe User,
OU=purchasing, O=Widget Inc
            </saml:NameIdentifier>
          </saml:Subject>
            . . .
 <wsse:Security ...>
```

Binding the original requestor's identity to the request itself is the way to propagate identities. The request may be modified throughout the lifecycle of the transaction, but the identity of the requester must always be attached to the request. In this context, identity propagation presents many advantages. At each step, the user's identity is used to determine access to any secured resource.

WS-Security provides the semantics for binding user information to SOAP messages. In the listing above, the identity of the user is Joe User, who is in the purchasing organization at Widget, Inc. This identity is bound to the SOAP message using SAML as defined by the open WS-Security.

The SAML token goes beyond just identifying the user. It can also package additional information about the user in the form of attributes, which are used for authorization decisions. Attribute statements provide specific details about the subject; for example, the user holds Gold status. Authorization decision statements identify what the subject is entitled to do. For example, SAML assertion attributes can be mapped to roles defined in an access control infrastructure. A relying party that processes a SAML token could use these statements for fine-grained access control.

SOA Identity Propagation

Let's return to the orchestrated business process example that accessed the credit rating and loan processing services. The ful-

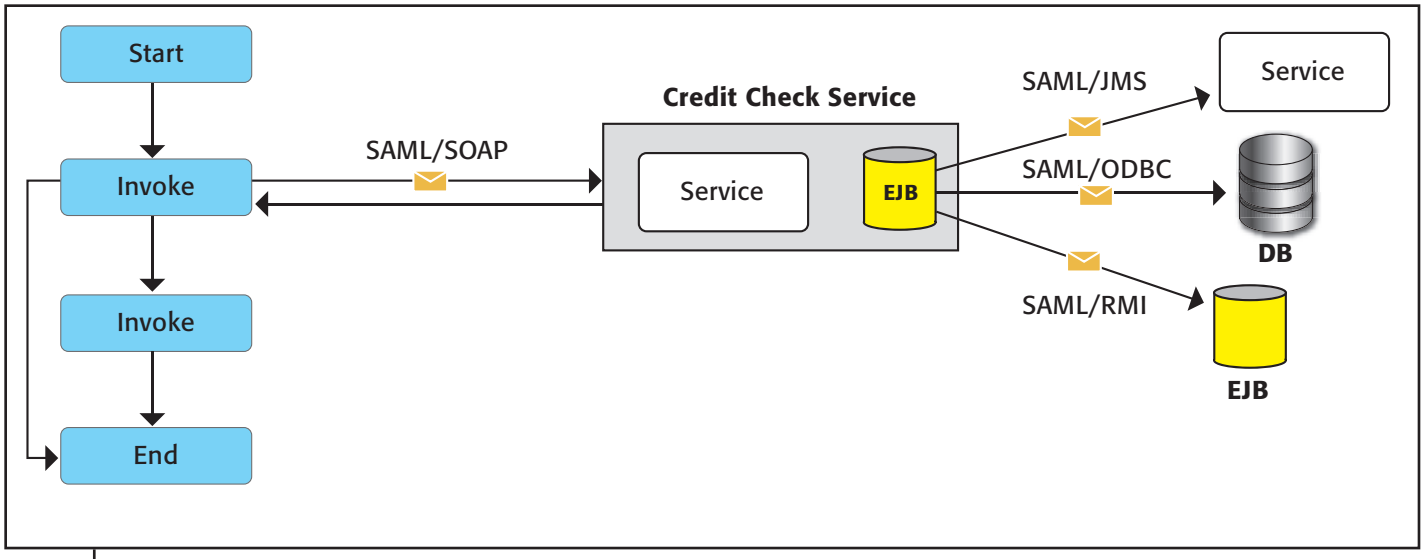


FIGURE6 SAML token propagation in the credit check service

fillment service could be exposed as a Web Service, but the credit rating service might use MQSeries to access a legacy database. These two services can use different means to propagate identities. If you need to tie these services together, how do you propagate the identity from SOAP into a native MQSeries? This can be the start of many headaches. If a business process spans multiple services, how do you relay the identity of the original requestor throughout the transaction?

To simplify this problem, identity propagation should ideally be carried out with a single security token – for instance, a SAML assertion as described above. Secure identity propagation lets you make sure that only appropriate requests are processed. It also provides an audit trail throughout a transaction. Identity propagation requires that the identity of the original requester be bound to each step of the business process or transaction.

Business processes found in SOAs often span a multitude of protocols. The security token should have a standard way to bind to these protocols. A SAML token, as a standard XML representation for describing user identity and attributes, is uniquely suited for this purpose. Figure 6 shows a simple example of a SAML token spanning the services and protocols in the credit check service.

Currently, a SAML token is attached only to SOAP protocols. It would be useful to extend it to other native protocols such as JMS, SQL*net/ODBC, or even Inter-ORB. Ideally, all SOA-protected resources should be able to leverage SAML tokens. These protected resources should also be able to use an identity management Single Sign-On (SSO) server to determine access rights based on the tokens. When a policy is changed in the SSO server, it would affect all of the components that use it for security decisions.

Looking Ahead: Identity Delegation

Complex business processes that are found in SOAs need the ability to delegate identities. Delegation is essential when a party needs to vouch for another party – for example, when a corporate buyer makes a purchase on behalf of his company. In this case, his company should vouch for him. This means that instead of using the employee's private key for cryptographic operations, the company's private key is used. Typically our fictional corporate buyer invokes a local procurement application (such as the one we used in our portal example), fills out a purchase order, and prompts the application to send the purchase order. The application uses the original user's credentials as is or maps them to another identity format such as a SAML assertion that will be inserted into the WS-Security header. The application certifies the request by providing its own cryptographic

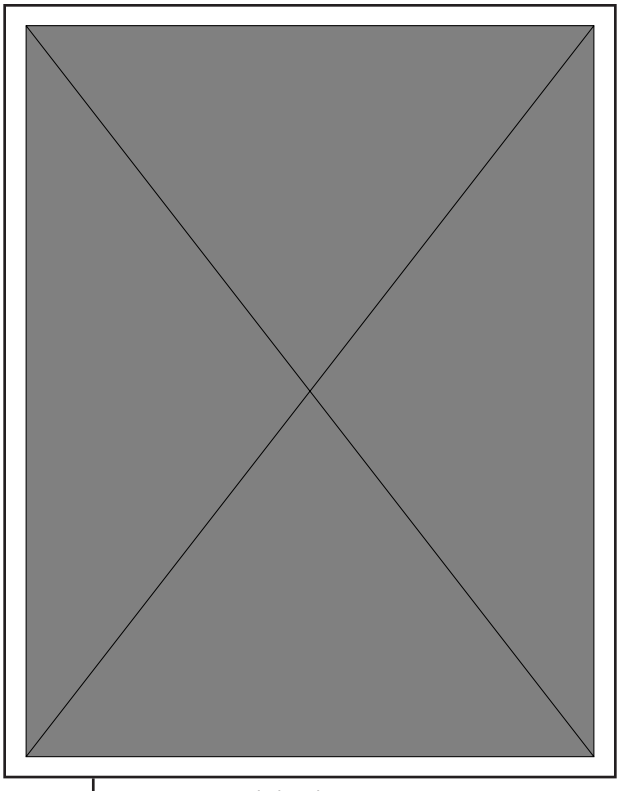


FIGURE 7 SAML assertion with digital signature

> “Currently there’s no open standard that completely addresses the issue of propagating identities across the multiple services in a SOA”

key (for example, the company’s private key or shared secret), which is where the delegation takes place. The application posts the purchase request to a purchasing Web Service at the provider’s site. The Web Service then authenticates and authorizes the request based on the information in the SAML assertion (see Figure 7).

Some scenarios like this have been implemented. However, part of the solution relies on standards – HTTP, SOAP, WS-Security, SAML, and possibly additional Web Service specifications such as WS-Trust if security token brokering is involved – and part of it consists of proprietary extensions to implement identity propagation and delegation. For example, there’s currently no standard way to express delegation. Existing standards such as SAML assume that the owner of a security token is fully responsible for the security process. As a result, delegation should be designed into the SAML standard. Likewise, other than SOAP, there’s no standard way to bind a SAML assertion to other prevalent SOA protocols. Standards bodies must work to profile SAML token usage for various SOA protocols and transports.

Conclusion

SOAs are full of complex business processes, often traversing multiple services and protocols. One of the challenges in this environment is to propagate identities across these services. In fact it’s a necessity in today’s age of compliance. Companies must be able to prove who has access to their services. Also, to have truly secure and auditable business processes in SOAs, you need a way to propagate identities. If a transaction spans multiple services, an identity must be bound to a payload and be able to span the service calls from beginning to end.

Currently, there isn’t an open standard that completely addresses this issue. We hope the next version of SAML will do so. SAML needs to have robust delegation capabilities and binding profiles added to its resume. Once these are added, IT organizations will have the tools they need to enable identity propagation throughout their SOA business processes.

About the Authors

William Bathurst is a senior product manager at Oracle with 18 years of industry experience. He is currently the product manager for J2EE security and web services management.

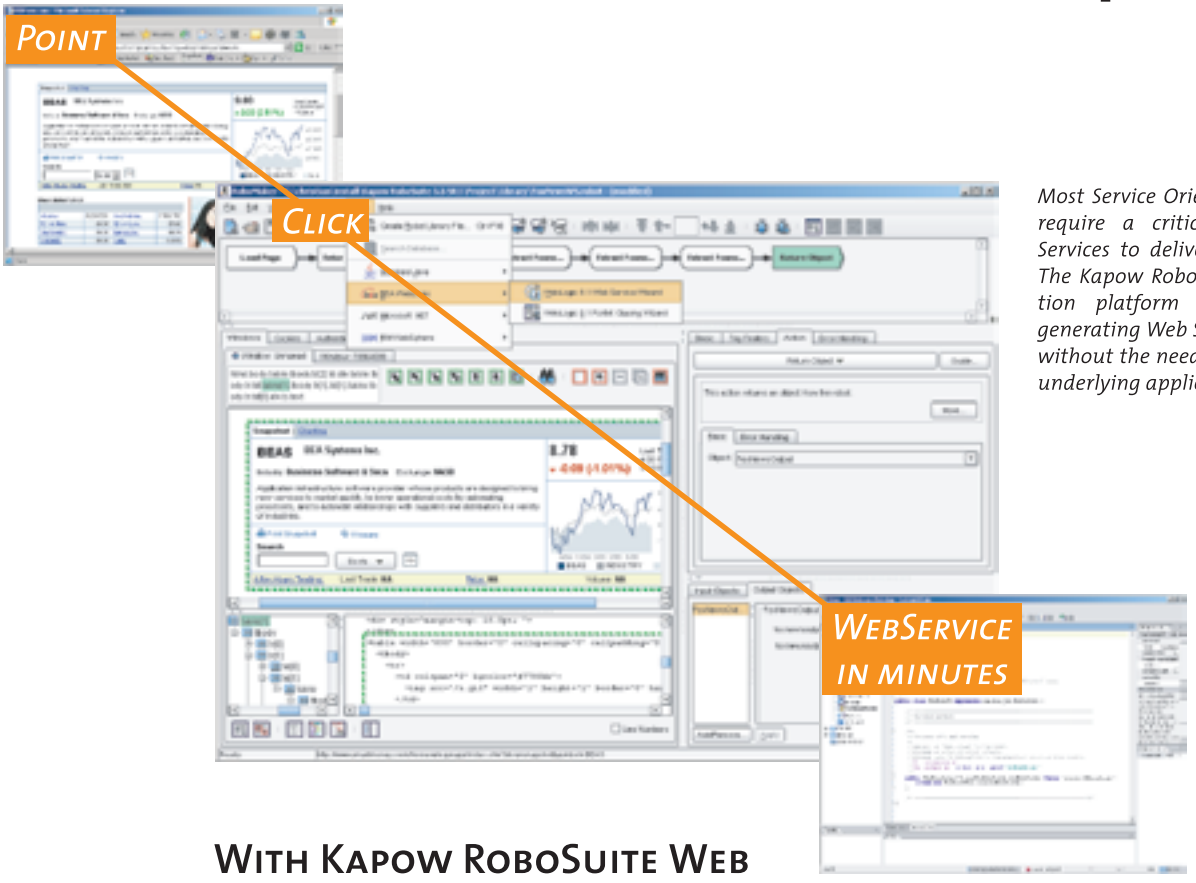
Ramana Turlapati is a consulting member of the technical staff at Oracle with 12 years of industry experience. In his current role as the security architect for Oracle Web Services Manager, he contributes to Oracle’s overall Web Services security strategies and solutions.

Marc Chanliau has been in the software industry for more than 20 years and is currently a director of product management at Oracle where he is responsible for identity management solutions and innovations. Mark is heavily involved in security and XML standards groups including serving as the first chair person of the OASIS Security Services Technical Committee (SSTC), which culminated in the adoption of SAML as an official OASIS standard, participating in the WS-Security Technical Committee and helping to define the Liberty Alliance 2.0 specifications.

Leverage existing IT to generate Web Services

[in minutes]

For free online demonstration, Gartner "Cool Vendor" release, and software trial: www.kapowtech.com/wsjprint



Most Service Oriented Architectures require a critical mass of Web Services to deliver tangible results. The Kapow RoboSuite Web Integration platform supports this by generating Web Services in minutes without the need to re-program the underlying application.

With Kapow RoboSuite Web Integration Platform you get

- *Fast generation of Web Services of any web application – simple or complex – with visual point and click*
- *Non-intrusive solution using the browser front-end to access any web-application*
- *Wizards enable quick, low cost deployment in SOA frameworks for production-ready SOA*

kapowtech.com

kapow TECHNOLOGIES

Kapow Technologies is a leader in Web Integration – a new integration paradigm using the broadly available web front-end. The Kapow RoboSuite platform uniquely enables flexible and fast integration of content, data and applications from any source available through a browser into portals, content management systems, applications, databases or web services.

Does a Web Service Make a Service for SOA?

SOA Service & SOA SLA as drivers for renovating legacy applications for SOA

What could be easier than to take your application, wrap it with a Web Service, announce it or register it in the UDDI and get a SOA Service? Even better – take a data warehouse, cover a SQL executing code with a Web Service and expose it to SOA, isn't it simple? This article is for those architects and managers who like such "simplicity." If you believe that a Web Service itself doesn't convert an application into the SOA Service, you might read the article just out of curiosity.

A Service or Not a Service

WRITTEN BY
MICHAEL POULIN

Discussions about Service Oriented Architecture (SOA) initiated by people working with legacy applications and data storages like Data Warehouse (DW) have gotten a lot of press attention recently. While it's good for SOA's popularity the discussions typically declare a few SOA characteristics and say something like "We have so rich/important/business crucial data and, if we just expose it into a SOA, it will be great opportunity for us." An intention of exposing DW to a SOA sounds suspiciously like the five-year-old movement of exposing all applications to the Internet. Have we forgotten that mistake already? Have we understood the difference between Web applications and Web-enabled applications? Did anybody count the man-centuries spent in IT to convert and modify existing applications to make them really work in the Internet environment?

When I participate in such discussions, I usually ask one question: "To make some ground for DW working in SOA, please tell me what is a service in DW?" I am given two types of answers: either "none" or "We do a lot of data transformations and/or perform sophisticated data aggregation and produce business intelligence reports out of the DW." Well, data manipulation procedures performed in the DW might be considered as services, however, I have not found a definition of DW that would describe a service as a part of it. That is, DW wasn't designed for services.

Anyway, what's so special about a "service"? Why is wrapping access to a DW or an application with a Web Service not enough to obtain a service? What has to be done, if anything, to make such applications work in a SOA? Here I'll try to identify a process that could take place when one prepares a legacy application to support a Service for SOA.

Approach to a Definition of a "Service"

In different spheres of human activity, there may be many definitions of a service. My understanding of service behavior with regard to software components is an action/activity performed by the service provider for the service consumer in accordance to a contract between the provider and the consumer.

While contracts may vary, the most flexible type unties/decouples/isolates a consumer from a provider. In the business, as we know, the consumer pays for the service and tends to consider a service provider a servant. This leads to two conclusions:

- a consumer looks for a provider with the contract that meets the consumer's goals.
- a consumer can agree with some of the constraints a contract puts on him if the contract still meets the goals and decouples the consumer from the provider's internal conditions.

The conclusions point to the difference between using a software component as a traditional application and using it as a service. In the former case, a consumer depends on the application specifics and, if something isn't suitable, the consumer has to adapt and wait for the application to be refined. In the latter case, a consumer deals with a service application only if the contract meets his needs and the application constraints are reasonable. For example, the constraints can provide some business values such as security, which adds business trust, or remote invocation, which can lead to service scalability and robustness.

Jumping ahead let me say that one SOA service can engage another SOA service doing this transparently to the consumer. If the second service provides the proper data based on an "update schedule", i.e., the data are obsolete for some period of time, the first service tends either to find a substitute service with the correct data for that period of time or switch to a totally different service, without any "schedule problems." In any case, the consumer shouldn't depend on that schedule. Otherwise, all service contracts have to reflect one service specifics, which leads to quite an insufficient architecture and, actually, couples services and consumer together. Described is not a SOA rule, it's a business rule and due to SOA agility principle, SOA promotes the same behavior.

Thus, the service contract can be used as a requirement when one transforms the application into a service. A well-known expression of a contract is a Service Level Agreement (SLA). There may be a single SLA for all consumers or the provider can maintain individual SLAs

with every consumer. If we took a typical SLA used for DW, for instance, and a SLA used in SOA, we'd get the starting and ending points for the process we have to implement when exposing a DW to a SOA.

The process includes not only a new connectivity interface but changes in behavior dictated by the Service SLA. For immutable applications, the process assumes building a service façade layer to interact with other services and consumers.

Service Interface in SOA

SOA implies certain requirements in the service interface. In short, the interface has to be:

1) stable
2) self-describing
3) independent from provider implementation and its resources
4) registerable/searchable
5) accessible programmatically
6) versioned
7) accessible under control (security)

SLA Section	Features
Functionality	• Values/functions provided – for example, "regular fund price" and "updated fund price" • Service Interface – WSDL or WSDL-less like SOAP, IDL, HTML, servlets, JMS, and .NET • Version compatibility – compatibility of the interface and service function(s); version control policy and procedures
Operational aspects	• Service availability – "24/7/365" or "on schedule" with a certain amount of downtime • Service Change Control protocol – how a new version of the service is made available with and without backward compatibility in its interface and functions • Service cost, if any, for the consumer or license model, if applicable
Quality of Service	• Conditional performance expectations • Scalability – horizontal (serve more consumers) and vertical (process more data) • Failover, execution exception handling protocol, and alerts • Behavioral targets – behavioral patterns that are recognized as 'normal' in given scenarios of the Service engagement • Business process exceptions – technically legitimate results that are considered abnormal/exceptional in a given business process • Expected response time • Broadband/throughput • Custom characteristics – like a quantity of processed purchase orders versus return orders
Quality of Data	• Returned data – a list of the data items • Metadata about returned data – the definition of the format (with possible conditions), source, name space, and validity of the data
Service Constraints	• Security policies – for users and other services • Country/industry policies – for instance, regulations on the information encryption algorithms • Pre-known conditions – like delivery schedules: pick hours and night processing • Service priority – may be different for different customer status

TABLE 1 | An example of a SOA Service SLA

As you see, there's nothing about the Web in the requirements. Indeed, any particular interface implementation is suitable if it meets the requirements. It may be, for example, CORBA IDL. As we know, many elements of Web Services specifications were derived from CORBA. So, why is a Web Service (WS) as an interface so attractive for SOA but still not enough?

Web Services technology defines the abstraction of service behavior via standard WS Description Language (WSDL). It also provides for the abstraction of data via XML and the metadata definition via XML Schema. WS technology offers a WS registry known as UDDI (Universal Description, Discovery, and Integration) and a set of standards to support security (WS-Security and related standards), transaction (WS-TXM, WS-Coordination, WS-AtomicTransaction, and WS-BusinessActivity), aggregation and management (WS-BPEL and WS-CDL), and interoperability (WS-I Basic Profile).

Agreement between leading technology vendors on XML and most WS standards makes Web Services really the best candidate for a universal interface for SOA. Some specialists include SOAP (Simple Object Access Protocol) in the basis of SOA while others disagree with it and consider SOAP only one of several possible and convenient protocols for WS binding. SOAP alone simply doesn't provide enough abstraction for a service definition, that's why we need WSDL and we already know a few WS models that don't use SOAP binding.

What's not included in WSDL is the information considered by a consumer when he makes a business decision about the service. That is, you can automatically invoke a WS but you're not supposed to do so blindly, without estimating the inherited risk. It's simple: if you want to cook some French fries, you'd not use a pan without checking if it burns oil. The same relates to WS – the service may be accepted if you know its connectivity interface and its quality characteristics like performance, change control rules, error-handling scenarios, and so on. This information is usually described in the SLA. So a SLA can be treated as a container that includes business knowledge and service interface definition – both to meet the consumer's needs. A lot of legacy systems also work on the basis of SLA, however, those SLA are traditionally provider/application-centric.

Service Level Agreement for SOA Service

Currently, some efforts are made in the industry to define and formalize SLA for WS – these are specifications for Web Service Level Agreements (WSLA) and the Web Service Offerings Language (WSOL). The WSLA provides a structure of definitions of basic SLA elements. While it is a subject for separate article, we note the fact that the WSLA pairs SLA parameters with their metrics. That is, providing for SOA SLA means constant monitoring, analysis, and compliance reporting of actual runtime parameter values that aren't a part of most legacy application SLAs.

Though a SOA SLA isn't standardized yet, it's very important to understand what can be included in it. Table 1 gives an example of a SOA SLA. Some parameters are measurable (as represented in WSLA) and some of them aren't.

There are no mandatory or optional parameters in the SLA because they are all business- and context-specific. As mentioned already, neither Web Service/WSDL nor any other programming interface can address such SLAs. On the other hand, not all services require such detailed SLAs. Its can be periodically refined and extended when more objective information

> A Web Service isn't enough to constitute a service in a SOA

is gathered about the Service. The SLA demonstrates that participation in SOA requires both – the service interface and service behavior.

The only other thing I'd like to note is that it's better for maintenance and further enhancements if all services in your system have a unified SLA, especially when the number of Services is expected to grow over the time. Otherwise, managing the Services quickly gets out of hands. That is, the better you define the Service, the fewer problems you have at runtime and the more consumers you attract to use it.

Thinking in SOA

We can find tons of publications describing how to wrap a legacy application with a Web Service. Many serious works recommend the same solution – the most scalable and flexible way of wrapping is to create a thin layer on top of the legacy application. The "preserve-and-extend" approach has gained a reputation as being the most reliable and cost-effective model for dealing with business-critical legacy apps.

Some vendors propose developing connectors and gateways where the former run in a mainframe and latter outside the mainframe (screen-scraping). Both models act as proxies shielding actual applications and helping to provide SLA. Merrill Lynch, in particular, has created an integration platform using plug-ins with parsers, metadata assemblies, and drivers that runs on the mainframe and exposes legacy apps through Web Services via HTTP and MOM protocols. This platform is even called a Service Oriented Legacy Architecture. However, many experts suggest that transitioning a legacy application to a Service for a SOA isn't that straightforward: simple wrapping just "preserves" the application but doesn't make it a "first-class SOA citizen" without an "extension."

Let me give you two examples. The first one is about a traditional application in the financial industry; it may be considered a base line for the second example. Assume that the task is to calculate the credit risk of swap transactions (a special type of financial transactions). The transactions comprise the transaction data themselves, related financial streams, and cash flow data. These are dynamic elements that usually persist in an operational data store. Other related data about supply feeds and financial clients are static and usually persist in a reference data store.

The traditional application-centric approach of building a credit risk calculation component addresses the access interface(s)–access protocol(s) and defines data location, the data access mechanism (direct SQL, Stored Procedures, Views, security controls, etc.) and the data availability schedule. The team developing the application has to deal with its consumers (dictating application constraints) and constantly negotiate data status and updates with the database maintenance team. It leads to per-application specialized code that has to transform data to meet particular application needs. Any change in the data affects the application and all its consumers (via production re-deployment, at least). Such complex daily management task may be sufficiently executed if the team has enough resources and deals with only a few applications but this is a dream nowadays. A more realistic consequence is a shortage of resources and degraded quality, as well we know. And, there's no room left to support business growth.

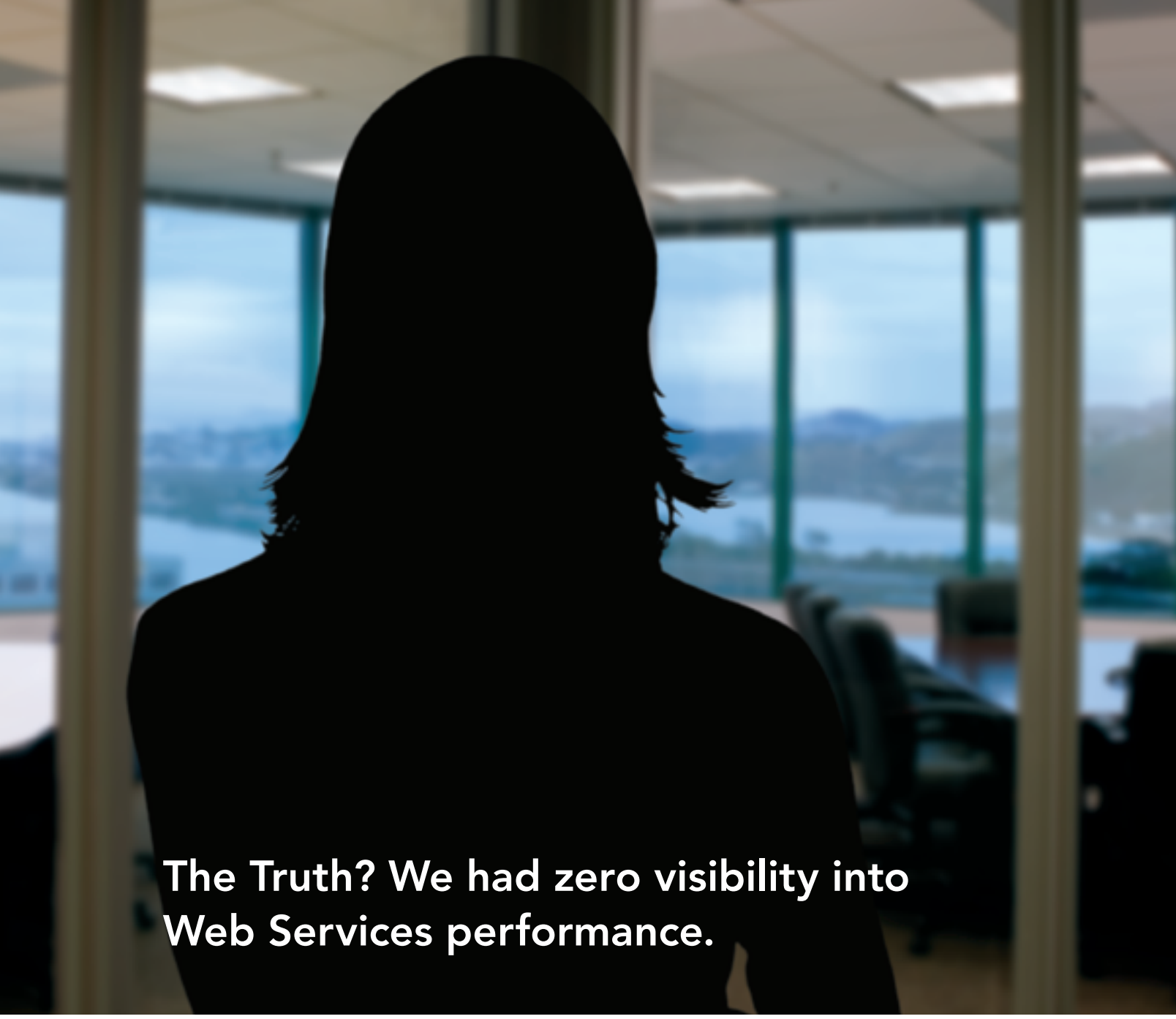
A service-oriented approach defines the credit risk calculation engine and the metadata that the engine can work with. That is, the engine knows about different calculation formulas and intermediary data storages, if needed; it also knows how to deal with data if it meets metadata requirements. That's it. The consumer of the calculation service specifies what calculation to perform – for intra-day or end-of-day transactions. The input data is provided by another service, e.g., the Data Access Service (DAS), which also knows where to place calculated results. The calculation engine hasn't a clue where to get data from, what transactions to process, when to do the job; it only cares whether the DAS provides a SLA and acts appropriately if the SLA is violated. Now every development and support team can concentrate on its own business and provide related SLAs.

Certainly we have to balance complexity and flexibility: how granular should a Service be to avoid unnecessary communications and function compositions? Is it always reasonable to extract data access from the business service and put it into separate specialized services? Answering these and similar questions, one caution has to be observed – never consider a solution in isolation from the environment. For instance, application developers believe that calling a database directly via JDBC/ODBC guarantees maximum performances. It's true, but if the deployment environment requires a security authorization control for a particular database column or raw, your application is responsible for implementing and supporting such a control. Would you prefer coding this yourself or using an existing Security Service? That is, are you sure the performance you can provide would be better than a professionally developed Security Service? If you choose the latter, you can easily find that a Data Service is already integrated with the Security Service and, probably, is the way to go. What will happen to overall performance when you use the Data Service? May be nothing bad if you consider the Services and optimize your application design upfront.

Now is the time to ask: can our legacy application act as a service (an aggregation of services)? The answer may be "yes" if we transform the application with a service-oriented model. It's a real opportunity if we can build or rebuild the application from scratch. Usually, the application is crucial for the business and/or not easily modifiable. Even in that case it's not a lost investment but it will be if the application can't keep up with the architecture evolution. For demonstration purposes, we'll describe a gradual transformation of a data store into a SOA Service.

From the Data Store to the Data Service

We'll start with a real-life example happened in a financial company providing support for a 401(k) plan over the Internet.



Wily can handle the truth. We've helped hundreds of organizations around the world manage critical Web Services and proactively identify problems before customers are affected. Using a single solution, you gain control over both your application infrastructure and the customer experience. To learn more about how Wily can help you achieve customer success with Web Services, visit **truth.wilytech.com**.

Get Wily.™

Enterprise Application Management

wily technology™

Its customers complained that they couldn't see some of their investments. The errors occasionally appeared and were reported by the end users. It took several days for the issue to work its way up the management line and for developers to locate the related data store. The data store provider worked on the basis of a regular service agreement that said that certain data had to be sent via a Web Service to the Web site component on-demand. The data was always sent, but sometimes one piece of data filed contained a NULL. For the data store, it wasn't a mandatory field and it could be a NULL. The service agreement didn't say anything about the quality of the data when the Web site was developed on the supposition that the data field was available and had a "non-null" value (as represented by the data store in the example).

The request to the data store to change the field constraint to avoid NULL was denied based on the fact that a lot of other customers considered the field optional. The behavior of the data store team was very typical – "We're glad to support SOA and we give you all that we have." Oops, there's the problem: the end users and the Web site weren't interested in what the data store could or couldn't do but in the correct portfolios. It's simple – if you want to wash you favorite Hawaiian shirt and the laundry only has bleach, will you use that service or go to another one?

Let's look at the situation from the data store's side and assume it's serious about providing a SOA Data Service. Figures 1 and 2 demonstrate possible actions the data store's management can take to transit the data store into a data resource for the Service. The transition takes two phases – analysis and execution.

The analysis phase can be organized in two ways depending on the demand for the Service: a) the Service is really required and its customers can be identified; b) the Service is just a proposal intended to facilitate customer demand. Activities involved in the analysis phase are applicable in both cases with light modifications. In our example, the data field's requirements are already identified. Further activities can be:

- Identify all dependent data and estimate the impact if the metadata has to be modified
- Identify the sources of the data and related SLA, if applicable, i.e., the quality of the available data
- If the Data Service has to engage other Services, review the relationship with the providers of the potential helper Services and related SLAs
- Define a basic SLA for the Data Service to meet the requirements; there may be multiple SLAs for different customers but they can't be contradictory
- Identify the customer community and its dynamics
- Identify policy-based constraints on the Data Service (security, accessibility, internationalization, etc.)
- Identify the available or needed software and hardware.

