

SOA Web Services JOURNAL

SEPTEMBER 2006 / VOLUME: 6 ISSUE 9

LOAD TESTING

Enterprise Data Integration
Business Boon or
Budget Breaker?

Watchfire AppScan
Assessing Security
Profiles

Security
SOA Access Control
Policy Management

User Interface
Don't Forget
the Consumer



PLEASE DISPLAY UNTIL NOVEMBER 30, 2006

\$6.99US \$7.99CAN



09>

0 71486 03420 9



October 2-4, 2006

Santa Clara Convention Center
Hyatt Regency Silicon Valley
Santa Clara, CA

EARLY-BIRD REGISTRATION!
SEE PAGE 44 FOR
DETAILS



Give your data direction

Link up with MapForce® 2006,
and exchange data with ease.

Spied in MapForce 2006 Release 3:

- Integration with Microsoft® Visual Studio® .NET and Eclipse
- Easy modification of EDI transaction sets and messages
- More versatility for user defined data processing functions
- Numerous usability enhancements

Altova MapForce 2006, the award-winning data integration and Web services implementation tool, makes it easy to convert between XML, database, flat file, and EDI formats and to map data to WSDL operations. Simply drag connecting lines from information sources to targets and drop in data processing functions. MapForce converts content on-the-fly and also auto-generates integration code in XSLT 1.0/2.0, XQuery, Java, C++, or C# for royalty free use in your data management and Web services applications. Get connected!

Download MapForce® 2006 today: www.altova.com

MapForce is also available as part of
the acclaimed Altova XML Suite.

Visit us online at WebServices.SVS-CON.com

Inside This Issue

TESTING

20

Bijoy Majumdar, Ujval Mysore, Lipika Sahoo, and Sunny Saxena



Load Testing Web Services

Software testing is crucial to SDLC and load testing is integral to any efficient testing scheme

CASE STUDY

26

Salman Akhtar



ChoicePay: SOA Testing Challenge

SOA testing doesn't stand alone

SECURITY

28

Kevin Smith



Access Control Policy Management

Approaches, Common Pitfalls, and Best Practices

FROM THE EDITOR

Development Blues

By Sean Rhody

5

ARCHITECTURE

The Challenges of SOA

Which rules are necessary and which are just nice to have

By Dan Foody

8

DATABASES

Enterprise Data Integration

Business boon or budget breaker?

By Erin Cavanaugh

10

SECURITY

Build Management

The potentially greater number of vulnerabilities in a SOA makes it more important

By Sean Blanton

14

PRODUCT REVIEW

Watchfire AppScan

A simple and effective tool for assessing the security profile of Web Services applications

By Brian Barbash

16

ARCHITECTURE

The Optimization Appliance

A field guide to distributed processing in a Service Oriented Architecture

By Tom Yohe

32

USER INTERFACE

There's a "C" in SOA

Don't Forget the Consumer in SOA

By Gus Bjorklund

36

PRODUCT REVIEW

WorcsNet IAB Studio

Tightly integrated set of development and runtime tools

By Paul Kaiser

40

BUSINESS

The Missing 'Discovery' Link

Process improvement based on user behavior, not estimates or anecdotes

By Stuart Burris

42

ADOPTION

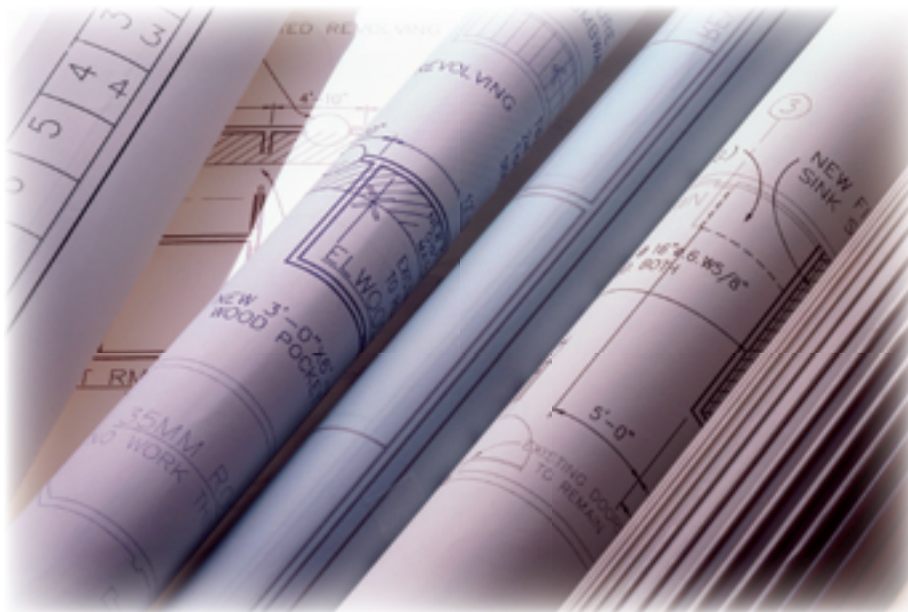
Best Practices for Building SOA Applications