Based on the results of the analysis, we can determine if a data transformation is needed for a particular SLA. This may mean modifying the data field metadata, particularly the database constraint that restricts the value from being equal to NULL. It's also very important to recognize all the risks – operational and programming – associated with the data and the Data Service for adequate Risk Management and compliance with corporate and industry regulations. The findings will drive the Execution Phase.

The execution phase aims at what and in which order things have to be done, when intermediary decisions have to be made, and what the controls have to be implemented and preserved in the transition process. This phase ends in actually implementing the plan. The common rule is – when implementing a SOA Service or orchestrating a Service execution scenario, it's not always necessary to exclude human intervention. First, it may be too costly to automate everything and second, because orchestration standards, e.g., BPEL, let long-living transactions integrate human actions into the service process.

Well, we assume that the metadata for a selected data field can't be modified right away. It requires a grace period to take care of all customers of the data. So, a temporary Transition Data Service may be a solution here.

We can also create an intermediary data field that meets the Data Service requirements identified in the analysis phase. The mechanism of refreshing an intermediary data field is also defined (it may be based on a schedule or on a value-change event initiated by a manual operation). The Transition Data Service binds the intermediary data field as its data source.

The Transition Data Service has one specific: it provides the data with an indicator of "freshness" because the intermediary data field isn't updated with new data if the data doesn't meet the Service SLA. For example, if the Transition Data Service provides a Mutual Fund price and the latter has been set to NULL for today (e.g., the real price wasn't calculated on time, by the deadline), the Transition Data Service will show the Fund price as a "yesterday" price but not as NULL.

Then we have to migrate existing customers of the selected data to the intermediary data field even if they don't use a Data Service. It's an operational process and it can take some time. This "delay" may even be good for

Webopedia – SOA definition: Abbreviated SOA, an application architecture in which all functions, or services, are defined using a description language and have invokable interfaces that are called to perform business processes. Each interaction is independent of each and every other interaction and the interconnect protocols of the communicating devices (i.e., the infrastructure components that determine the communication system do not affect the interfaces). Because interfaces are platform-independent, a client from any device using any operating system in any language can use the service.

Though built on similar principles, SOA is not the same as Web services, which indicates a collection of technologies, such as SOAP and XML. SOA is more than a set of technologies and runs independent of any specific technologies.

Webopedia – Data Warehouse definition: Abbreviated DW, a collection of data designed to support management decision making. Data warehouses contain a wide variety of data that present a coherent picture of business conditions at a single point in time.

Development of a data warehouse includes development of systems to extract data from operating systems plus installation of a warehouse database system that provides managers flexible access to the data.

The term data warehousing generally refers to the combination of many different databases across an entire enterprise. Contrast with *data mart*.

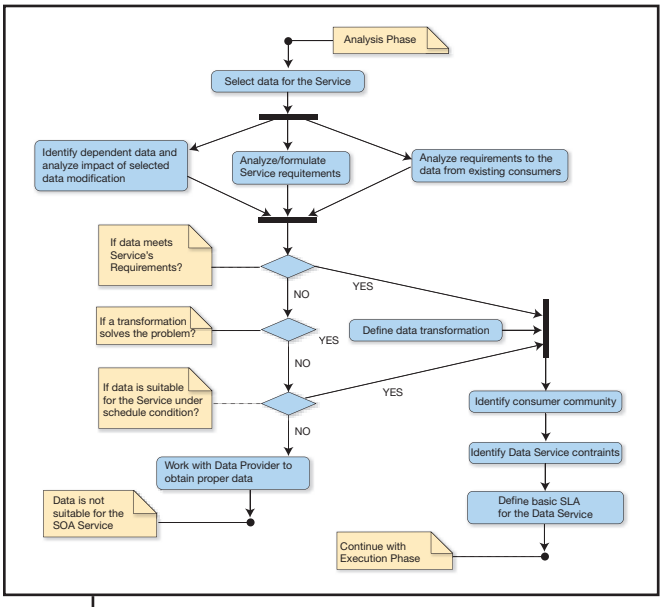


FIGURE 1 Analysis phase of a transition into the SOA Service

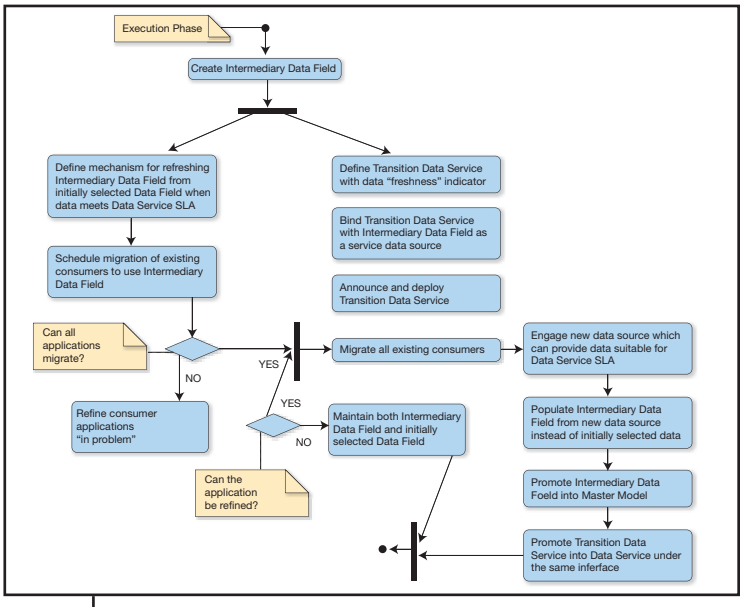


FIGURE 2 Execution phase of a transition into the SOA Service

the Data Service because it gets time to demonstrate its advantages. Simultaneously with the migration effort, we have to find either another data source that always has current data or work out this issue with the existing data provider.

When all the customers start using the intermediary data field and we get a proper data source, we promote the intermediary data field to a Master Data store and retire the initially selected data field. The Transition Data Service now becomes a fully scaled Data Service. Unfortunately, it's not always possible to get rid of the initial data field because some immutable legacy applications couldn't migrate. But it's not a hopeless situation: those legacy applications can be temporarily buffered with a SQL substitution component (assuming that there are only a few such applications left). Technology evolution dictates that data source providers improve their services and eventually the inappropriate data will go away. This will enforce the legacy applications either take the new data or retire themselves.

Conclusion

The notion of a Service in a Service Oriented Architecture goes far beyond the definition of a new interface even if it's a Web Service interface. Though the service interface is very important, the service provider has to provide the service, not the access to an application whose architecture and functionality may be inadequate for the service's behavior.

We reviewed a few service characteristics and identified Service Level Agreements as an instrument for effective service interoperability and as a driver for service-oriented transformation in a service provider architecture. The obvious conclusion that a Web Service isn't enough to constitute a service in a SOA was demonstrated by a real-life case of transition of a regular data store into a Data Service provider for SOA.

References

1. Sutor, Bob. "Something Old, Something New: Integrating Legacy Systems" http://www.ebizq.net/topics/legacy_integration/features/5229.html?&pp=1
2. Web Service Level Agreements. http://www.research.ibm.com/wsla/WSLASpecV1-20030128.pdf
3. Schmelzer, Ronald. "What Belongs in a Service Contract?" http://searchwebservices.techtarget.com/tip/1,289483,sid26_gci1120180,00.html
4. Meehan, Michael. "HP looks To Give Legacy an SOA Upgrade" http://searchwebservices.techtarget.com/originalContent/0,289142,sid26_gci1144084,00.html
5. Teubner, Russ. Integrating CICS Applications as Web Services. SWSJ, http://webservices.sys-con.com/read/39850.htm
6. Integrate existing assets and create new functionality. http://www.softwareag.com/Corporate/products/cv/leg_int/default.asp
7. Application Modernization & Legacy-to-SOA. http://www.interactive-objects.com/solutions/application-modernization/legacy_modernization_eng.pdf
8. Poulin, Michael. "Entitlement to Data" JDJ, Vol.10, issue 12, 2005 http://java.sys-con.com/author/poulin.htm
9. Service Oriented Legacy Architecture – SOA CICS. http://www.soa.com/index.php/section/products/sola/
10. Business Process Execution Language for Web Services. http://www-128.ibm.com/developerworks/library/specification/ws-bpel/
11. Web Services-Interoperability Basic Profile. http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/cwbs_wsiprofile.html
12. Web Services Security (WS-Security). http://www-128.ibm.com/developerworks/library/ws-secure/

About the Author

Michael Poulin is working as a technical architect for a leading Wall Street firm. He is a Sun Certified Architect for Java Technology. For the past several years Michael has specialized in distributed computing, application security, and SOA.

m3poulin@yahoo.com

WSDL 2.0: A Pragmatic Analysis and an Interoperation Framework

Minimizing interoperation issues with a WSDL version management framework

Web Service Description Language (WSDL) represents an IDL describing the contract between the service requestor and the service provider in much the same way that a Java interface represents a contract between client code and an actual Java object. The crucial difference is that WSDL is platform- and language-independent and used primarily (although not exclusively) to describe SOAP services.

WRITTEN BY
SENTHIL K M, SHAURABH BHARTI, ANSHUK PAL CHAUDHURI, AND SRINIVAS PADMANABHUNI

The WSDL 1.1 specification has been accepted at the World Wide Web Consortium (W3C) and is the predominant version for describing Web Services today. At W3C, work on the next generation of WSDL, i.e., version 2.0, has been under way for some time now. WSDL 2.0 promises to describe not only traditional SOAP Web Services, but also a wide variety of services provided over any network. W3C has published the Candidate Recommendation for WSDL 2.0. WSDL 2.0 is substantially different from WSDL 1.1. In this article, we'll dissect the WSDL 2.0 spec and point out the overall differences and some core benefits of the WSDL 2.0 specifications. We'll also point out the interoperability issues with WSDL 1.1 and detail InfWVM, a framework for seamlessly working with WSDL irrespective of the version.

The Structure of WSDL 2.0

There are significant differences between WSDL 2.0 and WSDL 1.1. WSDL 2.0 has three specifications:

- Core, which explains the abstract interfaces independent of protocol and encoding;
- Message Exchange Patterns (MEP) with predefined types of interactions; and
- Bindings pertaining to SOAP and HTTP.

The description of a Web Service in WSDL 2.0 is structured into two parts. In the abstract, WSDL describes a Web Service in the form of the messages it propels and receives through a type system. Message Exchange Patterns (MEPs) define the sequence and cardinality of the messages. An operation associates Message Exchange Patterns with one or more messages. An interface groups these operations in a transport- and wire-independent manner.

In the concrete element of the description, bindings denote the transport and wire formats for interfaces. A service endpoint associates the network address with a binding. Finally, a service clusters the endpoints that implement a common interface. Figure 1 shows the conceptual WSDL component model.

Further, different operation styles and message exchange patterns can be defined and a wide variety of message definition languages can be used in addition to XML Schema, with messages bound to any protocol for which a binding specification has been written.

Modifications in WSDL 2.0

Listed below are some of the major modifications introduced in WSDL 2.0 specification

- WSDL 2.0 is infoset-based, which is more precise and heavily oriented to a component-based structure.
- An XML Information Set (Infoset) provides a reliable set of information definitions, with up to 11 different types of information items, as explained in the W3C specification at http://www.w3.org/TR/xml-infoset/. There is no requirement for an XML document to be valid to have information set. WSDL 2.0 provides a set of components and their associated properties for describing Web Services.
- WSDL 2.0 includes additional semantics. Hence, targetNamespace is a required attribute of the description element in WSDL 2.0, as shown in Listing 1.
- WSDL 2.0 adds the capability of including documents. Different service definition components of the same target namespace are allowed to exist in separate WSDL documents (which can be used across various service descriptions) using of include element.
- WSDL 2.0 supports additional Message Exchange Patterns or patterns. WSDL patterns define the sequence, direction, and cardinality of abstract messages sent or received by an operation.
- WSDL 2.0 has introduced the idea of Interfaces, replacing PortType. The interface operation element has name and pattern as required attributes.
- The concept of interface inheritance is also introduced in WSDL 2.0. One interface can extend one or more interfaces (multiple inheritance). This adds one more layer of reusability.
- WSDL 2.0 introduces the concept of features.The presence of a feature component in a WSDL 2.0 description indicates that

"Everyone wants faster, better, cheaper. We thought they might appreciate smarter, too." Entering the integration market we had three advantages: we knew the costs and hassles of traditional integration solutions were limiting their adoption, we saw that a standards-based service oriented architecture would solve these problems, and we had the world's most scalable enterprise messaging server, SonicMQ®, as a core technology. We combined SonicMQ's performance and security with Web services, XML transformation, intelligent routing and a new distributed deployment and management infrastructure to develop the world's first Enterprise Services Bus, Sonic ESB™. With it businesses can easily integrate existing and future applications to create unprecedented business agility, and they can start today knowing they can scale to meet tomorrow's needs. We call it incremental integration. It's smarter. It's also faster, better and cheaper.

sonic
SOFTWARE®
CONNECT EVERYTHING. ACHIEVE ANYTHING.™

Gordon Van Huizen, Sonic Software

www.sonicsoftware.com

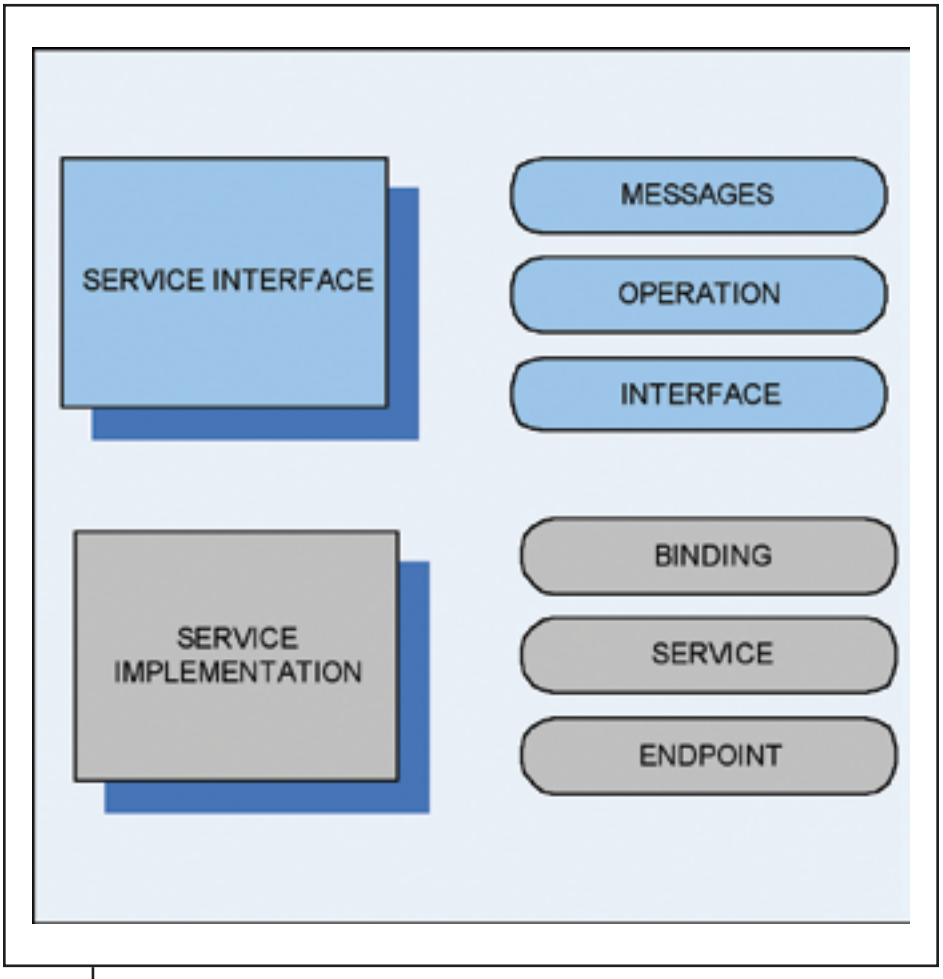


FIGURE 1 | WSDL 2.0 structure

the service supports the feature and may require that a client that interacts with the service use that feature.

- WSDL 2.0 supports the use of other type systems (it already supports XML Schema). Schemas are embedded into the types element of the WSDL document. Constructs in semantic models such as classes in OWL ontologies could be used to define the Web Service input and output data types.
- WSDL 2.0 also introduces the idea of operation style to apply restrictions to an operation.

Addressing WSDL 1.1 Limitations

Listed below are some of the enhancements in the WSDL 2.0 specification that address key gaps in WSDL 1.1.

- All WSDL constructs have been made extensible via both elements and attributes. WSDL 2.0 provides two extensibility mechanisms: an open content model thatallows XML elements and attributes from other (non-WSDL) XML namespaces to be interspersed into a WSDL document and via features and properties. Both mechanisms use URIs to identify the semantics of the extensions. For extension XML elements and attributes, the namespace URI of the extension element or attribute acts as an unambiguous name for the semantics of that extension. For features and properties, the feature or property is named by a URI.
- Legacy type definitions like DTD and RelaxNG, which have been in the industry a long time, have been made reusable with support for these definition types in the *type* definition. It gives an additional level of support for other semantic concepts in the future, e.g. OWL classes. This addresses a key restriction in WSDL 1.1 in the type definition.
- The lack of interface reusability was one of the major drawbacks of WSDL 1.1 that's addressed by WSDL 2.0. In WSDL 1.1 if we need to deploy an operation across multiple interfaces, it has to be replicated. Even the interfaces declared across different documents can be reused in the existing documents via import or include mechanisms in WSDL 2.0.
- Multiple services can be declared in the same document, which helps to explore more than one service out of a single WSDL.
- Previously, since WSDL 1.1 wasn't component-based, instances were used as singletons. Moreover, the component-based WSDL 2.0 has added more compatibility and support for Grid Services, based on a specialized GWSDL, which were started based on WSDL 1.1.
- Ambiguity in defining operations has been removed by making Qname, i.e, (namespace, local name) tuple unique.
- The inadequacy of the fault description supported by SOAP 1.1 is addressed and replacedment by the SOAP 1.2 fault format.
- In operations, WSDL 1.1 had no mechanism to enforce any restrictions, a shortcoming that is also addressed by 2.0. It includes the concept of operation style to apply restrictions to an operation. An operation style asserts that message schemas in a particular operation conform to certain rules.

Even though 'import' provides a mechanism to get the definitions from a different target space, there was no mechanism available to get the definitions from the same target space. This has been addressed by 'include' in WSDL 2.0

- WSDL 1.1 had hardwired support for Message Exchange Patterns (MEPs). Only four types of MEPs were supported:
  - Input-output
  - Output-input
  - Input only
  - Output only

These MEPs corresponded roughly to thefollowing traditional interaction patterns:

  - Request-Response
  - Publish-Subscribe

– One-way invocation
– Event

WSDL 2.0 supports additional MEPS or patterns. It supports the following eight patterns:

1. In-only –> consists of only a single message.
2. In-out –> consists of an input message followed by an output message.
3. Request-Response –> is identical to an in-out pattern except that the in-out message travels on the same channel.
4. In-multi-out –> consists of an input message followed by one or many output messages.
5. Out-only –> consists of only an output message.
6. Out-in –> consists of an output message followed by an input message.
7. Out-multi-in –> consists of an output message followed by multiple input messages.
8. Multicast-solicit-response –> consists of an input message followed by one or two input messages.

- WSDL 1.1 didn't support any mechanism for capturing the non-functional requirements of a service. WSDL 2.0 features are introduced for annotating with non-functional or specialized characteristics. WSDL 2.0 features might include "reliability," "security," "correlation," and "routing" related details. The presence of a feature component in a WSDL 2.0 description indicates that the service supports the feature and may require that a client that interacts with the service use that feature. Each feature is identified by its IRI. WSDL2.0's feature concept is derived from SOAP 1.2's abstract feature concept. The feature looks like Listing 2:
- The SOAP binding for WSDL 2.0 is much simpler and it's easier to write bindings than with WSDL 1.1. The HTTP and XML binding in WSDL 2.0 has also been improved.

Interoperability Issues with WSDL 1.1

Adopting WSDL 2.0 will lead to interoperability issues with existing WSDL 1.1 implementations. Most Web Service implementations are based on WSDL 1.1 on both the producer and consumer side. Interoperability issues with respect to the adoption of the new standard can arise in two ways. First, issues related to exposing already hosted services through the WSDL 2.0 specification and second, issues related to the consumption of the services from the client side using the WSDL based on the WSDL 2 specification. The latter seems to be more difficult to tackle than the former.

In the case of exposing an already existing service using the WSDL 2 specification, one might just have to convert the old WSDL to the new WSDL 2 specification and re-host the WSDL. Since the SOAP engine on the sever side hosting the service doesn't require the WSDL to do its processing, there seems to be no interoperability issues associated with it. Nevertheless, most SOAP engines support automatic generation of the WSDL once a service is deployed. Hence there might be a need to generate WSDL based on the WSDL 2 specification. But this component will be independent of other SOAP engine components and hence can be made available as a plug-in component from SOAP engine vendors and plugged into any of their existing versions of the SOAP engines.

In the latter case, from a consumption perspective, clients consuming a service using the proxy stubs generated from the WSDL 2.0-based spec require additional work. If a service is exposed through WSDL 2, then the issues clients will face in consuming the service are:

1. The commonly available WSDL 1.1-based proxy generators can't be used for generating the proxy stubs from WSDL 2 necessitating WSDL 2 proxy generators
2. Custom WSDL 2 proxy generator tools might not adhere to standards and can aggravate interoperability issues further.
3. Tight coupling of the WSDL specification version on the client side for consuming the service will lead to interoperable issues as new versions of WSDL are released.

Mitigating Interoperability Issues with a WSDL Version Management Tool

A mechanism to avoid such interoperability issues is to reduce the dependency of the program /component/code on any single version of a given specification. With this in mind we propose InfWVM (Infosys WSDL version manager), a tool that will relieve the burden of generating the client for services consumption based on WSDL. InfWVM generates a standalone WSDL version-agnostic client based on *jax-rpc* standards instead of tightly coupled stubs normally generated using available proxy generation tools. InfWVM takes the WSDL of the service (whatever specification version), introspects the WSDL and generates a standalone client. This standalone client can be easily modified to include code for customizing serialization or de-serialization activities, etc.

InfWVM can generate the de-serialization and the serialization components provided the necessary inputs like the schema of the input and the output messages, etc. are provided along with the WSDL. InfWVM also provides a mechanism for converting existing WSDL 1.1-based WSDL to WSDL based on the WSDL 2 specification and re-hosting it on the corresponding SOAP engines. InfWVM is available as a plug-in component that can be incorporated into any of the existing SOAP engines. The basic component design of InfWVM is shown in Figure 2. Its key components are detailed there.

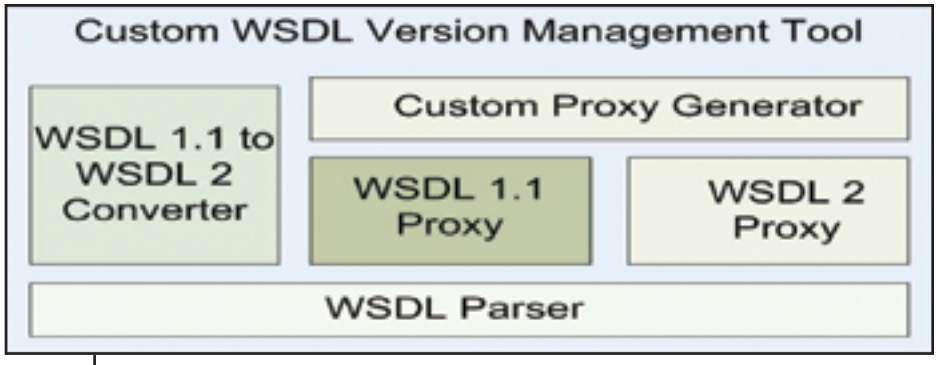


FIGURE 2 | InfWVM WSDL version management tool

Proxy Generator Component

This component is different from existing proxy generators in that it handles both WSDL 1.1 and WSDL 2 descriptions. It's designed as a façade, based on a set of logical version agnostic interfaces, over existing specific proxy generator tools. Hence the InfWVM proxy generator can accommodate future versions of WSDL too, by plugging the corresponding futuristic WSDL x.x version proxy generator into InfWVM and implementing the Custom Proxy Generator set of interfaces. When the client invokes the proxy generator with the corresponding WSDL, the generator goes through the following steps:

1. Identifies the WSDL version .
2. Loads the corresponding parsers.
3. Loads the corresponding WSDL Proxy Generators.
4. Generates the corresponding Proxies.

The component configuration is XML-based and editable to include variations in parser details and version-specific proxy generator details.

Version Converter Component

The version converter component in InfWVM converts WSDL 1.1-based WSDLs to WSDL 2-based WSDLs. Looking closely at the WSDL 1.1 grammar and specs, we see that they can be divided into parts to map into the component model of WSDL 2.0. WSDL 1.1 contains six major elements, namely types, message, portType, binding, port, and service. Broadly speaking, the types, portType, binding, port and service of WSDL 1.1 can be mapped to the types, interface, binding, endpoint and service of WSDL 2.0, respectively. This mapping looks fine based on the similarity of the description of these elements for any service. This forms the basis for our conversion tool.

The mapping logic is explained below:

- ***Definitions vs Description:*** The root of any WSDL 1.1 description is the *definitions* tag, while that of WSDL 2.0 is the *description* tag. Though *name* is optional in WSDL 1.1 specs and omitted in WSDL 2.0, *targetnameSpace* is required in WSDL 2.0 unlike WSDL 1.1 where it's optional. Hence, during conversion, a unique URI for *targetnamespace* is generated, even though it wouldn't fill in any more tags in that space.
- ***Imports:*** All the WSDL 1.1 documents that are imported also need to be converted to WSDL 2.0.
- ***Documentation:*** Documentation has to be put in the proper places.
- ***Types:*** The *types* of both WSDL 1.1 and WSDL 2.0 are compatible. Hence, types of a WSDL 1.1 document can simply be put into the translated WSDL 2.0 document.
- ***Messages:*** WSDL 2.0 doesn't differentiate between type elements and messages. Actually, type elements are renamed as message types. Hence, some care has to be taken here to omit messages from WSDL 1.1 description. WSDL 2.0 allows direct references to all elements defined under types except faults. This is to enable use of similar faults for similar operations and to create uniformity. Hence, messages carrying fault payloads are defined under the *interface* element of WSDL 2.0. Moreover, the references to messages in operations of *portType* in WSDL 1.1 are replaced by the payloads of these messages directly into the operations using the *element* attribute for the WSDL 2.0 *interface*. In case there are multiple message parts, a new message type/element can be created under *types,* containing the payloads in a complex type. Hence, a single element payload is put in the *operations* of the translated interface.
- ***Interfaces/PortTypes:*** WSDL 2.0 has molded the concept of portTypes into *interfaces*. Interfaces are more robust than portTypes on account of their more rigid and a well-defined structure. There are four transmission primitives defined for WSDL 1.1 and they map to the corresponding patterns in WSDL 2.0 as shown:

One way –> In-Only

Request response –> In-Out , *fault* is associated with *outfault*, as it replaces out message

Solicit response –> Out-In, *fault* is associated with *infault*, as it replaces in message

Notification –> Out-only

However, incases in which multiple fault messages are generated it becomes complex. A decision has to be provided for faults for the messages they replace (in or out).

- ***Bindings:*** Bindings are quite simplified in WSDL 2.0. *Operations* in *interfaces* are simply mapped to the MEPs of SOAP messages.
- ***Ports/Services :*** They are analogous to the *endpoints* and *service* in WSDL 2.0

Listings 4 and 5 show a sample WSDL based on 1.1 specifications that's been converted to a WSDL 2 specifications-based description by InfWVM.

Conclusions

The WSDL 2.0 standard takes care of the inherent gaps in representational issues in WSDL 1.1. It provides for capturing the required in-depth details of XML messages, simple exchanges between a service and a node, the transport in use, and the service locations. Its extensibility model provides the ability to include non-functional requirements related to a service, such as the security mechanisms or the privacy requirements for the service.

However, due to the current installed base of WSDL 1.1, there are bound to be interoperability issues associated with incorporating WSDL 2.0.

We have dissected the broad differences between both specs. To mitigate the interoperability issues, we propose InfWVM, a tool to help mitigate version management issues with WSDL. This primarly consists of version-specific client stubs generation tools, a logical version-agnostic set of interfaces, and a practical version converter component to help leverage existing WSDL 1.1 descriptions alongside newer WSDL 2.0 descriptions.

The version converter is based on a mapping analysis of WSDL 1.1 and WSDL 2.0 specs. The next step in InfWVM is to extend it and make it capable of defining the constraints and capabilities of Web Services, thereby leveraging the *features* of WSDL 2.0 beyond the scope of simply defining the definitions of XML messages.

References

- Web Services Description Language (WSDL) Version 2.0 Part 0: Primer. World Wide Web Consortium. http://www.w3.org/TR/wsdl-WSDL20-primer (accessed January 30, 2006). Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language. World Wide Web Consortium. http://www.w3.org/TR/wsdlWSDL20 (accessed January 30, 2006).
- Web Services Description Language (WSDL) Version 2.0 Part 2: Adjuncts. World Wide Web

Consortium. http://www.w3.org/TR/wsdlWSDL20-adjuncts (accessed January 30, 2006).
- Web Services Description Language (WSDL) 1.1. World Wide Web Consortium. http://www.w3.org/TR/wsdlWSDL. (accessed February 2, 2006).
- *Arulazi Dhesiaseelan.* What's New in WSDL 2.0. http://webservices.xml.com/pub/a/ws/2004/05/19/wsdlWSDL2.html. (accessed February 6, 2006).

About the Authors

Dr. Srinivas Padmanabhuni leads the Web services/SOA centre of excellence, SETlabs, Infosys, and specialized in Web services, SOA, semantic web and Enterprise Architecture. He has extensive publishing record in diverse conferences, journals, and international forums. He has also filed several patents, and spoken at various forums on the topic of Web services.

Anshuk Pal Chaudhuri is a member of the Web services centre of excellence, SETLabs, Infosys, and specializes in J2EE web services and Web services security. He has published extensively in different online journals and conferences, and filed patents in area of Web services.

Senthil Kumar K M was till recently a member of the Web services centre of excellence, SETLabs, Infosys, and specializes in Web services interoperability and Web services security. He has published extensively in different forums and conferences, and filed patents in area of Web Services.

Shaurabh Bharti is a junior research associate with the Web services centre of excellence, SETLabs, Infosys, and specializes in event driven architectures, and Semantic Web services. He has published extensively in different journals and conferences and has filed patents in area of Web services.

■ ■ ■ senthil_km@infosys.com
■ ■ ■ shaurabh_bharti@infosys.com
■ ■ ■ anshuk_palchaudhuri@infosys.com
■ ■ ■ srinivas_p@infosys.com

Listing 1: WSDL 2.0 skeleton