Seven steps to SOA Adoption - Part Two

By Dave Shaffer

48

BPEL is the SQL of SOA



**Get started building next-generation
SOA applications with the leading vendor of
BPEL technologies**

Download BPEL tooling & server software today

activebpel.org/soa

***active*BPEL**

**BPEL consulting, certification and training.
BPEL design tools, servers and source code for Eclipse, Apache Tomcat, JBoss,
WebSphere, WebLogic, BizTalk and Microsoft .NET.**

Copyright 2006 Active Endpoints, Inc. All Rights Reserved.
All product names are trademarks or service marks of their respective companies.

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

Associate Editor

Lauren Genovesi laureng@sys-con.com

PRODUCTION

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
Mandy Eckman mandy@sys-con.com

WRITERS IN THIS ISSUE

Raghu Anantharangachar, Riad Assir, Brian Barbash, Stuart Burris, James
Caple, Mohit Chawla, Daniela Florescu, Manivannan Gopalan, David
Linthicum, Tieu Luu, Sandeep Maripuri, Frank Martinez, Robert Morris,
Sean Rhody,

EDITORIAL OFFICES

SYS-CON MEDIA

577 CHESTNUT RIDGE ROAD, WOODCLIFF LAKE, NJ 07677

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

Woodcliff Lake, NJ 07677 and additional mailing offices

POSTMASTER: Send address changes to:

WEB SERVICES JOURNAL, SYS-CON Publications, Inc.

477 Chestnut Ridge Road, Woodcliff Lake, NJ 07677

©COPYRIGHT

Copyright © 2006 by SYS-CON Publications, Inc. All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy or any information storage and retrieval system without written permission. For promotional reprints, contact reprint coordinator. SYS-CON Publications, Inc., reserves the right to revise, republish, and authorize its readers to use the articles submitted for publication. All brand and product names used on these pages are trade names, service marks, or trademarks of their respective companies. SYS-CON Publications, Inc., is not affiliated with the companies or products covered in Web Services Journal.



www.SYS-CON.com

Development Blues



WRITTEN BY SEAN RHODY

Nothing is more enlightening for a technologist than to observe development in progress. We're faced constantly with a bewildering array of choices and tools. We see specifications on paper that then become something completely different when we actually get to see them implemented in actual software that we then configure to meet our needs, or at least we hope.

I've been spending some time working with a team doing an SOA proof-of-concept test and it's reminded me of what an open book the world of SOA is, and how few pages have really been written in it. The migration of ideas to specifications, and then their transformation within software is a strange process.

Given that SOA has so many optional parts, it's not hard to understand how difficult it is for a vendor to put together a product that actually guides developers in the development process. In what may be the biggest irony of SOA, the technology that we use to enable interoperability is really a set of standalone software, distinct and separate from one another.

If you think about it, there is a logical progression of development for SOA, but because so much of SOA is about enabling communications with existing software rather than creating new services from scratch, there is no one typical development path. This is unfortunate, because the current situation is very similar to a least-common denominator approach, one where each aspect of development is distinct and isolated. You have one console for creating UDDI registry entries, another tool for creating WSDL and other documents, yet another tool for the actual coding of a service, and still another, different place for defining security entitlements. None of which are aware of one another. This makes development a fragmented, disjointed process.

Some may argue that it has to be this way for a toolset to support the broadest range of capabilities. I would agree, but I also think it's possible to create an SOA-focused development tool in the same way that folks like Borland created a Java editor that understood the environments in which it was used. In the same way that code editors today can understand the differences between BEA WebLogic and IBM WebSphere, there is a need for a development environment that understands the various standards as well as the concrete implementations of those standards and how to interface with them to make a development process seamless.

I am well aware this is not as trivial as it sounds. Just keeping an environment in synch with the various levels of specifications is not trivial. Supporting the latest is never enough – think about what would happen if the actual deployment environment is behind in revisions and needs a previous version. Now add to that differing implementations of standards by various vendors and you can begin to imagine the scope and depth of this problem. A good number of vendors have shied away from even contemplating a solution to the issue, preferring to believe there is no solution.

That's a problem, and an opportunity. SOA is too complex to be implemented piecemeal by cobbling together a set of tools. There is a strong need for a product to manage the complexity and variety of the process in a structured fashion. While XML editors such as XML Spy are very good at what they do, what's really needed is a more structured approach to creating services that removes the need to edit XML at all in favor of a more integrated approach that allows the developer to see into the whole process. Simple services are easy enough, but once we start to build complex, composite services that use things like WS Transactions or WS Orchestration, there needs to be a holistic view of the entire process, including the documents and descriptions that go along with service deployment.

This issue focuses on development tools, techniques and practices. We'll show you how to do SOA now, and let you think about how it should be done better in the future. ■

About the Author

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



IBM®





WebSphere®

_INFRASTRUCTURE LOG

_DAY 18: Came to work and found everything frozen. Icicles are everywhere. It's our processes. They're inflexible. Hard coded so we can't respond to change.

_Why did we lock ourselves in like this? Brrrr.

_DAY 19: A way out. IBM WebSphere middleware for Business Process Management. It lets us streamline business tasks and optimize performance. We can simulate and test our processes so we understand the impact they'll have, then monitor performance once they're deployed. And because it's based on a service oriented architecture, it's easy to reuse and connect existing process-based services.

_Everything's unfrozen now. Wow, it's good to feel my toes again.

Take the BPM with SOA Assessment at:
IBM.COM/TAKEBACKCONTROL/PROCESS

The Challenges of SOA

Which rules are necessary and which are just nice to have

WRITTEN BY DAN FOODY

➤ “Our processes are bulletproof. Nothing gets into production that doesn’t go through the proper and complete approval process.” Famous last words uttered by far too many enterprise architects. Some of them actually believe it’s true – others think that by hoping it’s true, maybe, just maybe, they can make it true.

The reality, as any line-of-business developer can attest, is much less clear-cut. The challenge is that governance only gets harder the more an organization moves towards a service-based architecture.

One of the first myths that drives a number of enterprise architecture governance decisions is that adding more rules reduces risk. That may be true in theory, but in practice it actually increases risk. The reason is simple: complexity increases risk. A perfect case study of this, one that most people have probably experienced, is password-control policies. As many IT organizations have attempted to “improve security,” they’ve done things like disallow use of dictionary words in passwords, force passwords to change often, disallow reuse of older passwords, etc. The net result is that, because of the added complexity, more people write down their password on a Post-it note. And written-down passwords increase the likelihood of a security breach while, at the same time, making it harder to detect the breach. Increased complexity increases risk.

Avoiding the Complexity Pitfall

There are two ways to address this complexity issue:

- Have fewer rules, but make them more important rules
- Automate compliance with the rules

In terms of gauging the importance of rules, I’ve seen a number of cases where architects put too much emphasis on the technical side and too little emphasis on the business side. For example, let’s look at a technical requirement: the need to promote reuse. This often leads to many rules: Rules around the use of certain schemas, security mechanisms, designing a service

interface, and many others. Reuse is no doubt important so it makes sense to have rules to promote it. But, let’s contrast this with a business requirement: regulatory compliance – whether it’s Sarbanes-Oxley (SOX), European Union privacy regulations, HIPAA, or even Visa’s Cardholder Information Security Program (CISP). These lead to a large set of rules as well. So, let’s say you had to choose between rules to promote reuse or rules to ensure regulator compliance. Would you choose the rules that have no directly quantifiable upside and, at worst, lead to increased cost and reduced agility? Or, would you choose the rules that would keep you from going to jail, getting fired, getting fined, or force your company to shut down? When put in these terms it’s easy to see which rules are the most important.

The other approach is to attempt to automate as much rule checking as possible. There are solutions that help address this at every stage of the application lifecycle. Of course, not every rule can be automated, so you still need to be mindful to tightly control and prioritize the set of rules that development must follow.