```
<description targetNamespace="xs:URI">
<documentation />
<types />
[<interface /> | <binding /> | <service />]
</description>
```

Listing 2: XML representation of a feature component

```
<feature ref ="xs: URI" required = "xs: boolean"?>
 <documentation/>
</feature>
```

Listing 3: Multiple parts translated as a single complex type

```
<complexType name="Composite">
<choice>
<element name="PO" minOccurs="1" maxOccurs="1" type="tns:
POType"/>
<element name="Invoice" minOccurs="0"
maxOccurs="unbounded" type="tns:InvoiceType"/>
</choice>
</complexType>
```

Listings 4 and 5:

```
<?xml version="1.0"?>
<definitions name="StockQuote"
targetNamespace="http://example.com/stockquote.wsdlWSDL"
          xmlns:tns="http://example.com/stockquote.wsdl-
WSDL"
          xmlns:xsd1="http://example.com/stockquote.xsd"
          xmlns:soap="http://schemas.xmlsoap.org/wsdl-
WSDL/soap/"
          xmlns="http://schemas.xmlsoap.org/wsdlWSDL/">

<types> <schema targetNamespace="http://example.
com/stockquote.xsd" xmlns="http://www.w3.org/2000/10/
XMLSchema"><element name="TradePriceRequest"><complexType>
<all><element name="tickerSymbol" type="string"/></all></
complexType>
</element><element name="TradePrice"> <complexType> <all>
<element name="price" type="float"/></all></complexType>
</element></schema></types>

 <message name="GetLastTradePriceInput"><part name="body"
element="xsd1:TradePriceRequest"/>message>
<message name="GetLastTradePriceOutput"><part name="body"
element="xsd1:TradePrice"/>message>

<portType name="StockQuotePortType"><operation name="GetL
astTradePrice">input message="tns:GetLastTradePriceInput"/
><output message="tns:GetLastTradePriceOutput"/></opera-
tion> </portType>

<binding name="StockQuoteSoapBinding" type="tns:
StockQuotePortType"><soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
<operation name="GetLastTradePrice"><soap:operation
soapAction="http://example.com/GetLastTradePrice"/>
      <input><soap:body use="literal"/></input><output>
<soap:body use="literal"/></output></operation></binding>

<service name="StockQuoteService"> <port
name="StockQuotePort" binding="tns:StockQuoteSoapBinding">
<soap:address location="http://example.com/stockquote"/>
</port> </service>

</definitions>

<?xml version="1.0" encoding="utf-8" ?>
<description
 xmlns="http://www.w3.org/2005/08/wsdlWSDL»
 targetNamespace=»http://example.com/stockquote.wsdlWSDL»
 xmlns:tns=»http://example.com/stockquote.wsdlWSDL»
 xmlns:soap=»http://www.w3.org/2003/05/soap-envelope»
 xmlns:xsd1=»http://example.com/stockquote.xsd»
 xmlns:wsoap=»http://www.w3.org/2006/01/wsdlWSDL/soap»
 xmlns:wsdlWSDLx=»http://www.w3.org/2006/01/wsdlWSDL-
extensions»>

<types><schema targetNamespace=»http://example.com/
stockquote.xsd» xmlns=»http://www.w3.org/2000/10/
XMLSchema»><element name=»TradePriceRequest»><complexType>
<all><element name=»tickerSymbol» type=»string» /></all></
complexType></element> <element name=»TradePrice»><comple
xType><all><element name=»price» type=»float» /></all></
complexType></element> </schema></types>

<interface name=»StockQuotePortType»><operation
 name=»GetLastTradePrice» pattern=»http://www.
w3.org/2006/01/wsdlWSDL/in-out" style="REQUEST_
RESPONSE"><input messageLabel="In" element="xsd1:
TradePriceRequest" /><output messageLabel="Out"
element="xsd1:TradePrice" /></operation></interface>

<binding name="StockQuoteSoapBinding" interface="tns:
StockQuotePortType" type="http://www.w3.org/2006/01/wsdl-
WSDL/soap" wsoap:protocol="http://www.w3.org/2003/05/soap/
bindings/HTTP"> <operation ref="tns:GetLastTradePrice"
wsoap:mep= "http://www.w3.org/2003/05/soap/mep/soap-
response" /></binding>

<service name="StockQuoteService" interface="tns:
StockQuotePortType"> <endpoint name="StockQuotePort"
binding="tns:StockQuoteSoapBinding" address="http://exam-
ple.com/stockquote" /></service>