Automating Governance

For the rules that can be automated, one of the most common approaches is a deployment checkpoint. At deployment time your services are checked against a set of automated rules. These might validate that the services are WS-I-compliant, (increasing their interoperability) and follow further sets of rules that are specific to your organization. This might be that the services use specific predefined schemas, only use certain message transports, etc. The good thing is that this catches non-compliant services before they go into production. The downside is that by the time the service is



caught, it’s often too late. When it’s a choice between meeting a business deadline or following the architecture committee’s guidelines, most often the business wins.

The next aspect of automating governance rule validation is applying checks at development time. There are a number of products emerging that can validate the same sets of rules as the “deployment checkpoint” approaches, but do this as a normal and natural part of the development process itself. The advantage of these tools is that they guide the developer down the right path from day one as they build their services, so there’s no wasted effort. An added benefit of these tools is that they not only validate that the metadata (such as WSDL) is compliant with the rules, but they often validate that the content of the messages themselves is also compliant. This includes checks such as whether the messages actually match the WSDL, whether the use of the SOAP protocol is WS-I compliant, etc.

There is a major blind spot in these approaches: they can only validate what they can see. This is where the third aspect of automating governance comes in: runtime governance. There are three different kinds of blind spots in development and deployment time governance products that are addressed by the more advanced runtime governance products.

Blind Spot #1: Service Behavior

While development and deployment time approaches can validate metadata like WSDL and (in some cases) message content, what they can’t do is validate that a service behaves according to the rules. For example, does the service properly keep

an audit trail in all required cases? Does the service only allow authorized individuals to use it? These are things that can't be validated by development or deployment time governance tools. Even testing tools can't adequately validate that these kinds of rules are enforced in all the requisite cases. In many cases, when these types of rules are implemented in code, the only way to validate that they are properly enforced is by diving deeply into the code and evaluating it against a wide series of potential scenarios.

Alternately, you can take advantage of runtime governance tools. These products change governance rules related to behavior from being a coding task to a configuration task. In these products you point and click to declare auditing, security, and other policy behaviors. Moving the enforcement of these rules from a coding task to a configuration task addresses two issues: repeatability – configure these products the same way, and they behave the same way. The same can't be said about custom per-service code. Secondly, since the configuration itself is metadata, validating whether the service meets the governance rules can now be automated, eliminating or at least significantly reducing the chance of human error while simultaneously reducing the time and cost of validation.

Blind Spot #2: Process Awareness

Service Oriented Architectures dramatically change the way you need to think about your production applications. When a service goes into production, that's not the end – it's just the beginning. The reason is that every time a service is reused, it essentially becomes part of a new application – a new business process – and that business process may have an entirely new set of rules to obey. For instance, a service that's used to store an audit log of information. When the service goes into production you might apply a certain set of governance rules to it – checking those at development and deployment time. Let's say another service – part of a new application – now starts using the audit-log service to store order information as part of an ordering process and that order information includes credit card data. In this case, the service would now be subject to Visa CISP rules *even though the service wasn't changed and wasn't redeployed*. The only thing that changed was how the service was used, and now the set of applicable governance rules changes.

The net result is that you can't assume that development and deployment time governance checks on a service are enough. This is another role where runtime governance comes to the rescue. The most advanced runtime governance products can apply their governance policies not only to individual services, but across entire end-to-end business processes, regardless of when the services were deployed. Since the new business process and thus the new use of the service is what is being deployed, you can validate the policies effectively at business-process deployment time. In contrast, without awareness of a new context of use, the business process, you'd be forced to re-analyze each service that's already in production the moment another application is deployed – a very complex and time-consuming challenge.

Blind Spot #3: Rogue Services

Up until now, we've gone with the assumption that the governance review process is aware of all of the services and uses of services that are going into production. But, is this a realistic assumption? It turns out that in many cases it's not. If a service or service-use gets into production and it didn't go through the proper approval process, you have what's called a rogue service. Rogue services are organizational risks because you just don't know whether they're in compliance or not. It doesn't matter how well you tried to follow your process – if a service gets into production and it's not auditing financial data (and so isn't SOX-compliant) someone might go to jail. The SEC doesn't give you amnesty because you *tried* to follow your process.