</description>
```

WEB 2.0

Its Component Model and Message Exchange Patterns

The W3C released WSDL 2.0 as a Candidate Recommendation on January 6, 2006. The Web Services Description Group, part of the Web Services Activity, made three main documents publicly available for review:

Part 0: Primer – Intended to be a less-technical introduction to the main concepts described in the Core Language.

Part 1: Core Language – Describes the elements for the abstract concepts and the constructs for binding concrete implementations found in the Adjuncts document.

Part 2: Adjuncts – Defines the predefined extension points and mechanisms for pairing WSDL with its most likely partners SOAP and HTTP.

WRITTEN BY
CHRIS MADRID

The Candidate Recommendation, as a draft document, is intended for platform and tool vendors responsible for implementing the proposed standard. Implementations built using this version of the specification will help identify issues and potential gaps before the final specification is released. This draft release of the specification is also worthwhile for architects and developers to help them understand the promised capabilities assuming that the platforms and tools deliver.

The Highlights

The four years it took to bring the WSDL 2.0 specification to this point is indicative of how extensive it is. The main features that are introduced are:

- A Component Model that makes a distinction between abstract and concrete portions of a WSDL document.
- XML Schema is natively supported for Message Types.
- WSDL documents can be included and imported.
- XML Types can be included and imported.
- Interfaces serve as containers for faults and operations.
- Interfaces can inherit from each other.
- Message Exchange Patterns.
- Safe Operations.
- Bindings for SOAP 1.2 and HTTP 1.1.

The Component Model

By far the most interesting piece of WSDL 2.0 is the Component Model. It holds the promise of making services easier to describe and allowing those descriptions to be more readily reused. Unfortunately, what exactly defines a "Component Model" is not that easily understood. The specification states that it's a "set of components with attached properties, which collectively describe a Web Service."

Yet there's no evidence provided demonstrating why WSDL 1.1 couldn't claim to have a component model. The difference is that the structure of the elements found in WSDL 2.0 directly maps to modern programming language constructs like interfaces, classes, properties, and methods. For tool developers, this property of WSDL 2.0 makes it significantly easier to generate a WSDL 2.0 document from a service implementation and to generate a service implementation from a WSDL 2.0 document. For service developers, it becomes much easier to look at their WSDL definition and understand how it maps to their code.

The Component Model consists of the following:

- Description
- Interface
- Service
- Endpoint
- Binding
- Feature
- Property

Description

This component serves as the root container for other components, most notably interfaces, bindings, and services. Abstract and concrete components are distinctly identified and separated. The service interfaces compose the abstract elements of the document with their respective messages and operations. The binding, service, and endpoints compose the concrete elements of the document.

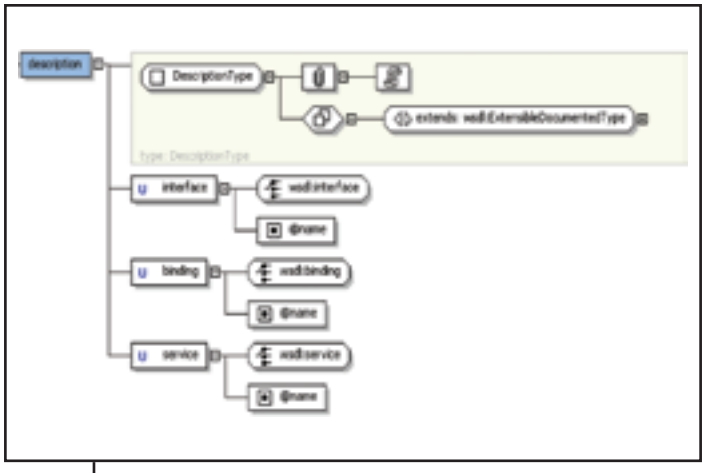


FIGURE 1 | Description

Don't Miss the 2006 JavaOne℠ Conference

More than 300 technical sessions and Birds-of-a-Feather sessions will be offered at the 11th Annual JavaOne℠ Conference. See and hear from the Industry leaders and technology experts over four content-rich days. Included are:*

PLATINUM COSPONSORS

GOLD COSPONSORS

SILVER COSPONSORS

Introduction to AJAX

Ben Galbraith, *Consultant* | Dion Almaer, *Adigio, Inc*

This session provides an introduction to AJAX and an orientation to the state of the AJAXian universe. It demonstrates the basic AJAXian techniques through live coding and demonstrates and deconstructs more-advanced examples of AJAX.

Effective Java™ Reloaded

Joshua Bloch, *Google, Inc.*

It has been five years since *Effective Java*™ was released. The Java platform has evolved, and we've learned more about how to use it to best effect. Therefore, a second edition of *Effective Java*™ is being released to coincide with the 2006 JavaOne conference. This presentation covers new material that has been added to the second edition, material that should be useful to every working Java technology programmer.

Spring Framework Update

Rod Johnson, *Interface21*

Rod Johnson, the father of Spring, brings attendees up to date on some of the many significant new features in the Spring 1.3 and 1.4 releases. He discusses what's new and cool in the Spring world and examines the implications of these new features for best practice in developing applications with the Spring Framework. Johnson shows code examples throughout the presentation, leaving attendees ready to try these features out for themselves.

To see more information on the conference offerings visit java.sun.com/javaone/sf

*Content subject to change.

Register by April 14, 2006 and SAVE $100.

java.sun.com/javaone/sf

JavaOne™ Conference | May 16-19, 2006

JavaOne℠ Pavilion: May 16–18, 2006, Moscone Center, San Francisco, CA

JavaOne
Sun's 2006 Worldwide Java Developer Conference

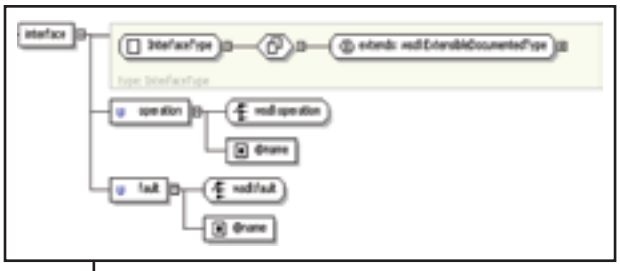


FIGURE 2 | Interface

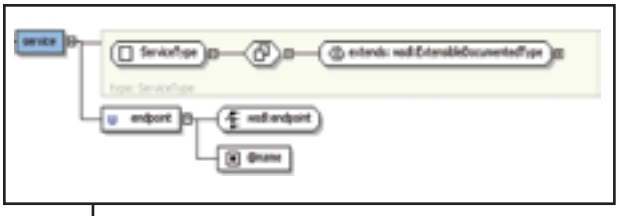


FIGURE 3 | Service

Interface

This component serves as a container for operations, which in turn serves as a container for messages. The importance of this feature arises from the inclusion of Message Exchange Patterns (MEPs), since an operation can consist of more than simple request and response messages.

A discussion of the predefined Message Exchange Patterns and Fault Propagation Rules is included below. One of the most discussed features of WSDL 2.0 in the industry is the ability for interfaces to extend one another. This inclusion as a feature, along with the ability to include and import WSDL, certainly makes the job of tool developers more difficult. Yet interface inheritance should increase the reusability of interfaces across an organization.

Service

This component represents the collection of endpoints where the service may be invoked.

Initial thoughts may envision that the Service component contains the bulk of the specification. However, it mainly serves to tie the service interface to one or many endpoints implementing that interface.

Endpoint

This component represents a collection of information for a given service implementation. The endpoint ties a specific binding to the specific address so the service can be invoked.

Binding

This component represents the detailed information required to access an endpoint. The binding provides the glue the makes the whole system work and provides mechanisms for defining concrete formats for faults and messages and protocol interactions for interface operations.

Feature

This component represents functionality implemented by the services that may or may not be required by service requestors. Features are scoped to a given component when that component directly declares the feature, or when a contained component declares the feature, or when a referenced component declares the feature. If the feature is declared in multiple components that bring it in scope, then the feature must be active if any of the components specifies that it's required. An example of a feature may be declaring support for a duplex channel.

Property

This component represents an out-of-band parameter affecting the service's behavior, visible or otherwise. In combination with a secure message feature, a property contained by the feature can indicate the encryption algorithm.

Message Exchange Patterns

Operations in WSDL 2.0 specify one of eight defined message exchange patterns. As part of the abstract Interface component, the binding addresses the specifics regarding the synchronous or asynchronous exchange of messages and whether those messages are exchanged over a simplex or duplex channel. Each message pattern follows one of three fault propagation rules specified as part of the Adjuncts. Before looking at the message exchange patterns, it's necessary to become familiar with the fault propagations rules that they leverage. The three Fault Propagation Rules are:

- Fault Replaces Message
- Message Triggers Fault
- No Faults

These rules are intended to specify the recipient of the fault message, which may or may not be the service requestor. Overlap with WS-Addressing is apparent and the WSDL 2.0 specification states that WS-Addressing will be "used in lieu of the recipient nominated by the ruleset."

Fault Replaces Message

This behavior is accurately captured by the name. For any node that generates a fault after receiving a message, the next message in the message exchange pattern is simply replaced with the fault.

In more complex scenarios, the fault may not be sent to the original message sender. However, with the included Message Exchange Patterns that specify a Fault Propagation Rule other than "No Faults" the end result is that faults are returned to the original sender.

At first glance this behavior is counter-intuitive, especially for developers of the message-sending node. Yet further thought reveals that in a system of related services, it may be desirable for a concrete implementation of the sending node to delivers its message asynchronously. If

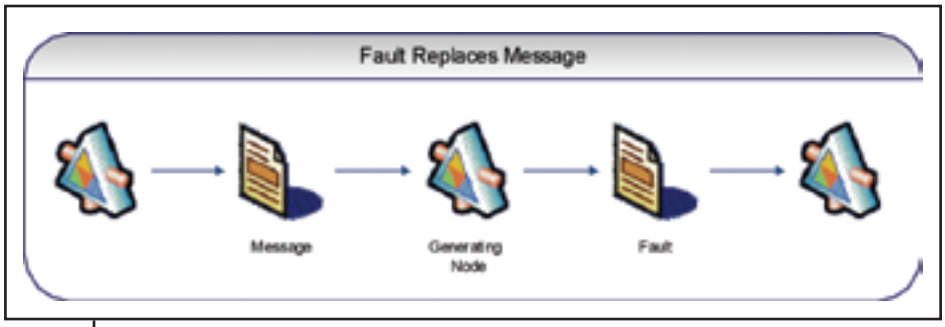


FIGURE 4 | Fault replaces message

JAVA J2EE-Ready Managed Dedicated Hosting Plans:

Xeon III

SAMEDAY SETUP

Dual 3.2 GHz Xeons
4GB RAM
Dual 73GB SCSI
1U Server
Firewall
Linux
Monitoring
NGASI Manager

$399 *monthly*

Pentium 4 I

SAMEDAY SETUP
FREE SETUP

2.4 GHz P4
2GB RAM
Dual 80GB ATA
1U Server
Firewall
Linux
Monitoring
NGASI Manager

$199 *monthly*
2nd month FREE

4Balance I

1 Database Server and 2 Application Servers connected to 1 dedicated load balancing device.
Dual Xeons.
High-Availability.

$1724 *monthly*

NEW! Geronimo 1.0 Hosting . Virtual Private Servers(VPS) starting at $69/month

At **WebAppCabaret** we specialize in **JAVA J2EE Hosting**, featuring **managed dedicated servers** preloaded with most open source JAVA technologies.
PRELOADED WITH:
JDK1.4 . JDK1.5 . Tomcat . Geronimo . JBoss . Struts . ANT . Spring . Hibernate
Apache . MySQL . PostgreSQL . Portals . CRM . CMS . Blogs . Frameworks
All easily manage via a web based control panel.
Details:
-All Servers installed with the latest Enterprise Linux
-Firewall Protection
-Up to 60 GB daily on site backup included at no extra charge per server.
-Database on site backup every 2 hours
-Daily off site database backup
-A spare server is always available in case one of the server goes down
-Intrusion detection.
-24x7 Server and application monitoring with automatic self healing
-The Latest Bug fixes and Security updates.
-Tier 1 Data Center. 100% Network Uptime Guarantee
-Guaranteed Reliability backed by industry-leading Service Level Agreements

Log on now at **http://www.webappcabaret.com/jdj.jsp** or call today at **1.866.256.7973**

NGASI POWERED

WebAppCabaret sm

JAVA J2EE Hosting

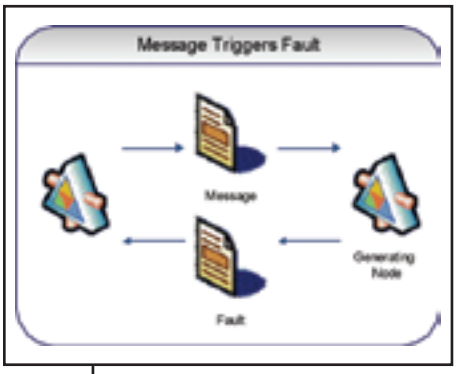


FIGURE 5 Message triggers fault

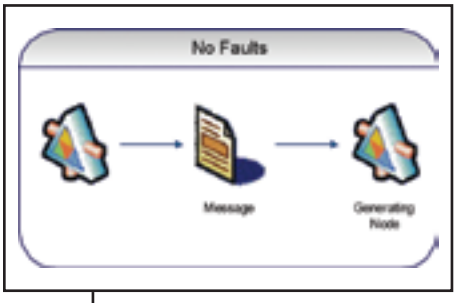


FIGURE 6 No faults

a fault is generated, it may be another service that's responsible for taking corrective actions.

Message Triggers Fault

This behavior means that for any node that generates a fault after receiving a message, the fault is sent to the message sender.

This ruleset differs from Fault Replaces Message in several ways. First, the fault is always sent to the sender of the message that generated the fault. This may or may not be the case in Fault Replaces Message. Second, in Fault Replaces Message a node can receive a Fault instead of an expected Message; while in a service using Message Triggers Fault, a node can receive a Fault even though it wasn't expecting a message in return.

No Faults

This behavior means that for any node that generates a fault after receiving a message, the fault is discarded.

With a basic understanding of the Fault Propagation Rules, the Message Exchange Patterns in WSDL 2.0 can now be analyzed. Each of the following Message Exchange Patterns references one of the previously defined Fault Propagation Rules:

- Inbound Patterns
  - In-Only
  - Robust In-Only
  - In-Out
  - In-Optional-Out
- Outbound Patterns
  - Out-Only
  - Robust Out-Only
  - Out-In
  - Out-Optional-In

In-Only

This pattern is intended for fire-and-forget services and is analogous to operations in WSDL 1.1 that failed to define an output message. By using the "No Faults" fault propagation ruleset, faults are discarded.

Robust In-Only

Even though the specification leaves the decision of synchronicity to the concrete bindings and service implementation, this pattern is intended for services that desire asynchronous invocation, but needs notification that a fault was generated. Because this pattern follows the "Message Triggers Fault" ruleset, the node generating the fault is responsible for attempting to send the fault message to the sender. Potential concrete implementations may involve a duplex channel to accomplish the fault notification.

In-Out

This pattern is the traditional request-response pattern prevalent in many services currently in production. Its use of the "Fault Replaces Message" ruleset indicates that if the "In" message generates a fault, then the "Out" message would be replaced by the Fault message. This behavior is consistent with the existing generation of services described by WSDL 1.1. These patterns provide a way of describing those same services in WSDL 2.0.

In-Optional-Out

This pattern is a variation of the In-Out pattern where the Out message is optional. It uses the "Message Triggers Fault" ruleset, so faults would be returned to the message sender; however, the Out message may or may not be sent to the sender. Potential concrete scenarios include, but aren't limited to, services bound to a duplex channel so that the sender isn't required to block guessing whether or not a message will be returned.

Out-Only

The outbound patterns aren't obviously applicable and most certainly generate a number of questions. However, large enterprises that face the challenge of data subscribers and dynamic composability of services might find significant value here. This specific pattern can be implemented concretely as a fire-and-forget service. A potential scenario can include wiring up a packaged application to an enterprise logging solution.

Robust Out-Only

As another outbound pattern, this pattern is much like Robust In-Only. It's intended for scenarios where the main interaction will be implemented asynchronously, but the generator of the first message in the pattern has to be notified of the fault.

Out-In

This pattern is the outbound representation of the traditional request-response pattern. In essence, the contract is saying that "clients" have to provide an endpoint for the "service" to invoke. Like its inbound counterpart, it returns the fault to the message sender.

Out-Optional-In

This pattern uses the "Message Triggers Fault" ruleset so that faults will be returned to the message sender, but the response, indicated as the "In" message, may or may not be sent. Integration scenarios are likely to be the primary users of this pattern if implementations follow an asynchronous messaging approach.

Conclusion

The Component Model and Message Exchange Patterns found in WSDL 2.0 are promising additions to the language. Even though implementations of the WSDL 2.0 specification aren't readily available, service development and service deployment are expected to become much easier with the next generation of platforms and tools built with the specification in mind.

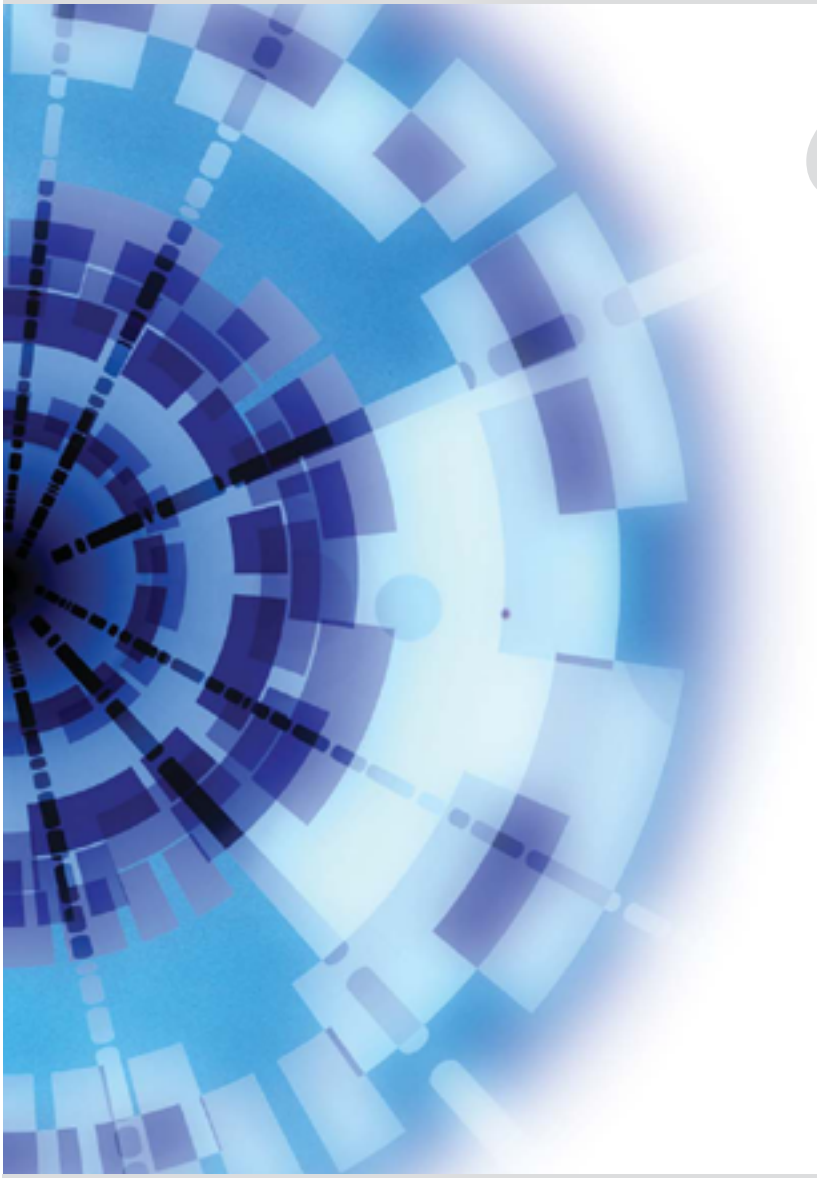
About the Author

Chris Madrid is a senior solution architect at Avanade focusing on strategic enterprise SOA initiatives and the technologies, processes, and tools to make it a reality.

christopherma@avanade.com

#1 Rated AJAX and Rich Internet Application toolkit* TIBCO General Interface

Build 100% Pure Browser Rich Internet Apps with TIBCO General Interface™



"TIBCO General Interface is a friendly, capable toolkit for building sophisticated JavaScript Web applications that run in a browser."

InfoWorld 2006 TECHNOLOGY OF THE YEAR AWARD

Download today
at **http://www.tibco.com/mk/gi**

TIBCO®
The Power of Now®

ActiveBPEL 2.0 from Active Endpoints Excels at BPEL

Business process execution Language support or BPEL is at the top of every enterprise SOA punch list. It's an XML-based language designed to support long-running complex business transactions in the form of orchestrated Web Service interactions. Like most XML formats, you wouldn't want to construct and debug a process of any complexity by hand and an "engine" is required to recognize and execute BPEL.

This is where the tool vendors come in and Active Endpoints, Inc. has a design tool and engine product combination that we'll cover in this review.

WRITTEN BY
PAUL T. MAURER

ActiveBPEL Designer

The designer is a world-class visual environment for working with BPEL-based processes. ActiveBPEL Designer is built on the seemingly ubiquitous Eclipse extensible development platform and, as you can see in Figure 1, has an interface with a clear and logical layout.

The "Navigator" tab in the upper left region displays a hierarchical view of projects, folders, and files in the workspace. To the right of the Navigator is the "Web References" tab. This tab contains a registry of namespaces, messages, type definitions, and sample data, used in BPEL processes. It's populated automatically as WSDL files and XML schemas are added to the workspace. The "Web References" tab has many features for slicing and dicing the view, but my favorite is its ability to drag Web references and drop then in the process editor canvas.

Just below the process editor canvas in the center is the Outline tab. This view displays all the major components of the current BPEL process. To the right is the Properties view that displays names and values of properties of the currently selected resource. The process editor, outline and properties views are all selection-synchronized. That is, select a component in the outline view and the component is also selected in the process editor and its properties are displayed.

Another nice feature is the "Problems" tab at the bottom of the display. The ActiveBPEL Designer generates a list for all the incomplete or invalid BPEL constructs in a process. The messages are informative and useful when completing a process.

Tutorial

To exercise the designer, I walked through the tutorial included in the 446-page ActiveBPEL Designer User's Guide provided by Active Endpoints. The tutorial was thorough, covering not only the basics, but providing steps for fault handling, compensation, simulation, deployment, and debugging. The designer was very responsive and the user interaction required to build a process was well thought out.

One of the nice features in Designer is the ability to store groups of activities to the palette for later reuse. This provides a simple tool to shortcut repetitive work.

Testing

The features for testing and simulating a process execution are simple but very useful. The ActiveBPEL Designer lets the user enter sample data in the Web References view for all messages defined in the process. Files containing test data can also be used. During a simulation run, this test data is fired at appropriate points in the process.

The designer tool comes with the ActiveBPEL engine embedded for simulation and debugging. During a simulation, process execution can be suspended or resumed and process variables can be inspected and modified. Designer allows a user to alter sample data values, expressions, correlation property values, and the partner link addresses of an executing BPEL process.

Remote debugging of a process running on the server is fully supported. In remote debugging, one of the options available is to automatically suspend the process when it's

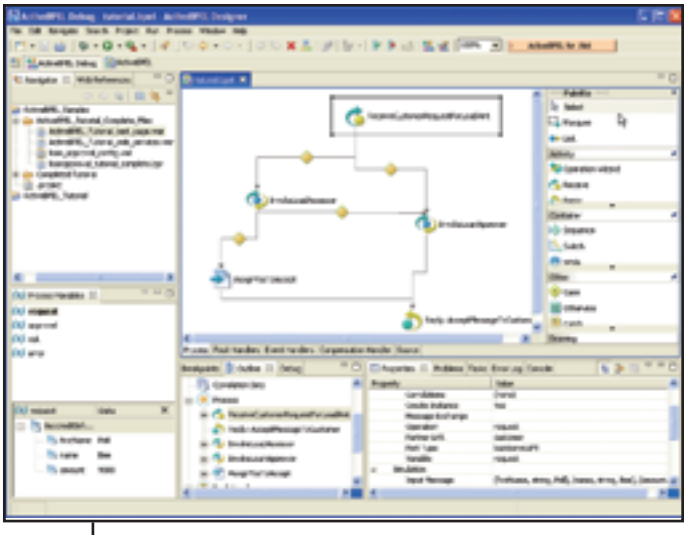


FIGURE 1 | ActiveBPEL Designer

first invoked from the Designer. This lets the user easily step through a process executing on the server. There are far too many knobs and levers in the remote debugging feature to go into in this review. Suffice it to say that it seems the team at Active Endpoints has probably experienced and provided support for most of the remote debugging scenarios you're likely to encounter.

Custom Functions

For those that like to tinker under the hood, Active Endpoints provides two support mechanisms for custom functions in a BPEL process. First, the Jaxen (java XPath engine) FunctionContext interface is supported. This interface provides an API for XPath expression evaluation.

The preferred interface is the ActiveBPEL server Java framework. This framework supports custom function extensions as a set of Java Interfaces and classes. A developer basically creates a custom function context and custom function using the interfaces from the framework. Then he packages the code as a jar file and deploys it to the ActiveBPEL designer for simulation and the ActiveBPEL server for execution.

A nice feature is that custom functions can be made available in the expression and query builders and are supported under simulation just like any other built-in function.

> "In this era of Open Source and its hybrid business models, Active Endpoints has crafted an interesting licensing model that is developer-friendly"

Licensing

In this era of Open Source and its hybrid business models, Active Endpoints has crafted an interesting licensing model that is developer-friendly.

First the ActiveBPEL engine core is an Open Source product and has been since July 2004. The engine isn't a feature-crippled version but a fully functional BPEL engine that can run all the BPEL processes created by the ActiveBPEL Designer tool. Second, although not an Open Source product, the designer tool is freely downloadable from the Active Endpoint's Web site. So, for the cost of a little bandwidth and your patience during a download, you can have a fully functional BPEL-enabled development and runtime environment. Pretty slick, huh?

"But how do they pay the bills?" you may ask. Well, aside from the standard services and support fees, Active Endpoints has created an enterprise version of its BPEL engine with a few key high-end features. Process versioning is one key feature but most importantly it's added all the hooks, bells and whistles to ensure that the engine is integrated tightly with today's top application server platforms. High availability, clustering and seamless platform systems management integration are the money features. A process can also be set to suspend on an uncaught fault and breakpoints can be set at appropriate points in the process. For large enterprises that run mission-critical processes, the enterprise version of ActiveBPEL engine is a must. The cost of the enterprise product varies based on the deployment platform and ranges from a low-end Tomcat-enabled version to a J2EE-enabled version that can run on a mainframe.

Support

Active Endpoints offers a few support options. First, for Open Source users, forum and mailing list-based support is provided. I registered and logged onto the free forum to check it out. There seemed to be plenty of activity and strong participation from the user community as well as Active Endpoints people.

There's also a premium support forum for users with a valid product serial number. For those enterprises with higher support demands, incident-based support can be purchased.

Active Endpoints has an excellent Web-enabled bug tracking system that lets customers search and filter the entire bug database. The layout is clear and easy-to-use and a good self-service support feature.

Conclusion

Active Endpoints has created an excellent BPEL design tool and execution engine that is freely downloadable, well documented and has good community support. There's virtually no cost of entry and enterprise reliability features can be purchased for mission-critical applications.

For any of our readers out there who are looking at BPEL solutions, Active Endpoints' products should definitely be at the top of their list.

About the Author

Paul Maurer is a principal in the financial services practice of a leading consulting services company.

paul@paulmaurer.net

Company Info
Active Endpoints, Inc.
Three Enterprise Drive
Shelton, CT 06484

Web: http://www.active-endpoints.com/
E-mail: info@active-endpoints.com

Licensing Information
ActiveBPEL Designer: Free
ActiveBPEL Engine (Open Source): Free
ActiveBPEL Enterprise:$7.5k-$30k per CPU based on the deployment platform

Testing Environment
OS: Windows XP Professional (Service Pack 2)
Hardware: Intel Pentium M Processor (1300MHz) – 1.29GHz with 1GB RAM

BEST PRACTICES FOR SECURING WEB SERVICES

Ensuring complete Web Service security

Security has the inherent nature of spanning many different layers of a Web Services system. Web Services vulnerabilities can be present in the operating system, the network, the database, the Web server, the application server, the XML parser, the Web Services implementation stack, the application code, the XML firewall, the Web Service monitoring or management appliance, or just about any other component in your Web Services system.

Therefore security testing, which is important for any software application, is even more crucial for Web Services. This article explores security issues specific to Web Services and illustrates the engineering and testing best practices required to ensure Web Service security throughout the Web Services development life cycle.

WRITTEN BY
ADAM KOLAWA

Step 1: Determine a Suitable Web Services Security Architecture

Web Services security architecture not only depends on the required security measures, it also depends on the service scope and scale of deployment. For instance, security can either be enforced in the application server itself, or as a separate security appliance (such as an XML firewall) that can virtualize the service by sitting in the middle between the service and its consumers. Most Web Service architects recommend decoupling the security layer from the application server to achieve better maintainability, flexibility, and scalability. However, using a security appliance as an intermediary may not be necessary for simple end-to-end Web Service deployments.

Another architectural decision to make is whether to implement the security on the transport layer or on the message layer. TLS (Transport Layer Security) is a mature technology so both standards and tools have already been developed. It also provides a good transition path for engineers who are somewhat familiar with transport-level security but are new to Web Services. On the other hand, TLS has inherent limitations that make it inappropriate for some situations. Fortunately, message layer security provides an alternative solution for situations where TLS's limitations are troublesome.

Transport Layer Security

The security of the transport that's being used for the Web Service can be used to protect the Web Service. For example, for HTTP one can enable Basic Authentication, Digest Authentication, or SSL. When JMS API is used to transmit SOAP messages, then JMS Authentication can be used.

The main benefit of using TLS is that it builds on top of existing Web application experience to implement the security. Many developers know SSL and it's easy to enable it in common Web and application servers. SSL is a particularly ideal choice for end-to-end Web Service integrations. SSL can enforce confidentiality, integrity, authentication, and authorization, thus protecting the Web Service from capture and replay attacks, WSDL access, and scanning.

The drawback of SSL is that it's an all-or-

nothing protocol. It doesn't have the granularity to secure certain parts of the message, nor can it use different certificates for different message parts. Besides, all intermediaries on the message path would have to have the proper certificates and keys to decrypt the entire message to process it then resend it over SSL again, which can be difficult or even impossible in some cases.

Message Layer Security

Currently, there's a lot of activity in the area of message level security and it's fair to say that it's not nearly as mature as TLS. With that disclaimer, here are the message layer security technologies that may become important because they address some of the same concerns as TLS (privacy, authentication, and message integrity) at the message level instead of the transport level:

- The XML Signature provides a mechanism for digitally signing XML documents or portions of XML documents. The signature doesn't have to be in the document that's being signed, so you can also use XML Signature to sign non-XML documents.
- XML Encryption provides a mechanism for encrypting portions of the XML documents. Encrypting a complete document is pretty easy; just treat it like a text document. There are some subtleties involved in encrypting portions of a document, however, these are what XML Encryption addresses.
- WS-Security is perhaps most easily understood as a specification that defines a standard way of securing a single message by applying Username Tokens, XML Signatures, and XML Encryption to a SOAP envelope. The Username Token profile provides a simple way to describe authentication data, i.e., usernames and passwords.

When using one or more of the message-layer security standards, it's important to use the combination that provides the required security protections. For example, Username Tokens alone can't secure a message against capture and replay attacks unless they include a signed nonce and time stamp, and the SOAP Body is signed. The signature can be generated using the password in the token, or it can be independent of the password. However, if that password isn't digested as recommended by the specification, then the token itself should be encrypted. As another example, XML encryption doesn't ensure the message authenticity or integrity, in which cases XML signature can be combined with encryption to ensure all three requirements.

Step 2: Adhere to Technology Standards

As in other security fields, adherence to standards is a necessary practice for Web Services. There's a consensus among security professionals that publicly available, commonly used, well-analyzed cryptographic algorithms are the best choice, simply because they've already undergone a great deal of research and scrutiny since they were adopted by the industry. The same principle applies to Web Services security.

For example, compliance with the WS-Security specification from OASIS will likely be safer than developing your own custom security implementation because it's been developed by experts in the field with threat protection in mind. Furthermore, you can reduce development time by using a readily available implementation of the specification and your service would be able to interoperate with other implementations of the same standard.

Another issue to consider with regard to adherence to standards is compliance with the Basic Security Profile (BSP) from WS-I. The BSP is intended to address interoperability, but in some cases it restricts the W3C and OASIS specifications in a manner that favors stronger security practices. Moreover, section 13 or "Security Considerations" of the BSP lists a number of useful security considerations that should be taken into account when deploying secure Web Services using WS-Security.

Step 3: Establish an Effective Web Services Testing Process

Understanding security threats isn't enough. It's necessary to have a mature engineering process that makes security vulnerability detection and testing an indivisible part of the Web Services development process so the threats are mitigated to the maximum extent. Thinking about security as early as possible throughout the Web Service lifecycle is key to achieving the best results in the most efficient manner.

A common pitfall that companies encounter is their attempt to use the same human and technological resources of Web QA and testing for Web Services without implementing the proper training, processes, and technology changes that can leverage such resources successfully. The same resources used for Web QA and testing can't be used for Web Services for the following reasons:

- Web Services testing requires a different skill set in XML, SOAP, WSDLs, and other WS standards, let alone experience in security issues unique to Web Services
- Web Services testing can be better implemented with specialized tools rather than tools that are designed for traditional Web testing. The features of these tools need to support Web Services standards and have the ability to design tests along these standards.
- Web Services security testing requires the facilitation of tools and practices that can exercise the tests for exposing vulnerabilities that are general to Web applications and specific to Web Services alike.

To detect and prevent security vulnerabilities in Web Services, several engineering activities must be done on multiple fronts. These activities can be summarized into three tiers.

Tier-One Testing: Static Analysis

Knowledge of unsafe coding practices is crucial when developing secure software, but detecting such practices can be a tedious, time-consuming process unless it's automated as much as possible.

Static analysis tools are proving to be very effective in exposing dangerous method calls, insufficient validations, or poor code quality. Although manual code inspections can expose some of these problems, such problems can be subtle and difficult to find manually. Static analysis doesn't eliminate the need for code inspections completely, but it can significantly reduce the time and effort required to do them since static analysis tools can scan the entire source code to identify unsafe coding patterns then the code reviewer can analyze these instances to verify their severity. Without such automation, much more time would be spent in finding the unsafe coding patterns in the first place.

For example, in Java using "PreparedStatement" is recommended over plain "Statement" to prevent SQL injections. A static analysis rule that searches for Statement.executeQuery() invoked with a dynamic string can pinpoint an engineer to this statement and provide a first line of defense against SQL injection problems. Other common insecure code patterns that can be found with static analysis include XPath Injections, uncaught exceptions that cause improper error handling, and some denial of service conditions caused by resource intensive operations.

Suspicious code patterns can also be identified with static analysis. Some security bugs result from programming negligence. However, more dangerous code can come from malicious programmers who hide Trojans, easter eggs, or time bombs in their code to provide discreet access at a later time. Such code often relies on random numbers or date/time checking to avoid detection, and it can change the normal security settings to allow surreptitious access. Static analysis rules that find all random objects and time date objects, called "triggers," that find custom class loaders and security managers, can help a code reviewer identify and inspect suspicious code pattern.

Besides detecting vulnerable or suspicious code, it's important to keep in mind that coding best practices play a role in producing secure code. There are also several different types of coding standards that can be enforced through static analysis that have general, rather than specific security relevance, and can improve the overall security posture of an application. For example, if code is found that has a synchronization problem, such a problem impacts security because synchronization problems tend to have unexpected effects. Indeed, coding practices should be considered during security testing.

Tier-Two Testing: Penetration Testing

Not all security vulnerabilities can be found though static analysis, so penetration testing comes into the picture to expose such problems. Penetration testing dynamically exercises and scans the Web Service deployed on a staging or production server.

Understanding the security threats lets the tester design tests that can expose them with the help of good tools. For example, external entity attacks and XML bombs can be thrown at the service to see if the service refuses to process XML processing instructions or DTDs by returning a SOAP fault. WSDL access vulnerabilities can be detected by attempting to get a WSDL without the expected security channel if it's protected. For example, if the WSDL is protected with client-side SSL on port 443, it shouldn't be accessible on port 80; it's possible to forget an open connector in the Web server, which leaves multiple open channels. When it comes to thwarting WSDL scanning threats, then it's important to inspect the WSDL for redundant artifacts such as schemas or unused message definitions.

Capture and replay attacks can be simulated by sending multiple requests with the same message identifier that determines its uniqueness. For example, if you're using Username Tokens, you should test the service by sending multiple messages with the same nonce values and verify that the service rejects such requests properly. The service should implement a sufficient, but limited cache size for the recently accepted nonce values. Many WS-Security implementations don't take this into consideration by default, which makes them vulnerable to capture and replay attacks.

To test a Web Service's vulnerability to DoS attacks caused by heavy loads, such DoS attacks should be simulated in fashion that's suitable to Web Services. You can't tell if a service can sustain a certain load scenario unless such a scenario has been tested. However, it's important to execute such load tests in a manner that's effective.

Some test engineers have the tendency to do load tests with the same static request to generate a load. Although this is a viable test scenario, it's not sufficient because such DoS attacks can be detected by network security appliances. Therefore, Web Service DoS attack simulations should be generated with dynamic request values that are semantically valid and can exercise wider code coverage in the Web Service's application logic to test the Web Services to its limits. Such attacks are difficult to generate by manual coding, but they're possible with load testing tools that are specialized for Web Services. In fact, the mere existence of such tools should alert Web Service engineers that such attacks can be done easily by a hacker if such tools fell into their hands. For example, to test a Web Service that accepts Username Tokens with timestamps and nonces, it's important to apply a load on the service where the timestamps and nonces are dynamically generated for each request. Otherwise, errors such as the ones caused by concurrency problems would go undetected. Another example would be load tests that send signed requests, where the hash and signature values should differ from one message to another.

Not only should Web Service load tests generate dynamic requests, but such tests should also simulate real-use case scenarios or usage patterns. For example, a use case scenario could be a Web Service client retrieving an authorization token (such as a SAML assertion) from a security authority, then using that token for subsequent Web Service invocations on different services. To test that scenario, load tests that keep using the same authorization token over and over again don't represent the real-world scenario since a real-use scenario would have multiple users requesting and using multiple tokens at the same time. Executing such a realistic load test can expose concurrency or scalability problems that result in vulnerabilities. In this example, it's possible for the Web Service to reject valid requests or accept unauthorized ones under a certain load even if such problems don't occur during regular functional testing.

To detect invalid responses during a load test, the load tests should be backed with sufficient response validations that ensure the detection of regressions from the correct behavior, because it's difficult to verify that all requests were met with the correct responses unless regression detection was done while the load is being generated. Without response validation, only network connections and HTTP errors would be exposed, which doesn't provide sufficient test coverage. For example, responses can be well-formed SOAP messages but with invalid data, or perhaps they contain an error message when they shouldn't. Without placing sufficient response validation during a load test, such incorrect responses can go undetected.

Tier-Three Testing: Runtime Analysis

Runtime analysis of the state of Web application code is needed to detect certain security problems that can't be detected with the previous two tiers of testing. For example, in C/C++ applications that are exposed as Web Services, memory corruption (especially memory corruption on the stack) indicates a potential for buffer overflows that could cause serious security problems, and memory leaks make the application more vulnerable to denial of service attacks. Dynamic analysis can find security vulnerabilities that can result from the integration of otherwise secure components because it takes data flow analysis into consideration, whereas static analysis provides large code coverage with a narrower scope on data flows.

Combining the Three Tiers

Since each security-testing tier provides a methodology exposing vulnerabilities from a unique aspect, combining two or more of the three tiers could provide a powerful approach to security testing. For example, static analysis can be used to determine the scope of the required penetration testing by recommending a more selective set of possible vulnerabilities to penetrate.

Runtime analysis combined with penetration testing gives the tester visibility into the application as it performs under a variety of conditions. For example, one can do runtime analysis during load testing to find memory leaks.

> "Security can be enforced in the application server itself or as a separate security appliance that can virtualize the service"

Step 4: Create & Maintain Reusable, Re-runnable Tests

The above testing practices can become too expensive to do unless proper automation is applied to the testing process. Many organizations don't have the resources to do these tests if they were to be done manually and repeated for each project milestone.

Modern software development processes are iterative. Software engineering activities should be done on a recurring, iterative basis rather than following a rigid, one-directional development model that tests only at the end. Testing only at the end of the development cycle is one of the main reasons for late deliveries and exceeded project costs and Web Services are no exception to this fact.

However, such an iterative development model can only be effective if the engineering activities are backed with proper automation. Therefore, it's necessary to establish a Web Services testing environment that's driven by automation that can help create the tests, maintain them, manage them, and execute them on a regular basis; typically every night as part of the existing "nightly" build and test process for the product. The alternative would be to run the various Web Services tests manually, each one at a time, by modifying a client's request, which is a tedious, non-efficient process. It's therefore better to keep and maintain all the Web Service tests that are created so they can be re-run quickly, easily, and so you can run them all automatically as regression tests whenever a Web Service is updated.

After running security tests along the three tiers we described, one can find problems that require fixes that ripple through Web Service at a time when they're too risky or expensive to fix, which is why such tests are better executed early and regularly.

When a problem is discovered then the test that exposed the problem should ideally be added to the existing test pool and re-run on a recurring basis with all the other tests so it prevents that error from occurring again.

Conclusion

Securing your Web Services is a vital aspect of ensuring a successful deployment. When deployed externally for consumption by partners or customers, only secure Web Services can provide a justifiable integration solution, because the benefits they expose should far outweigh the risks. The key to effective Web Services security is to know and be aware of the various types of security threats, understand the technical solutions for mitigating these threats then establish and follow a defined engineering process that takes security into consideration from the beginning and throughout the Web Service lifecycle. By following the four steps outlined in this article, you can ensure complete Web Service security.

About the Author

Dr. Adam Kolawa is cofounder and CEO of Parasoft, a vendor of automated error-prevention software and services based in Monrovia, CA. Dr. Kolawa, who is the coauthor of Bullet-proofing Web Applications (Wiley, 2001), has written and contributed hundreds of commentary pieces and technical articles for publications such as The Wall Street Journal, CIO, Computerworld, Dr. Dobb's Journal, and IEEE Computer. He has also authored numerous scientific papers on physics and parallel processing. He holds a PhD in theoretical physics from the California Institute of Technology.

ak@parasoft.com

The #1 i-Technology Educational and

Register at... www.SOAEOSConference.sys-con.com

The **10th International SOA Web Services Edge Conference**, colocated with the First Annual **Enterprise Open Source Conference**, will take place on **June 5-6, 2006** at the historical **Roosevelt Hotel in New York City**. SOA Web Services Edge and Enterprise Open Source Conference together deliver the #1 i-technology educational and networking opportunity of the year.

This year, more than 100 distinguished conference faculty members will present **96 cutting-edge SOA, Web Services and Enterprise Open Source topics** in educational classes, panels, and keynotes in six simultaneous tracks for two information-packed days, plus a bonus track, a **"Real-World AJAX"** two-day seminar, a $1,295 value!

The conference program does not merely present a comprehensive view of all the development and management aspects of integrating a **SOA strategy** and an **Open Source philosophy** into your enterprise, its organizing principle is that delegates will go away from the intense two-day program replete with why-to and how-to knowledge delivered first-hand by industry experts.

According to analyst firm Gartner Group, **by 2008 more than 60 percent of enterprises will use SOA** as the guiding principle when creating mission-critical applications and processes. **"Businesses that ignore the potential of SOA will find themselves outpaced by rivals who improve their agility and transform themselves into new kinds of enterprises,"** says Gartner analyst Yafim Natis.

Meanwhile, the **Open Source revolution** has spread into traditional areas of enterprise IT, with all major vendors and a global group of **Open Source entrepreneurs** developing the technology and delivering the solutions that are changing the way the world does business and the way governments serve their citizens.

SPONSOR BY:

Cal-Amp · COMPOSITE SOFTWARE · ROGUE WAVE SOFTWARE · SUGARCRM · CENTRICCRM · metallect · WEB-AGE

Receive FREE WebCast Archives of all Entire Conference!

The best news for this year's conference delegates is that your "Golden Pass" registration now gives you full access to all 96 conference sessions. We will mail you the complete content from all the conference sessions in seven convenient DVDs after the live event takes place.

This on-demand archives set is sold separately for $995

Receive 8 Days Worth of Education F

Jeffrey Barr
Amazon
As Web Services Evangelist for Amazon.com, Jeff Barr focuses on creating developer awareness for the Amazon software platform. He has a longstanding interest in Web services and programmatic information interchange. Jeff has held development and management positions at KnowNow, eByz, Akopia, and Microsoft, and was a co-founder of Visix Software. Jeff's interests include collecting and organizing news feeds using his site, www.syndic8.com . He holds a Bachelor's Degree in Computer Science from the American University and has done graduate work in Computer Science at the George Washington University.

Israel Hilerio
Microsoft
Israel Hilerio is a program manager at Microsoft in the Windows Workflow Foundation team. He has 15+ years of development experience doing business applications and has a PhD in Computer Science.

Adam Kolawa
Parasoft
Adam Kolawa, Parasoft co-founder and CEO, is considered to be a visionary in his field. In 1983, he came to the United States from Poland to pursue his Ph.D. In 1987, he and a group of fellow graduate students founded Parasoft to create value-added products that could significantly improve the software development process. Kolawa's years of experience with various software development processes has resulted in his unique insight into the high-tech industry and the uncanny ability to successfully identify technology trends. As a result, he has orchestrated the development of numerous successful commercial software products to meet growing industry needs to improve software quality.

Jason Levitt
Yahoo!
Jason Levitt, Technical Evangelist on creating Flash-based Yahoo! Maps applications.

Duane Nickull
Adobe
As senior standards strategist for Adobe Systems, Duane Nickull is responsible for managing Adobe's participation in OASIS and UN/CEFACT, as well as ensuring that Adobe's enterprise solutions support emerging XML standards. Previously Mr. Nickull co-founded Yellow Dragon Software Corporation, a privately held developer of XML messaging and metadata management software, recently acquired by Adobe. Mr. Nickull currently serves as a vice chair of the United Nations Centre for Facilitation of Commerce and Trade (UN/CEFACT) where he oversees the United Nations Electronic Business strategy and architecture.

Bob Pasker
Azul
Bob Pasker is deputy CTO with Azul Systems. He has been designing and developing networking, communications, transaction processing, and database products for 25 years. As one of the founders of WebLogic, the first independent Java company (acquired by BEA Systems in 1998), he was the chief architect of the WebLogic Application Server. Bob has provided technical leadership and management for numerous award-winning technologies, including the TribeLink series of routers and remote access devices, and the TMX transaction processing system. Bob graduated magna cum laude and Phi Beta Kappa from San Francisco State University and holds a Masters degree from Brown University.

Brian Behlendorf
CollabNet
Brian Behlendorf founded CollabNet, with O'Reilly & Associates, in July 1999. The company provides tools and services based on open source methods. Before launching CollabNet, Behlendorf was co-founder and CTO of Organic Online, a Web design and engineering consultancy located in San Francisco. During his five years at Organic, Behlendorf helped create Internet strategies for dozens of Fortune 500 companies. During that time, he co-founded and contributed heavily to the Apache Web Server Project, co-founded and supported the VRML (Virtual Reality Modeling Language) effort, and assisted several IETF working groups, particularly the HTTP standardization effort.

Marc Fleury
JBoss
Born in Paris in 1968, Marc Fleury got his Ph.D in physics from the Ecole Polytechnique in Paris. He started in Sales at Sun Microsystems France and then moved to the US where he worked on early java enablement of SAP at SAPLabs. Marc started the JBoss project in 1999. An ex-Lieutenant in the paratroopers, Marc holds a degree in Mathematics from the Ecole Polytechnique, a master in Theoretical Physics from the Ecole Normale ULM and was a visiting scientist at MIT during his thesis. Marc's research interest focuses on aspect oriented middleware.

Andy Astor
EnterpriseDB
Andy is President and CEO EnterpriseDB, the world's leading enterprise-class, open source database company. Previously, Andy was vice president webMethods, leading the company's open source, standards, and Web services agendas. Andy was elected twice to the Board of Directors of the Web Services Interoperability Organization (WS-I), and led WS-I's marketing efforts. Prior to joining webMethods, Andy was vice president at D&B, where he led worldwide development of all on-line products. His work at D&B included the development and launch of one of the earliest commercial Web services.

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

Peter Yared
ActiveGrid
Peter Yared is the founder and CEO of ActiveGrid. Most recently, he was CTO of Sun Microsystems's Liberty Network Identity initiative. Mr. Yared was also CTO of Sun Microsystems Application Server Division. Before its acquisition by Sun, Mr. Yared served as CTO of NetDynamics, which pioneered the then-leading J2EE application server. Earlier, Mr. Yared was founder and CEO of JRad Technologies, an enterprise Java company acquired by NetDynamics. Additionally, Mr. Yared was Chief Architect of client/server products at object-oriented tool maker Prograph International and the architect of several mission-critical systems deployed by U.S. government agencies and the GED Testing Service.

David Temkin
Laszlo
David Temkin is Chief Technology Officer of Laszlo Systems, Inc. In this role, he has positioned the company to become the next technology standard for rich Internet applications. Under his direction, Laszlo developed its patent-pending open-source product suite and extended operations to both coasts of the United States. Before founding Laszlo, Temkin was senior director of engineering at Excite@Home where he led a team of 55 engineers, designers and technical writers responsible for developing the company's consumer software. Prior to Excite@Home, Temkin was an engineering manager in the Newton division at Apple Computer and developed enterprise software at EDS.

Kevin Hakman
TIBCO
Kevin Hakman is Co-founder, TIBCO General Interface, TIBCO Software Inc. Prior to TIBCO General Interface, he was the co-founder of Versalent Inc. a leading provider of enterprise client technology. Prior to Versalent, he founded a series of successful emerging Internet technology and ecommerce ventures. He has also written for eBusiness Journal and HotWired.

Coach Wei
Nexaweb
Coach Wei currently serves as CTO for Nexaweb, which develops the leading XML-based rich client technology platform for building and deploying Enterprise Internet Applications. Previously, he played a key role at EMC Corporation in the development of a new generation of storage network management software. Coach is a graduate from MIT, holds several patents, and is an industry advocate for the proliferation of open standards.

Luis Derechin
JackBe
Luis Derechin is CEO and Co-Founder of JackBe. Mr. Derechin has over 12 years of entrepreneurial and management experience. He has been part of the founding team of successful startups, including a catalogue retail company that achieved $15M in sales.

Jouk Pleiter
Backbase
Jouk Pleiter is the CEO of Backbase, a leader in the field of Rich Internet Applications and AJAX development software. Backbase's clients include ING, ABN AMRO, TNT, KPN, Comsys and Heineken. Backbase operates globally with offices in San Mateo (North America) and Amsterdam (Europe). Since 1995, Jouk has been an entrepreneur: he founded three successful Software companies. Prior to Backbase, Jouk was part of the founding team at the web content management company Tridion, where he led the product management operations, and was driving the company's efforts to become a leader in the European WCM software market.

Networking Opportunity of the Year!

SOA 10th International
WebServices
Edge conference

ENTERPRISE> OPENSOURCE CONFERENCE

REAL-WORLD AJAX SEMINAR

Discounted Price
Incl. Golden Pass to Both Conferences* $1,495

Conference Price (On June 5-6, 2006)
Incl. Golden Pass to Both Conferences* $1,695

*BONUS: "Real-World AJAX" Two-Day Seminar!

Attention Sponsors:
A Forum will be available to display leading Web services, OpenSource, and AJAX products, services, and solutions.

TOPICS INCLUDE...

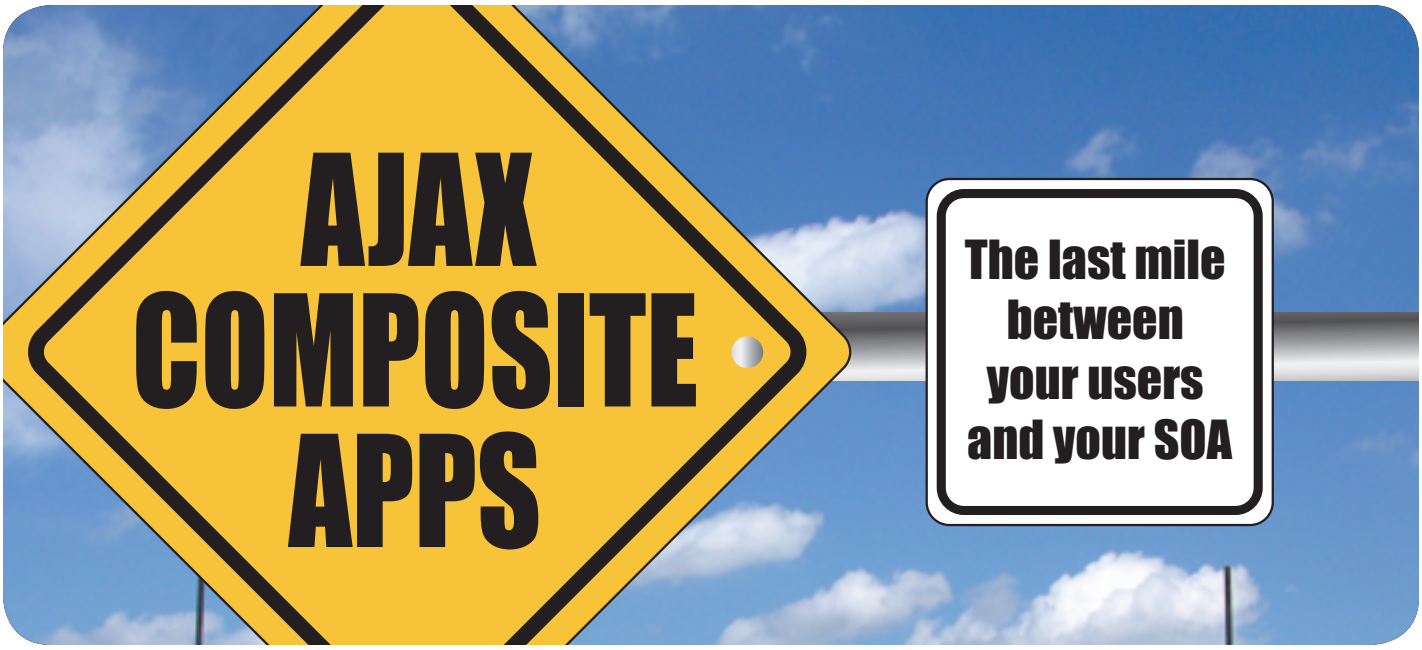
SOA Web Services Edge:
> Transitioning Successfully to SOA
> ebXML
> Orchestration
> The Business Case for SOA
> Interop & Standards
> Web Services Management
> Messaging Buses and SOA
> SOBAs (Service-Oriented Business Apps)
> Delivering ROI with SOA
> Java & XML Web Services
> Security
> Systems Integration
> Sarbanes-Oxley
> Business Process Management

Enterprise Open Source:
> Open Source Licenses
> Open Source & E-Mail
> Databases
> ROI Case Studies
> Open Source ERP & CRM
> Open-Source SIP
> Testing
> Grid Computing
> Open Source Middleware
> LAMP Technologies
> Professional Open Source
> Open Source on the Desktop
> Open Source & Sarbanes-Oxley
> IP Management

AJAX User Experience:
> Rich Internet Applications
> Flash-AJAX Integration
> Developing Interfaces
> AJAX on Rails
> Best Practices
> Enterprise AJAX
> Technology Demos
> Lessons from the Frontline
> Bringing Desktop Apps to the Web
> AJAX Power Panel
> and more...

SYS-CON EVENTS www.EVENTS.sys-con.com

Packed into 2 Information-Filled Days!



In the telecommunications industry there's a special phrase for that bit of technology that carries data from the last pole or relay box into the customer's home. It's called "the last mile" and it's often seen as one of the biggest challenges because this last step in the technology chain can be a considerable physical undertaking. In the IT industry we also have our "last mile": putting the right application in the hands of the end user. Composite applications address this "last mile", combining a rich user interface with SOA-driven application integration technology.

WRITTEN BY
CHRIS WARNER

Composite applications are nothing new. Analysts have been talking about composite applications since the birth of the Internet and in more urgent tones during the Enterprise Application Integration (EAI) generation of integration technology. Today there are plenty of *consumer-focused* composite applications, colloquially referred to as mashups. Popular examples, fueled by the mapping-oriented Web Services provided by Google and others, have been growing at an exponential pace. These applications typically take data from one source of information and plot that data on interactive maps. Literally dozens of these are introduced every week.

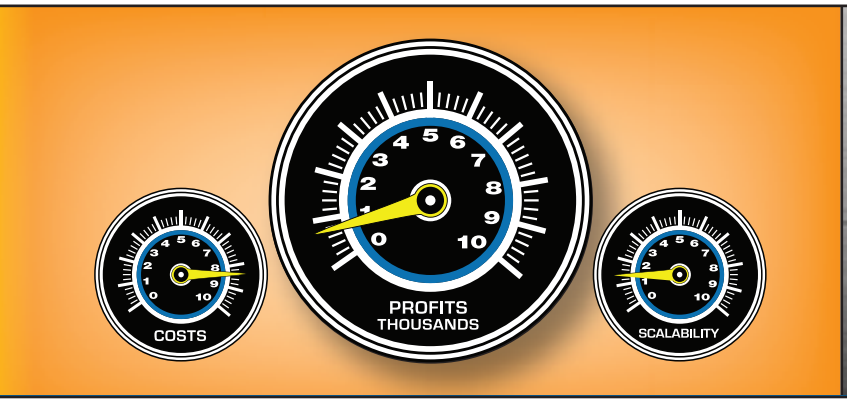
One recent example plots Associated Press news articles by the location of the news story (http://asap.ap.org/fronts/newsmap.s - ASAP); another displays airports with their associated air traffic control delays (http://www.usaflightinsurance.com/gmap.htm - FAA Flight Delay Information); a third displays movies, theaters and show times by location (http://www.mashmap.com/ - Mash Map).

Composite applications are useful for the consumer. This article will discuss the next logical generation of these applications – the business use of composite applications or, as one market analyst put it, "enterprise mashups."

The Basics of Composite Applications

Since the term "composite application" isn't subject to formal definition by any standards-setting body, there's considerable overuse and misuse of the term in software marketing. The "composite application" label has been applied by consumer and business-oriented software vendors alike in such diverse functional areas as Web portals, business process management, document management, knowledge management, and collaboration. A "true" composite application is probably not any one of these things but rather an application that combines services such as intranet-based search engines, applications and databases exposed as Web Services, messaging systems, business intelligence engines and data integration solutions with extranet services such as data sources and application utilities. Simply defined, a composite application combines functionality from multiple applications to support business tasks that span those existing applications.

A slapdash approach won't get you there...



ExtenXLS will.

NEW! Round Trip Reporting™

"ExtenXLS makes it possible to re-use spreadsheets representing a tremendous investment in time and business logic and to 'connect' those spreadsheets to various SQL data sources to provide access to usable data in a way never before possible."

- Project Leader, AT&T

"We use ExtenXLS for 'pushing' accounting data into an Excel template. A key customer wanted web access to Excel format reports. We tried .NET but had problems with preserving charts & formatting. ExtenXLS provides a flexible, cost effective solution that gives us the ability to stay ahead of user requirements."

- Project Leader, John J. McMullen Company

It's What's behind the Dashboard that Counts!

Slapping a dashboard onto a static spreadsheet file may **look** good, but it's like installing a flashy dashboard on an outdated clunker. Install that same front-end on ExtenXLS — **a scalable, server-based spreadsheet engine** — and watch your flashy dashboard come alive.

Now with Round Trip Reporting™

Reuse your existing spreadsheets as a data-entry tool. Modified data is extracted from your spreadsheets and turned into Java Data Objects for reuse in all your programs.

Reach New Heights without the Learning Curve

With the familiar look and feel of Excel™, ExtenXLS eliminates the learning curve imposed on your end-users by other reporting tools. ExtenXLS unlocks the business logic stored in static spreadsheets throughout your organization, saving time and money.

Escape the Gravitational Pull of Obsolete Reporting!

Output to customized HTML for maximum compatibility or to XML for further processing. Keep your users happy with native Excel™ output which preserves the VB macros, images, charts, and other features that transform live data into actionable reports. You can even embed a familiar spreadsheet component in your Swing applications.

Don't rely on a slapdash approach for your mission-critical reporting needs.
Put a rocket under the hood and achieve escape velocity.

Visit the mother ship at: www.extentech.com/jdj and take your free evaluation copy into orbit for a test flight.

EXTENXLS 4™
JAVA | XLS REPORTING TOOLKIT

extenTECH™
Call Us: 415-759-5292

How do composite applications work?
Composite applications include two elements: a rich user interface and service-oriented application integration technology.

Like two streams that join to make a large river, the composite application "buzz" is due to the convergence of two trends: Internet-based user interface technology that's rich (i.e., one that looks and feels like a desktop application but is delivered via a Web browser) and evolving open integration technologies and standards like XML and Web Services.

Read enough of the vendor case studies and you find that the current sweet spot for composite application implementations includes two or more of the following source applications: sales force automation, customer support, contracts/billing, content management, and the ever-present custom legacy application. Composite applications are still new enough that very few qualified and quantified ROI studies exist, but anecdotal evidence indicates that the benefits of composite applications can be great. For IT, composite applications are developed faster than previous integration approaches and can be easier to maintain and upgrade, while also improving the return-on-assets of the existing systems they are built on. For business users, they address issues as far ranging as data quality and data latency, throughput, and end-to-end cycle time, and qualitative areas such as customer service and enhanced ease-of-use.

Now let's take a close look at the underlying architectural components of composite applications.

The 'Application' Side of Composite Application

Internet application development tools have matured greatly in the last decade and they have made great strides towards the goals of true portability, ease of development, and most importantly, true user utility. Internet-based user interface technology has also improved, allowing a "smarter" client that has sophisticated client-side capabilities such as statefulness, fat client-like logic, local data validation, and client-side data control such as local caching and programmatic server interaction, without requiring large downloads or the installation and maintenance of native client-side code.

The most interactive of applications that fit this description are often referred to as "Rich Internet Applications" or RIAs. SOA analyst firm ZapThink estimates that the RIA tools market currently accounts for less than 10% of all application presentation tools but by 2010 will account for over 50% of that market and represent a $1.1 billion industry. There are a number of practical toolsets available to bring a RIA to the user. The three most popular approaches today include Java, Macromedia Flash, and a combination of native browser functionality referred to as AJAX (a combination of Asynchronous Javascript, XML and some supporting technologies). Table 1 compares some the more notable pros and cons of each technology.

In this article, we'll focus on the newest of these approaches, AJAX.

AJAX, or Asynchronous JavaScript and XML, is a Web development technique for creating interactive Web applications. According to the Wikipedia, "The goal of AJAX is to make Web pages feel more responsive by exchanging small amounts of data with the server behind the scenes, so that the entire Web page does not have to be reloaded each time the user makes a change. This is meant to increase the Web page's interactivity, speed, and usability." It's certainly a worthy goal.

Technically, AJAX consists of a combination of:

- Presentation using XHTML and CSS;
- Dynamic display and interaction using the Document Object Model (DOM);
- Data interchange and manipulation using XML and XSLT;
- Asynchronous data retrieval using XMLHttpRequest; and
- JavaScript binding everything together.

AJAX designs are commonly deployed to the user's browser in two parts a pre-built generalized AJAX runtime component, or "client," and your application-specific design specifications. The latter is typically in some XML format. With an AJAX runtime client, the initial download of "code" to the browser will be relatively larger (due to the dual download of AJAX runtime client and your application-specific design specifications) but will make the overall user experience better.

Once the AJAX runtime client is initialized in the browser, it loads the screen layout specification, generates dynamic HTML to render the screen accordingly, and starts to handle all the document object model (DOM) events generated by the user's interaction with the browser. The AJAX runtime client typically handles many user-generated interface events locally, such as field navigation and any associated simple data validation. When communication with the server is necessary (for example, to fetch additional information or to post a transaction), it mediates the network interaction with the server and may, for example, cache results in hidden data tables in the browser to optimize future data-centric activities and manage server interactions better. The AJAX runtime typically communicates with the server in an asynchronous fashion, without stalling a user's interaction with the application or requiring a refresh of the user's HTML interface.

The kinds of tasks an AJAX runtime can handle locally, without server interaction, are rapidly evolving from simple data "checkbox-type" validation constraints such as checking

Technology	Strengths	Weaknesses
AJAX	• Uses technologies built in to most browsers	• IDEs are still maturing
Flash	• Very rich interface capabilities	• Requires browser plug-in • IDE is still animation-centric
Java	• Mature development tools • Large community of skilled developers	• Requires hefty, potentially version-specific plug-in

TABLE 1 | Toolsets for bringing a RIA to the user

for the presence of mandatory data items or date formats to more complex business rules (perhaps originally represented as BPEL fragments) that are executed in the AJAX runtime. This is some of the magic of AJAX, as it lets you move functions from server to client as necessary. A good generalized architecture of AJAX, depicted in contrast with the more traditional Web application architecture, is shown in Figure 1.

Much like the early days of HTML, you *can* code AJAX-based applications by hand in Notepad or vi. But considering the many technologies in AJAX (DOM, CSS, HTML, XML, JavaScript, etc.), maintenance and debugging would be at best difficult. You certainly want to use an AJAX integrated development environment (IDE). A good IDE can generate some or most of this fairly complex code, both the server-resident code and the browser-specific code that's ultimately deployed with and executed by the AJAX runtime client.

AJAX IDEs are still maturing. Here are some important functions to look for when acquiring your AJAX toolset:

- ***Pre-built control library:*** Look for a large library of pre-built visual interface controls like those you would see in a typical client/server development tool. Ask whether the IDE lets you extend the toolset to meet your own unique needs. Business applications often require custom controls that are unique to your organizational needs and composite applications are no exception.
- ***Logic/Data binding:*** The tool should enable the automatic mapping of GUI controls directly to simple services and the generation of code stubs to bind controls to more complex services. In the latter case, this allows the separation of application interface development from business logic/data binding, which can be done in parallel.
- ***Code development support:*** Closely related to logic/data binding, code development support allows for that last bit of hand-coding you need to do in Java or .NET to bind a button to a custom application or API. Does the tool include some type of plug-in (or is entirely resident in) for a code development environment like Eclipse, Visual Studio, and/or Dreamweaver?
- ***Standards support:*** In the next section we'll outline a number of standards important to an AJAX-based composite application architecture. Choose an AJAX IDE that supports the ones that are important to you project. Get a feel for the product's roadmap for future standards and technologies by asking if the vendor participates in standards bodies like OASIS and W3C. And if the tool doesn't support WS-such-and-such today, when will it?
- ***Development repository:*** A repository can be the "lumber yard" from which your composite application engineers pick the raw materials they need to build their composite applications. A repository can store many things: application interfaces exposed as Web Services; logical software services like database tables, stored procedures, and reusable composite application "widgets" such as custom controls; or specialized application binding code. The real trick is to find a repository that makes the storage, searching, and retrieval of these objects easy through simple interaction standards like UDDI 3.0 and WebDAV, and perhaps even an API that allows for advanced programmatic interaction such as the XML Query for Java (XQJ) and the Java API for XML Repositories (JAXR).
- ***Odds and ends:*** Some final items to look for in your AJAX toolset include dynamic charts, multi-language support, style templates, and thorough installation and "getting started" documentation. Most but not all AJAX tools support these areas, so trust but verify.

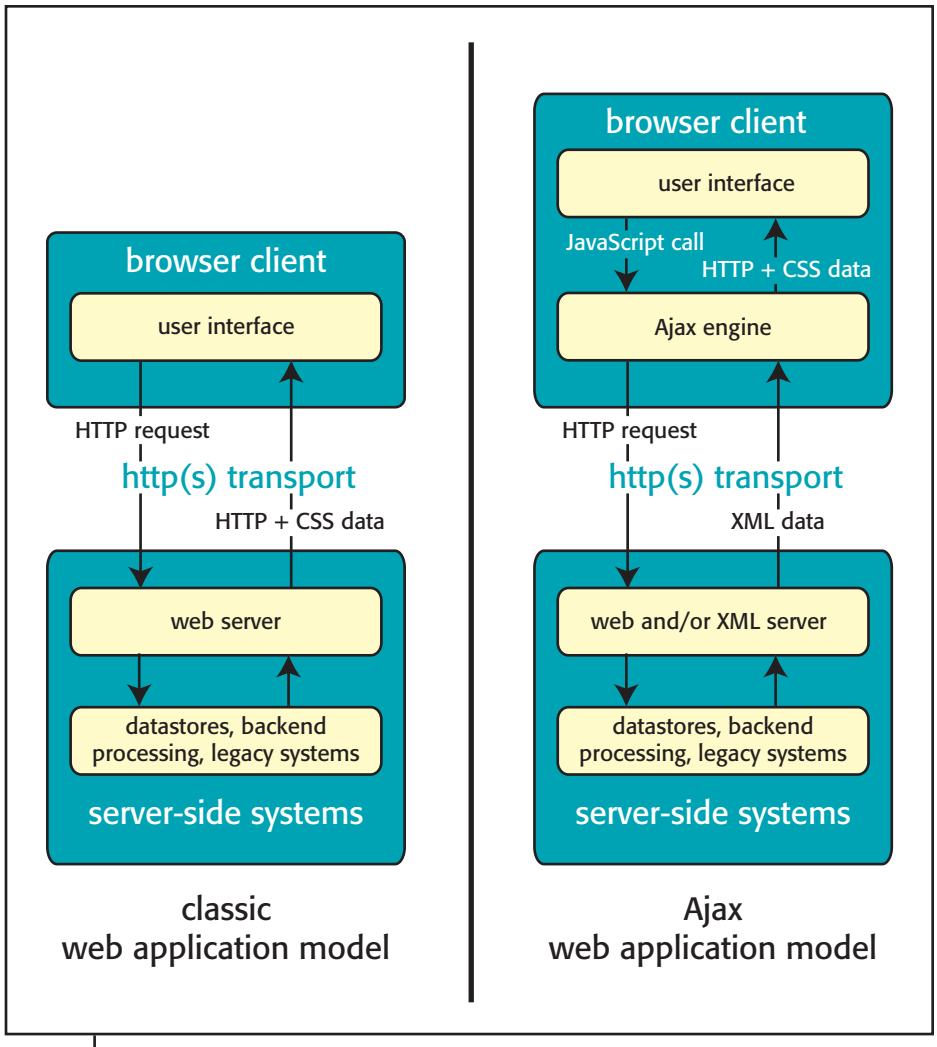


FIGURE 1 | Generalized AJAX architecture

Composite Applications and Portals

Portals were arguably the first generation of composite applications. Today composite applications can offer a much richer application development platform through service-based integration (instead of the kind of integration in the application server that runs the portal) and the highly interactive user environment offered by AJAX-enabled rich Internet applications. With this trend, some analysts have predicted that composite applications signal the beginning of the end for portal-based applications. In reality, the primary functions of portals and composite applications complement each other well. A portal can "wrap" and pass through users seamlessly to a composite application, containing it wholly. Conversely, a composite application can wrap a portal (or portlet more likely), as part of the application, perhaps even invoking it with parameters that tie it into the rest of the composite application. But they have their differences too.

Different tools for different problems – Portals began by aggregating functions that typically didn't require coordination between them or transactions across them. For example, Web portals allow UI designers to assemble useful utilities for their user communities but the Company News portlet in Frame 1 doesn't connect to the Customer Reference Database in Frame 2. They don't focus on what one composite applications vendor described as "flow" and "wiring."

While portal technology has greatly matured in recent years, the "Functionality of Composite applications" section in the main article outlines a number of functions and standards that you won't likely see in everyday portal software. Examples of portal maturity are the OASIS WSRP Producer and Consumer and JSR-168 standards, which allow portals to consume portlets from other places (one through Web Services and the other via Java). This is not a service-oriented integration approach to a composite application but does let a portal vendor simulate it.

Pay me now or pay me later – Portal-based applications are typically HTML-driven and can be counted on for fairly lean-and-small sets of 'code' driven to the user's browser. In contrast, composite applications have a larger upfront "load" since there can be a certain amount of code (like the Ajax runtime client described earlier in this article) as well as data that needs to be cached. After this initial load, however, well-designed composite applications can have developer-controlled incremental exchanges with the server that compare favorably with the complete page or frame loads/reloads common to portals.

Finally, you want to remember your deployment target when choosing a composite application toolset. Today's estimates are that around 75% of browser installations support AJAX. While this may be less of an issue inside corporations that have constant maintenance/upgrade programs, it's important to know this in advance. While this issue isn't exactly unique to AJAX, it can become a real issue when you want to deploy your new composite application to 10,000 business users.

The 'Composite' Side of 'Composite Application'

Perhaps not coincidentally, Service Oriented Architectures (SOAs) fit nicely into the AJAX-based composite application design goal. SOAs, in case you've missed the buzz so far, bring value to an organization by enabling it to expose important business information and logic in a readily accessible way. Technically SOAs are collections of standardized interfaces, or "services" to and from business applications and data sources allowing loosely coupled access to this data and logic through standardized mechanisms such as XML-based messaging over HTTP. (In fact, if you're reading this journal you likely already understand the ubiquity and utility of an SOA.) There are many parts to an SOA, so we'll restrict ourselves to the SOA principles and technologies that are most relevant to composite applications.

The behind-the-curtain component of composite applications loosely corresponds to the higher-order functions of the Web Services protocol stack. Described top-down from the highest to the lowest these are functions that include:

- ***Orchestration*** is functionality that lets the composite application define and execute a "flow" between services Today, this kind of functionality is usually found in Enterprise Service Buses and some BPM products. Popular relevant standards include BPEL (Business Process Execution Language) and BPML (Business Process Modeling Language). BPEL, in particular, is increasingly accepted as an interoperable standard for orchestrated process flows. It emphasizes a loosely coupled orchestration solution that reuses lower-level services (instead of hand-coding an orchestration that is tightly coupled to lower-level APIs). BPEL also includes native functions to persist long-running orchestrations.
- ***Choreography,*** or the interaction between independent processes, can be more complex, involving business rules rather than simple conditional logic, and is often laden with elaborate rules for handling exceptions. Because of these factors, it's a tougher nut to crack than orchestration. Relevant standards include the well-supported BPEL, the less popular WSCI, the newer WSCDL, and even human-centric standards like XPDL and the proposed "BPEL for People." There may not be a perfect fit within the standards world for choreography, , as evidenced by recent attempts to "abstract" BPEL for this type of purpose.
- ***Composition*** creates coarse-grained services from fine-grained services. The purpose of composition is to eliminate direct application-to-application connections and instead create "composed" services that are reusable many times. For example, fine-gained data services "Get Name," "Get Balance," and "Get Credit Score" can be accessed by every AJAX composite application that needs customer information. But it's smarter to compose a corresponding coarse service, "Get Customer Info," accessible by all applications, particularly if this new service has extra logic like the resolution of data semantics or data rules. BPEL can be used appropriately for composition assuming that your "services" are Web Services only; if your environment includes other "services" such as a SQL interface or JMS API, you may want leave the composition functionality to a tool other than your AJAX tool, such as an Enterprise Service Bus.
- Finally, ***transaction control*** manages a reliable relationship between two or more services. Although you could explicitly connect to the source systems yourself and perform your own transaction management, this is hard enough even when connecting to two instances of the same type of system, like two Microsoft SQL Server databases. However, in real-world situations, transactions are just as likely to span multiple, heterogeneous systems over multiple communication channels and APIs such as JDBC, SOAP over HTTP,

JMS, and so on. For these reasons, you'll probably want to rely on the more sophisticated, dedicated transaction management capabilities of an Enterprise Service Bus or TP monitor. > Looking forward, a standard you might employ is WS-CAF, a draft OASIS standard "for supporting coordinated and transactional compositions of multiple Web Services applications," that is composed of three existing OASIS specifications: WS-Context, WS-Coordination Framework, and WS-Transaction Management. When complete and approved, WS-CAF could eliminate much of the custom coding inherent in today's approaches to Transaction Control.

In addition to the above, you want to remember a few fundamental areas of technology:

- ***Security:*** You should design your entire composite application stack to allow for security to be passed from the highest to lowest level service. This can be complicated by simple issues like different IDs, passwords, and rights. The solution is the management of identity through a "policy" that can be used in all parts of the composite application instance lifecycle. This complex area makes WS-Security and SAML must-have standards to include in your composite application project.
- ***Management:*** Once you've deployed your composite application, you'll want to know that it's running and available. Your AJAX runtime environment should include some kind of management interaction through SNMP or JMX.

It's important to realize from this long list of functions that there's unlikely to be any one AJAX IDE tool that addresses all of these functions unless you're willing to include a lot of homegrown code as part of the answer. For example, you might already own an orchestration tool (sometimes called an Enterprise Service Bus or ESB). But using more than one tool to complete your composite application project need not be a deal killer. Many vendors "pre-assemble" some or all of the tools you need, including an AJAX IDE, an ESB, a development repository, and other supporting software. And you may find that your tools interoperate through standardized protocols.

Obviously, all this talk of services means that the composite application development team has to be aware and involved in the service enablement of the source systems. Getting your legacy service enablement wrong will result in longer development times and less value for the ultimate user. Issues that a legacy service enablement vendor can help you address might include:

1. ***Level of service enablement:*** Do you enable the legacy database, the legacy application, or the legacy user interface? These are complementary technologies that tackle legacy systems in different ways and with different tradeoffs.
2. ***Type of service enablement***: Service interfaces can be through pre-SOA standards like SQL, the exchange of XML-formatted documents, a more formal WSDL contract, or some sort of messaging system.
3. ***Granularity of function:*** It's rare to find a match between the level-of-functional granularity natively offered by a packaged application and what your composite application would "prefer" to consume, so some kind of orchestration, choreography, or composition is often necessary to aggregate a set of lower-level functions exposed by an application into higher-level business services that can be used directly by other consumers.

Finally, keep in mind that not all of the technologies and standards may be relevant to your composite application. Equally important, we couldn't cover them all in this article, so we focused on the most popular and most formal of them. If you embark on a composite application project, you should try to keep up with the relevant technologies and standards.

Finishing the Last Mile

Composite applications benefit the businessperson and the technologist. Business users get a richer user experience. IT has a smarter way to create composite Web applications. Make sure you have a plan to walk this last mile. It's worth the trip.

About the Author

Chris Warner is director of technical marketing at Software AG in Reston, VA. Software AG's SOA integration suite, crossvision, includes an AJAX-based composite application tool, Application Composer. Crossvision also includes Web Service-enabling software for legacy data sources and applications, an Enterprise Service Bus, an enterprise information integration tool, and a centralized SOA registry-repository.
In true composite application style, he wants to thank John Fitzgerald, Tom Michaud, and Ken Stewart for their contributions to this article.
chris.warner@softwareagusa.com

SOA: Focus is on Approaches Not Technology

So what's hot these days in the world of SOA? Governance, registries, orchestration...? Nope. As folks looking to implement SOA seek that first killer project the emphasis is on what to do, not what you use, and that's exactly the right way to think. As SOA becomes more of a reality among the Global 2000, the focus on discipline as a concept will be as important as solutions, perhaps more important than many expect. Let me explain.

WRITTEN BY
DAVID S. LINTHICUM

This is a clear trend that I see in the SOA space, those charged with building SOAs in their enterprise are working on establishing approaches to the implementation of their SOA instance, and aren't yet looking for "key enabling SOA technology," at least not yet. This means that they are setting up methodologies, defining deliverables, and how all of these artifacts are related. What's more they're focusing on education, understanding just what they're doing before they do it. We've learned from the past that quick movements towards a technology trend, without the proper amount of upfront thinking typically means failure.

If you ask me this is a good trend. While SOA is attractive as the hot new technology, or perhaps the reinvention of existing technology, most enterprise architects view SOA as a key strategic initiative, and aren't willing to risk failure. This is evident in the SOA's slow uptake, now accelerating, as larger organization do some advanced planning as well as get a bit smarter in dealing with the notion of SOA, keeping in mind that it's really a journey not a destination.

Finding an approach isn't that easy, however. There certainly is a great deal written on the topic by some very smart people, but the right approach for your particular organization may be a bit different from the generalized approaches/methodologies you see around today. In other words, you'll be doing some planning to create the plan. For instance, when I wrote the 12 Steps to SOA a few years ago, I was creating a general-purpose checklist of both tasks and deliverables to assist organizations in implementing their SOA. However, now I'm finding that some organizations have expanded it to 14 steps, and other reduced it to 11, again customizing the approach to their specific requirements, and that's okay by me.

What this trend will result in, just you wait and see, is a focus on design and planning tools more than implementation technology. Truth be told, there aren't many design and planning tools out there for SOA, and the ones on the market aren't impressive at all. Hopefully, we'll see some creative and well-funded start-ups in this space soon. Categories of planning tools should include:

1. **Modeling and implementation.** Holistic modeling of the SOA and all of its working parts.
2. **Security design and implementation.** The

> Those that jump directly to 'what to use' are fooling themselves into thinking that a lack of understanding of the problem domain will be overshadowed by killer technology – that never works

ability to figure out how you're going to secure and govern your SOA.

3. **Semantic understanding and metadata modeling.** The ability to identify all application semantics and define a common metadata model.
4. **Service design and implementation.** The ability design services properly, implement them, and track them.
5. **Orchestration and process modeling.** The ability to model processes and implement them directly from the model.

There are tools on the market that do some of these things; you just have to find the ones that work for you. First I would suggest, however, that you pick your approach then pick your tools. Moreover, always consider both your tool selection and approach as fluid notions. Don't be afraid to change them as your needs change.

So, if you're focusing on what you need to do, you're in good shape. Those that jump directly to "what to use" are fooling themselves into thinking that a lack of understanding of the problem domain will be overshadowed by killer technology. That never works.

About the Author

David S. Linthicum is the author of three books on application integration and SOA, a frequent speaker at industry conferences, and the host of the "Service-Oriented Architecture Expert Podcast" (www.soaexpertpodcast.com).

linthicum@att.net

BY NOW THERE ISN'T A SOFTWARE DEVELOPER ON EARTH WHO ISN'T AWARE OF THE COLLECTION OF PROGRAMMING TECHNOLOGIES KNOWN AS AJAX!

REAL-WORLD AJAX SEMINAR

Hilton — April 24, 2006, San Jose, CA

The Roosevelt Hotel — June 5-6, 2006, New York, NY

AjaxWorld Conference & Expo, Rich Internet Applications — Oct 3-4, 2006, Santa Clara, CA

How, in concrete terms, can you take advantage in your own projects of this newly popular way of delivering online content to users without reloading an entire page?

How soon can you be monetizing AJAX?

This "Real-World AJAX" one-day seminar aims to answer these exact questions...

HURRY! LIMITED SEATING THESE SEMINARS WILL SELL-OUT CALL 201-802-3022 TO REGISTER!

Led by "The Father of AJAX" himself, the charismatic Jesse James Garrett of Adaptive Path, "Real-World AJAX" has one overriding organizing principle: its aim is to make sure that delegates fortunate enough to secure themselves a place – before all seats are sold out – will leave the seminar with a sufficient grasp of Asynchronous JavaScript and XML to enable them to build their first AJAX application on their own when they get back to their offices.

Jeremy Geelan
Conference Chair, Real-World AJAX
jeremy@sys-con.com

Register at...
www.AJAXSeminar.com

April 24, 2006 (San Jose, CA)	$1,195
See website or call for group discounts	
June 5-6, 2006* (New York, NY)	$1,295
See website or call for group discounts	
October 3-4 (Santa Clara, CA)	$1,495
See website or call for group discounts	
On-Deman Online Access (Any Event)	$695
Purchase both New York and San Jose seminars for on-demand access	

*Includes SOA Web Services Edge plus Enterprise Open Source Conference & Expo

LIVE SIMULCAST!
AROUND THE WORLD ON SYS-CON.TV

SPONSORED BY

Adobe · telerik · BACKBASE · TIBCO / Perficient · Microsoft

What you'll walk away with...

A New understading of AJAX and how to make it work for you, plus:

A) **Conference Proceedings Binder**
Full-conference proceedings in a self-contained commemorative binder.

B) ***Ajax in Action* Book**
A copy of the Best-Selling AJAX book by Dave Crane.

C) ***Real-World AJAX "Secrets of the Masters"* Book + DVD**
New AJAX book edited by Dion Hinchcliffe (Release Date: Summer 2006)

D) **Notebook**
Executive style keeps you looking sharp for note-taking wherever your business might take you!

E) **Computer Back-Pack**
A hip new way to conceal your laptop! The Urban Wonder Compu-Pack is an ingenious creation - a 600-denier polycanvas and nylon knapsack!

F) **T-Shirt**
100% preshrunk 6.1 oz. heavyweight cotton tagless shirt features shoulder-to-shoulder tape, and double-needle sleeves and bottom.

G) **Porcelain Mug**
Lightweight and durable, you can use it as a teacup, coffee mug or keep it as a pencil holder.

H) **Satin Silver Contemporary Pen**
Satin-silver plastic barrel with the look and feel of brass. Rubber comfort grip improves writing control and comfort.

I) **Real-World AJAX Seminar on DVD**
Watch complete seminar video and slide presentations (Release Date: Summer 2006)

Featuring... Real-World AJAX Rock Stars!

FACULTY

Jesse James Garrett (San Jose April 24)
Father of "AJAX" Who Coined the Term in 2005

Jesse James Garrett is the Director of User Experience Strategy and a founding partner of Adaptive Path, the world's premier user experience consulting company. He is author of The Elements of User Experience (New Riders), and is recognized as a pioneer in the field of information architecture. Jesse's clients include AT&T, Intel, Crayola, Hewlett-Packard, Motorola, and National Public Radio. Since starting in the Internet industry in 1995, Jesse has had a hands-on role in almost every aspect of Web development, from interface design and programming to content development and high-level strategy. Today, information architects around the world depend on the tools and concepts he has developed, including the widely acclaimed "Elements of User Experience" model. He is co-founder of the Information Architecture Institute, the only professional organization dedicated to information architecture. He is also a frequent speaker and writer whose work has appeared in numerous publications, including New Architect, Digital Web, and Boxes and Arrows.

Adam Bosworth (San Jose April 24)
Vice President of Engineering, Google
One of the Fathers of XML & the Creator of MS Access

Adam Bosworth is Vice President of Engineering, Google. He joined Google in 2005 from BEA Systems, where he was Chief Architect & Senior VP of Advanced Development. Prior to joining BEA, Bosworth co-founded Crossgain, a software development firm acquired by BEA. Known as one of the pioneers of XML, he previously held various senior management positions at Microsoft, including General Manager of the WebData group, a team focused on defining and driving XML strategy. While at Microsoft he was also responsible for designing and delivering the Microsoft Access PC Database product and assembling and driving the team that developed the HTML engine of Internet Explorer 4.0.

Dion Hinchcliffe (San Jose April 24)
Cofounder & CTO, Sphere of Influence Inc.
Editor-in-Chief, Web 2.0 Journal

Dion Hinchcliffe, newly appointed editor-in-chief of SYS-CON's pioneering Web 2.0 Journal, is cofounder and chief technology officer for the enterprise architecture firm Sphere of Influence Inc., in McLean, Virginia. A veteran of software development, Dion works with leading-edge technologies to accelerate project schedules and raise the bar for software quality. He is highly experienced with enterprise technologies and he designs, consults, and writes prolifically. Dion actively consults with enterprise IT clients in the federal government and Fortune 1000. He is a frequent speaker on AJAX, Web 2.0 and SOA and is currently the top-read SYS-CON.com blogger.

Christophe Coenraets (San Jose April 24)
Senior Technical Evangelist, Adobe
AJAX/Flex Integration Guru

Christophe Coenraets currently works as a Senior Technical Evangelist at Adobe. Before joining Adobe, Christophe was an evangelist at Macromedia, focusing on Rich Internet Applications and Enterprise integration. Prior to Macromedia, Christophe was the head of Java and J2EE Technical Evangelism at Sybase, where he started working on Java Enterprise projects in 1996. Before joining Sybase in the US, Christophe held different positions at Powersoft in Belgium, including Principal Consultant for PowerBuilder, and Manager of the Professional Services organization. Before joining Powersoft, Christophe worked as a developer and architect on several retail and BPM projects. Christophe has been a regular speaker at conferences worldwide for the last 10 years.

Paul Rademacher (San Jose April 24)
Google, Creator of HousingMaps.com

Paul Rademacher is the creator of HousingMaps.com, which combined Craigslist and Google Maps for the first web mashup. Paul holds a Ph.D. in Computer Science from UNC-Chapel Hill, and worked as an R&D Engineer at Dreamworks Animation on such movies as Shrek 2 and Madagascar. Since creating HousingMaps, Paul is now at Google.

Jouk Pleiter (San Jose April 24)
Co-Founder & CEO of Backbase

Jouk Pleiter is the CEO of Backbase, a leader in the field of Rich Internet Applications and AJAX development software. Backbase's clients include ING, ABN AMRO, TNT, KPN, Comsys and Heineken. Backbase operates globally with offices in San Mateo (North America) and Amsterdam (Europe). Since 1995, Jouk has been an entrepreneur: he founded three successful software companies. Prior to Backbase, Jouk was part of the founding team at the web content management company Tridion, where he led the product management operations, and was driving the company's efforts to become a leader in the European WCM software market. Jouk previously was part of the founding team at the Interactive Agency Twinspark where he grew the company to a leading market position in Europe and was instrumental in the sale of Twinspark to Agency.com. He has an MBA from the University of Groningen.

Kevin Hakman (San Jose April 24)
Director of Product Marketing for TIBCO
General Interface TIBCO Software

Kevin Hakman is the director of product marketing for TIBCO General Interface, the award winning AJAX and Rich Internet Application framework and toolkit. Kevin Hakman pioneered AJAX in the enterprise co-founding General Interface in 2001. Since that time General Interface (aka 'GI') has been powering Web applications that look, feel and perform like desktop applications, but run in the browser at Fortune 500 and U.S. Government organizations. General Interface was also the first to use its own toolkit to provide full visual tooling for AJAX when it released it's 2.0 Version in 2003. TIBCO acquired General Interface in 2004 to extend its vision for service oriented applications to the end user. Kevin is a contributor to the SOA Web Services Journal and the AJAX Developer's Journal.

Shanku Niyogi (San Jose April 24)
Product Unit Manager of the UI Framework and Services Team Microsoft Corporation

Shanku is Product Unit Manager of the UI Framework and Services (UiFX) team, which is responsible for delivering high-productivity UI framework technologies for the .NET platform, including ASP.NET, Atlas, Windows Forms, and frameworks for smart clients. Prior to his current role, Shanku was Group Program Manager of the Web Platform and Tools team on the Whidbey release of ASP.NET and Visual Web Developer. Shanku joined Microsoft in 1998 as a developer, having spent several years shipping products in the Windows ISV industry. Shanku holds a Bachelor of Mathematics degree in Computer Science from the University of Waterloo.

Coach Wei (New York June 5-6)
Chairman, Founder and CTO, Nexaweb
The Creator of First Commercial AJAX Applications

Coach Wei combines in-depth IT industry expertise with extensive education and research experience at MIT to drive technology innovation and business direction for Nexaweb. He founded Nexaweb in 2000 and served as CEO until summer 2003. Before founding Nexaweb, Coach architected and designed enterprise software for managing storage networks at EMC Corporation. As a graduate researcher at MIT, Coach developed software and hardware systems for non-destructive evaluation as well as signal/image processing algorithms. Coach was a finalist in the 1999 MIT $50K entrepreneurship competition and holds several U.S. patents. An accomplished writer and speaker, Coach has published numerous articles on topics including: AJAX, J2EE and .NET, RIA development, XML, signal/image processing, composite materials and ultrasonic imaging. He has spoken at top industry events, such as JavaOne and Web Services Edge. Coach holds an MS in information technology from MIT.

Ajit Jaokar (New York June 5-6)
CEO, futuretext
Author, "Mobile Web 2.0"

Ajit Jaokar, based in London (England), is the CEO of a publishing company, futuretext (www.futuretext.com). He is currently writing a book about Mobile Web 2.0 (Mobile Web 2.0: The Innovator's Guide to Developing and Marketing Next Generation Wireless / Mobile Applications). Ajit also chairs Oxford University's Next-Generation Mobile Applications Panel and, since January 2006, has been a member of the Web 2.0 Workgroup. In his "Real-World AJAX" conference session, Ajit will discuss the "AJAX Use in Mobile Applications" as part of the wider impact of Web 2.0, sometimes referred to as the "Global SOA."

Jonas Jakobi (New York June 5-6)
AJAX Evangelist and Co-Author, "Ajax and JSF: Friend or Foe?"
Jonas will autograph a copy of his book for all delegates!

Jonas Jacobi is a principal product manager and evangelist for Oracle's Java/J2EE tool offering, JDeveloper, and over the past three years has been responsible for JavaServer Faces, Oracle ADF Faces, and Oracle ADF Faces Rich Client development features within Oracle JDeveloper. Jonas has been in the software business for 15 years. Prior to joining Oracle, he worked at several software companies in Europe, covering many roles including support, consulting, development, and project team leadership. Jonas' new book "Ajax and JSF: Friend or Foe?" released by Apress on February 25, 2006.

John Fallows (New York June 5-6)
AJAX Evangelist and Co-Author, "Ajax and JSF: Friend or Foe?"
Jonas will autograph a copy of his book for all delegates!

John Fallows, former lead developer for Oracle ADF Faces Rich Client, has been working in distributed systems for over a decade. After five years spent focused on designing, developing the JavaServer Faces standard to provide AJAX functionality, playing a leading role in the Oracle ADF Faces team, he recently joined an AJAX start-up. Originally from Northern Ireland, John graduated from Cambridge University in the United Kingdom and has worked in the software industry for more than ten years. Prior to joining Oracle, he worked as a research scientist for British Telecommunications Plc.

Steve Benfield (New York June 5-6)
Well-known AJAX Evangelist and CTO of Agentis Software
Steve's first talk on "Aspect-Oriented Programming & AJAX"

Steve Benfield is CTO of Agentis Software and one of the pioneers of AJAX technology, a gifted writer and a technical visionary. A technology marketeer and strategist with 20 years of software entrepreneurism experience, a combination of qualities that made him the perfect choice of editor-in-chief for SYS-CON Media's inaugural publication 12 years ago. Steve's proven ability to determine marketing and technology strategies that align with market needs led to successful stints at SilverStream, where he started as technology evangelist and ended as CTO, and at ClearNova, an open source AJAX company, where he was CTO and AJAX evangelist.

Jeremy Geelan
Conference Chair
Group Publisher & Editorial Director, SYS-CON

Jeremy Geelan is group publisher and editorial director of SYS-CON Media, and is responsible for all print titles and online i-technology portals for the firm. He regularly hosts SYS-CON.TV, is executive producer of the "Power Panels with Jeremy Geelan" iTV series, and represents SYS-CON at conferences and trade shows, speaking to technology audiences both in North America and overseas. His i-Technology Blog is at jeremy.linuxworld.com and he is conference chair of the upcoming iTVcon - "Internet TV Conference & Expo 2006".

For more information...

Call 201-802-3022 or email events@sys-con.com

SYS-CON EVENTS **For more great events visit www.EVENTS.SYS-CON.com**

NOTE: SPEAKER LINE-UP SUBJECT TO CHANGE WITHOUT NOTICE
VISIT WWW.AJAXSEMINAR.COM FOR THE MOST COMPLETE UP-TO-DATE INFORMATION

Welcome to the Future of Video on the Web!

REGISTER NOW!
www.iTVcon.com

CALL FOR PAPERS NOW OPEN!

LIVE SIMULCAST!
AROUND THE WORLD ON SYS-CON.TV

iTVCON.COM
INTERNET TV CONFERENCE & EXPO 2006

Coming in 2006 to New York City!

"Internet TV is wide open, it is global, and in true 'Web 2.0' spirit it is a direct-to-consumer opportunity!"

For More Information, Call 201-802-3023 or Email itvcon@sys-con.com

Welcome to the Future!

Did you already purchase your ".tv" domain name? You can't afford not to add Internet TV to your Website in 2006!

2005 was the year of streaming video and the birth of **Internet TV**, the long-awaited convergence of television and the Internet. Now that broadband is available to more than 100 million households worldwide, every corporate Website and every media company must now provide video content to remain competitive, not to mention live and interactive video Webinars and on-demand Webcasts.

20 years ago the advent of desktop publishing tools opened the doors for the creation of some of today's well-known traditional print media companies as well as revolutionized corporate print communications. Today, with maturing digital video production, the advent of fully featured PVRs, and significant advances in streaming video technologies, **Internet TV** is here to stay and grow and will be a critical part of every Website and every business in the years to come.

It will also very rapidly become a huge challenge to network and cable television stations: **Internet TV** is about to change forever the $300BN television industry, too.

The Internet killed most of print media (even though many publishers don't realize it yet), Google killed traditional advertising models, and **Internet TV** will revolutionize television the way we watch it today. You need to be part of this change!

Jeremy Geelan
Conference Chair, iTVCon.com
jeremy@sys-con.com

PRODUCED BY SYS-CON EVENTS

List of Topics:

- Advertising Models for Video-on-demand (VOD)
- Internet TV Commercials
- Mastering Adobe Flash Video
- How to Harness Open Media Formats (DVB, etc)
- Multicasting
- Extending Internet TV to Windows CE-based Devices
- Live Polling During Webcasts
- Video Press Releases
- Pay-Per-View
- Screencasting
- Video Search & Search Optimization
- Syndication of Video Assets
- V-Blogs & Videoblogging
- Choosing Your PVR
- Product Placement in Video Content
- UK Perspective: BBC's "Dirac Project"
- Case Study: SuperSun, Hong Kong

Track 1	Corporate marketing, advertising, product and brand managers
Track 2	Software programmers, developers, Website owners and operators
Track 3	Advertising agencies, advertisers and video content producers
Track 4	Print and online content providers, representatives from traditional media companies, print and online magazine and newspaper publishers, network and cable television business managers

XML JOURNAL SUPPLEMENT

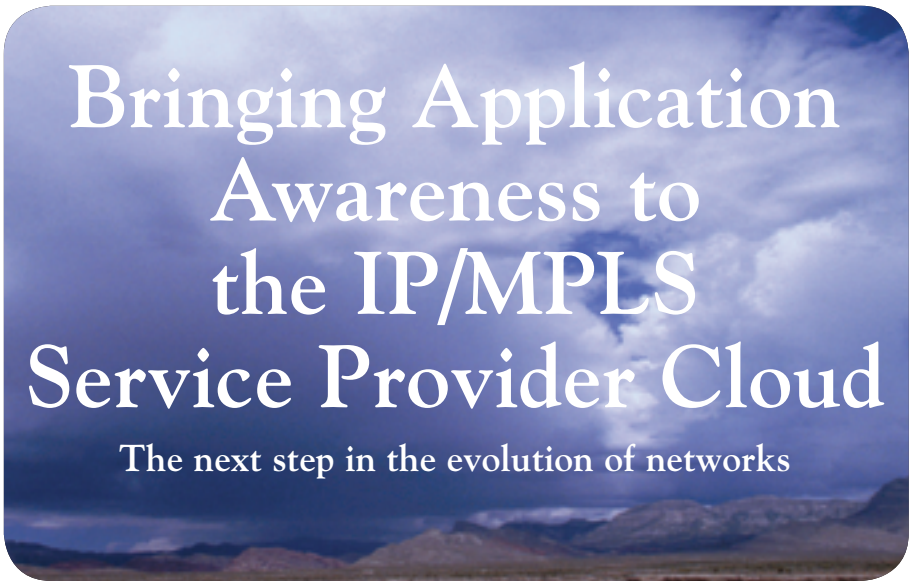
This Month

Bringing Application Awareness to the IP/MPLS Service Provider Cloud

Jonathan Bosloy

Service Oriented Architecture (SOA) and Web Services (along with the underlying XML protocol) promise to greatly simplify the implementation of distributed computing applications, both within the enterprise and between enterprises. The widespread acceptance of Web Services by application vendors and across operating system middleware will lead to simplified interoperability, thus allowing for increased business agility and lower development costs.

Today enterprises can use IP (Internet Protocol) Virtual Private Network (VPN) offerings from their IP service provider for secure high-performance interconnect services both within the enterprise's own sites and with partners. Price erosion in voice services and intense competition in data services is leading many IP-based service providers to seek out new revenue opportunities by offering new services to their enterprise customers. As service providers move to offer innovative new Web Services, enterprises will benefit just as they did when service providers moved from traditional private line



XML-Based Interop, Close up

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

It is our privilege to bring XML-Journal directly to readers of Web Services Journal, and vice versa. Anyone already familiar with the Web services world of SOAP, UDDI, and WSDL will find here articles and features each month that will interest them – about the cutting-edge technologies and latest products that are changing not only our industry, but the way the world exchanges information. To make it easy for you to find your way around, we have four distinct sections:

Content Management:
Organization, dissemination, and presentation of information

Data Management:
Storage, transformation, representation, and general use of structured and unstructured data

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

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

Bringing Application Awareness to the IP/MPLS Service Provider Cloud

The next step in the evolution of networks

WRITTEN BY
Jonathan Bosloy

Service Oriented Architecture (SOA) and Web Services (along with the underlying XML protocol) promise to greatly simplify the implementation of distributed computing applications, both within the enterprise and between enterprises. The widespread acceptance of Web Services across operating systems, middleware, and application vendors will lead to simplified interoperability, thus allowing for increased business agility and lower development costs.

Today enterprises can use IP (Internet Protocol) Virtual Private Network (VPN) offerings from their IP service provider for secure high-performance interconnect services both within the enterprise's own sites and with partners. Price erosion in voice services and intense competition in data services is leading many IP-based service providers to seek out new revenue opportunities by offering new services to their enterprise customers. As service providers move to offer innovative new Web Services, enterprises will benefit just as they did when service providers moved from traditional private line services to IP-based services.

The next step in the evolution of networks is the move to application-aware networking. Networks will become "smarter" by identifying, prioritizing, and moving traffic faster. In addition, distributed application-awareness in the Wide Area Network (WAN) provides a cost-effective and scalable solution for messaging applications in many sectors from financial services to supply chain services.

Today's Situation

Currently, the wide area network is unaware of the application messages that flow across it. A service provider network can offer differentiated Quality of Service (QoS) to control latency and packet loss for applications, but it's not actually aware of the applications being supported.

In the point-to-point integration across a traditional WAN, as shown in Figure 1, the number of point-to-point relationships explodes as the complexity of the Service Oriented Architecture expands across the WAN. While Web Services support location transparency through service registries, services may still need to be moved due to faults, e.g., moving services to a backup location. Techniques such as intelligent DNS can be used to redirect requests to a backup location, but they have limited flexibility.

Intelligent routing of application messages is also not easily available with the architecture shown in Figure 1. For example,

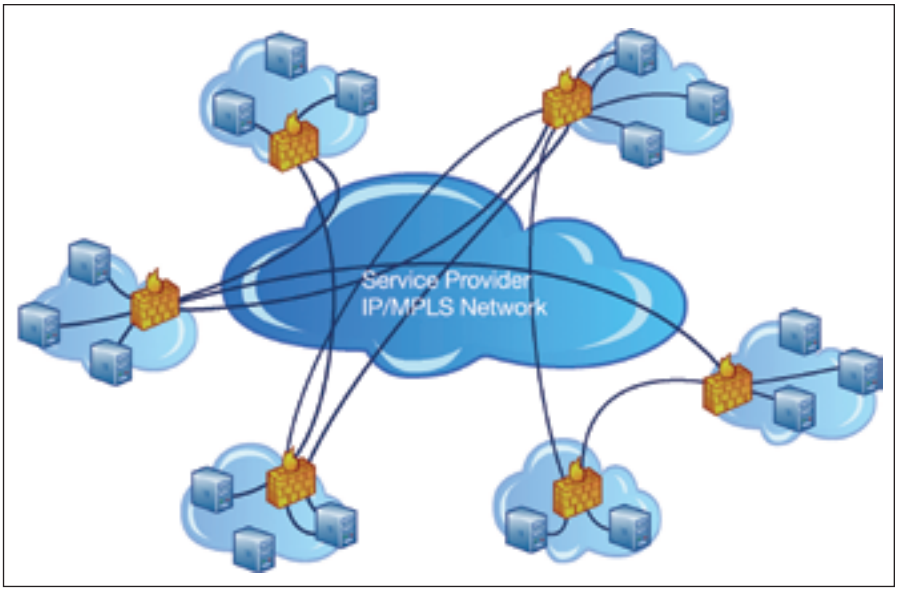


Figure 1 • Today's WAN isn't application-aware and simply provides routing and quality of service on the basis of IP packets.

consider an enterprise that wants purchase orders over a certain amount to go to one data center and other purchase orders to go to a different data center. If a Web Service interface is used to submit purchase orders, the invoking application would have to be aware of a policy to invoke the service and connect to the correct Web Service endpoint. In a truly loosely coupled environment, the invoking application shouldn't have to be aware of the business logic of its partners.

Point-to-point integration also requires the service requestor and service provider's Web Services endpoints to use the same transport protocol. In a loosely coupled architecture, the service requestor is shielded from such details.

There are also advantages to event-driven architectures in which an application can generate events that may be of interest to many other applications. Normally enterprise middleware can provide publish/subscribe services to loosely couple the applications. This means that the application that generates an event doesn't need to be aware of the applications that are interested in the event. Loose coupling increases business agility since new applications can be added quickly without disturbing existing applications. While such middleware solutions are available within the enterprise, they are typically difficult to scale across the WAN and difficult to implement between enterprises.

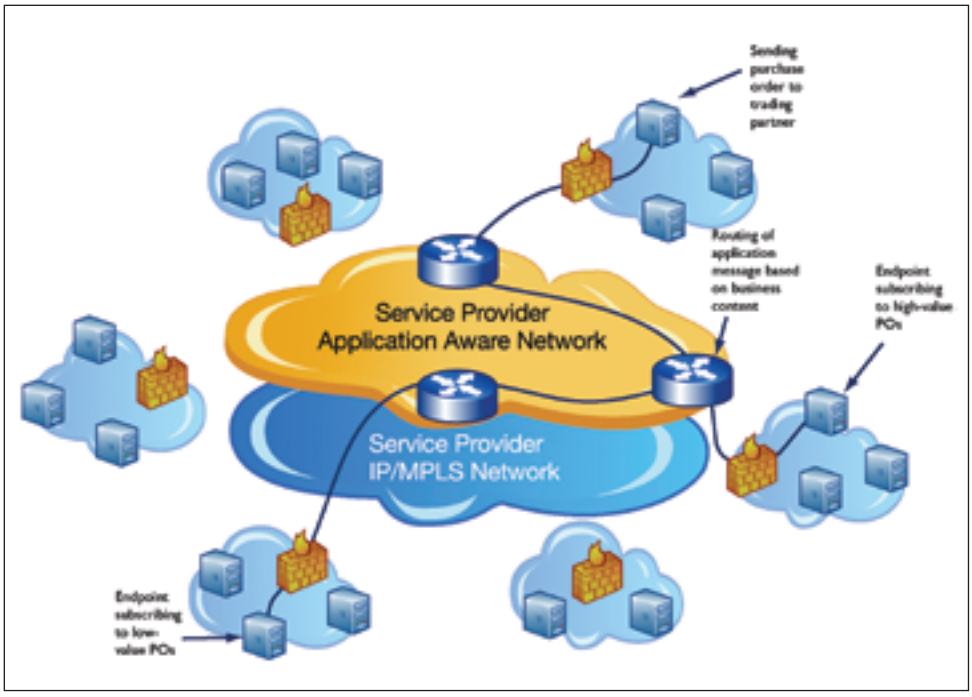


Figure 2 • An application-aware service provider network brings value-added application services to the enterprise.

Application-Aware Networking: Enabling Value-Added Services

Application-aware networking addresses two current distributed technology challenges: first, the inefficiencies associated with middleware, and second, speed and scale. Speed is obvious – networks always have a need for speed – but the speed required for today's real-time financial networks, for example, is a truly demanding benchmark (see the section on financial services below). Middleware – the software that connects network applications – is an integral part of most enterprise data centers. Middleware acts as a translator between incompatible applications by exchanging information in a common format, usually XML, to distribute messages between applications, distributed geographies, or even different companies. These middleware deployments are very complex, add performance overhead, and are challenging to manage. Industry experts have recently called for this middleware function to be moved into the core of the network as a shared resource to simplify the task of running distributed applications. Today's applications become value-added services that take advantage of application-aware, silicon-based hardware to manage information distribution at wire-speed and add a layer of intelligence to the network.

Most service providers have spent the last few years upgrading their core networks to the latest IP standards at great financial expense. Although service providers are very aware of the critical need to develop new revenue-generating services, they are naturally reluctant to make any additional changes to the core of their network. A successful application-aware networking approach should function as an overlay to the existing network to avoid the risk associated with once again upgrading the entire network. Value-added services can then be delivered over the existing infrastructure and capture new revenue.

As shown in Figure 2, the service provider's application-aware network routes messages more efficiently. For example, applications can connect to the network and automatically receive messages of interest. Other applications can send messages without any knowledge of the routing policies being used by other parties. This allows intelligent application message routing across the WAN, leading to true loosely coupled applications.

The application-aware network also allows the WAN to bridge different transport protocols (e.g., between a hyper-text transport and a Java messaging service). It can route messages based on their content at wire speed, provide QoS based on the content of the message being routed, and deliver a message to many interested endpoints (i.e., publish/subscribe on a per-message basis based on message content). It can also perform wire-speed message format transformations inside the network.

The application-aware network allows endpoint applications to dynamically alter the message routing rules so that the enterprise users retain control of message routing. Similar to the underlying IP/MPLS network, the application-aware network also provides VPN so that messages are only routed to entitled endpoints.

Fostering Enterprise and Service Provider Partnerships

Service provider enterprise customers can benefit from this move to the network. As the trend towards outsourcing parts of the data center grows, more and more enterprise CIOs are seeking trusted third-party hosts. A prime example of the success of the outsourcing trend is salesforce.com, which has been providing outsourced Customer Relationship Management (CRM) solutions since 1999. Salesforce.com recognized that most CRM requirements are the same from company to company. For those companies, it makes little sense to spend years and millions of

dollars building a custom system with SAP or Siebel. A standard set of services, implemented with minimal configuration, saves them time and money. This same business model is a key opportunity for service providers to help companies outsource expensive infrastructure that doesn't in itself generate revenue and focus on their core business requirements.

Beyond Connectivity - The New Service Revolution

While the promise of new revenue streams is highly attractive, service providers know there are technical challenges that must be solved to deliver new services to the supply chain, business monitoring, and financial markets.

Supply chain services

As companies begin to implement the full potential of radio frequency identification (RFID) to track goods in real-time through the supply chain, they encounter a massive increase in data traffic. Supply chain managers need to know when their pallet of cola leaves the warehouse, when it passes key milestones in transit, and when it arrives at its destination. However, enterprise networks are ill-equipped to transmit this data quickly and provide the level of monitoring required. Value-added services routing systems let service providers develop outsourced RFID monitoring solutions and track shipments at wire speed.

Business monitoring services

As networks transform from simple connections that blindly carry data to intelligent networks that understand the content flowing through them, new services that monitor and report on enterprise business activity become possible. Customers can define business rules that represent intended behavior and the network can track actual traffic against these metrics. This kind of monitoring technology lets service level agreements (SLAs) move from simple uptime and bit-flow guarantees to application-level SLAs assuring high-quality behavior of individual transactions. It can also give enterprises new opportunities to offload the burden of corporate regulatory compliance initiatives to service providers.

Financial services

A core area of focus in the financial services market (including banks, brokerages, and investment firms) is reducing information latency. Currently market data is sent from the trading floors to aggregators like Reuters, which then distribute it to the subscribers. However, this middleman system is not only expensive, it delays delivery. Although the lag is a second at most, a financial services company relying on increasingly popular algorithmic trading software needs to know that shares in Google or Exxon have changed before its competitor does. The lag can mean the difference between making or losing millions. With value-added services routing systems, service providers can provide a direct wire-speed link among trading floors around the world. They can deliver traffic, including these algorithmic trading applications, between trading offices or to remote locations. Not only does this method of distribution reduce latency times, it's more efficient. Unlike traditional direct feeds, the solution is completely interest-driven: only the data streams that traders or applications have requested flow across the network.

These services are just a snapshot of the possibilities presented by moving messaging onto service provider's networks. Since value-added service routers can identify and deliver *any* kind of content intelligently to interested parties at wire speed, the potential is infinite.

Moving Forward – Making Application-Aware Networking a Reality

Today WANs lack awareness about the content of the information they transport and how to route information to applications and users more intelligently. The next step in the evolution of the network is to build additional intelligence into the network with hardware that is aware of the content and nature of the traffic moving through it. As this network evolution occurs, enterprises will benefit by being able to leverage the network's application intelligence to build more scalable and agile business solutions.

The end result of this network intelligence will be to blur the line between the enterprise and the service provider, or the data center and the outsourcer, to a point where it's almost invisible. Services like e-mail are already offered in a virtual outsourced way, but the move towards Services Oriented Architecture in all sectors will increase this location transparency and undoubtedly improve the functionality of the applications we depend on and enjoy today. However, a traditional software approach is simply unable to provide the functionality that is required.

The Bottom Line

So what is the business value of this latest transformation of the network? The answer lies in more than just speed and efficiency. An application-aware network will deliver messages that execute actual business events with incredibly low-latency, the importance of which cannot be underestimated. It also brings the promise of distributed network computing and Web Services to life. Enterprise IT departments will be able to compose new applications from their existing software assets and extend those applications across the WAN to employees, suppliers, and customers around the world. Application-aware networks (and shared Web Services) allow IT departments to skip complex development and integration processes and literally roll out new applications. They also let network managers implement uniform policies across the network, reducing risk, improving security, and eliminating redundant systems and processes. Simplifying network complexity and building fewer larger systems also reduces operating and capital costs, and this, in turn, frees up precious staff resources for priority projects and longer-term planning. Best of all, CIOs and CTOs will be able to respond to market pressures, instead of development schedules, making them more nimble. This latest network evolution makes the mandate "do more with less" truly achievable.

Author Bio

As Solace Systems' chief technology officer, Jonathan Bosloy has invested significant effort in meeting with leading service providers and enterprises to ensure that the company's market-leading solutions tightly match customer requirements. Jonathan has also been responsible for spearheading the rapidly expanding portfolio of patents covering the company's intellectual property. Prior to Solace Systems, he was director of systems at Ceyba, a developer of optical networking equipment, and held senior engineering roles on ATM and IP products at Newbridge Networks. He was graduated from Carleton University with a bachelor's degree in systems and computer engineering.

jonathan.bosloy@solacesystems.com

MAKE YOUR SOA SECURITY MOVES WISELY...

XWALL
WEB SERVICES FIREWALL

XRAY
WEB SERVICES DIAGNOSTICS

VULCON
VULNERABILITY CONTAINMENT SERVICE

SENTRY
SOA SECURITY GATEWAY

PUTTING TOGETHER THE PIECES FOR THE WORLD'S MOST DEMANDING SOA SECURITY SYSTEMS

FORUM SYSTEMS ENTERPRISE SOA SECURITY SOLUTIONS:

- TRUSTED SOA MIDDLEWARE
- WEB SERVICES SECURITY
- XML ACCELERATION

WWW.FORUMSYSTEMS.COM

FORUMSYSTEMS

THE LEADER IN WEB SERVICES & SOA SECURITY

Ensure Interoperability.

Validate functionality.

Eliminate security vulnerabilities.

Test performance & scalability.

Confirm compliance.

Collaborate and reuse.

Ensure Secure, Reliable Compliant Web Services

PARASOFT®

SOAtest™

As enterprises adopt Service Oriented Architectures (SOA) to deliver business critical data, ensuring the functionality and performance of Web services becomes crucial. Complex web services implementations require the means to thoroughly validate and test them to assure they are truly production ready.

Parasoft SOAtest is a comprehensive, automated tool suite for testing web services and complex Service Oriented Architecture (SOA) solutions to ensure they meet the reliability, security and performance demands of your business. SOAtest provides a total and holistic testing strategy for your SOA implementations including automated unit testing, graphical scenario testing, scriptless performance/load testing, security penetration testing, standards validation, message authentication, and more.

If you are building serious web services, you need SOAtest. For more information regarding Parasoft SOAtest, call 888-305-0041 (x-3501).

Download a copy of SOAtest for a free evaluation today at www.parasoft.com/WSJmagazine

Parasoft SOAtest clients include: Yahoo!, Sabre Holdings, Lexis Nexis, IBM, Cisco & more.

PARASOFT®

We make software work.™

Automated Error Prevention™