Rarely are rogue services the result of malicious acts. Most people in an organization don't try to bypass the approval process – it can happen for a lot of innocent reasons. For example, let's say you're deploying a packaged application or an application built by a third-party outsourcer – you might not be aware of all of the services contained in this application. Even when you are, sometimes there are just too many to fully evaluate – SAP, for example, has hundreds of services ready to use out-of-the-box. A second case might be a service that was built purely for internal use in an application and so wasn't subject to the approval process – but someone in another application gets a hold of it and starts using it. When you talk about rogue service use, the set of cases where this can occur grows even longer. One organiza-

—continued on page 24

CORPORATE

President and CEO

Fuat Kircaali fuat@sys-con.com

Group Publisher

Jeremy Geelan jeremy@sys-con.com

ADVERTISING

Senior VP, Sales & Marketing

Carmen Gonzalez carmen@sys-con.com

VP, Sales & Marketing

Miles Silverman miles@sys-con.com

Advertising Director

Robyn Forma robyn@sys-con.com

Advertising Manager

Megan Mussa megan@sys-con.com

Associate Sales Managers

Kerry Mealia kerry@sys-con.com

Lauren Orsi lauren@sys-con.com

SYS-CON EVENTS

Associate Event Manager

Lauren Orsi lauren@sys-con.com

CUSTOMER RELATIONS

Circulation Service Coordinator

Edna Earle Russell edna@sys-con.com

SYS-CON.COM

VP information systems

Robert Diamond robert@sys-con.com

Web Designers

Stephen Kilmurray stephen@sys-con.com

Paula Zagari paula@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 or 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

For list rental information:

Kevin Collopy: 845 731-2684, kevin.collopy@edithroman.com;

Frank Cipolla: 845 731-3832, frank.cipolla@epostdirect.com

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

Enterprise Data Integration

Business Boon or Budget Breaker?

WRITTEN BY ERIN CAVANAUGH

➤ Data is king in today's information-driven economy, which is why organizations are willing to spend tens or even hundreds of thousands of dollars on data integration frameworks and applications. These organizations understand two critical truths – they have yet to capitalize on the potential business value stored in relational databases, EDI, flat files, and XML systems. And they must seamlessly connect with customers, suppliers, and business units – all of which may store and process data in different formats – to remain competitive.

Open standards-based technologies like XML promise to unify enterprise data and enable advanced Web Services and SOA. But while XML may be standards-based, most existing enterprise data isn't, nor is it easily extensible. Complicating matters further is the fact that many large enterprises rely on EDI systems to exchange business information with their partners. EDI is generally not interoperable with other systems.

XML gives organizations the ability to leverage existing systems and increase their usefulness by adding the flexibility required for real-time data exchange. Furthermore, it can facilitate the exchange across departmental and geographical boundaries and through system and programmatic constraints. But XML is not, in and of itself, a cure-all for data integration. Successfully integrating XML with other data formats requires applications that integrate system interfaces and map between data structures.

No Two Data Formats Are Alike

There are various formats for storing and exchanging data in use today and the fact that no two are alike add to the challenges of information accessibility and data integration. Let's take a look at the most popular formats and what makes each of them unique.

Relational databases

This is the dominant storage mechanism for structured enterprise data, an efficient means of storing, searching for, and retrieving information from large collections of data. Relational databases specialize in relating individual data records grouped by type

in tables. Records can be joined together as needed using SQL and presented to business users as meaningful information.

The technology is mature, and so the sheer volume of information stored in relational databases and the number of hours invested developing structures and specialized systems make them valuable assets. But their flexibility remains limited when it comes to integrating with other systems. Also, the differences between major commercial implementations can make data integration difficult.

Electronic data interchange (EDI)

Long before the Internet made business-to-business electronic trade a standard practice, there was EDI. This set of widely used formats allows for the electronic exchange of information, and was developed to enable independent organizations to reliably exchange various types of data, including purchase orders, invoices, shipping notices, medical and insurance claims, and the like.

EDI has proven valuable for supplanting paper-based business processes. It has also enabled organizations to exchange large amounts of information with partners and other companies quickly in a fairly standardized interaction.

Many larger organizations have substantial investments in EDI technology. But smaller companies have been less likely to invest in EDI technology partly because the implementations are infrastructure-, training-, and maintenance-intensive. To partner with larger enterprises, however, these smaller companies must find a way to handle EDI-based business transactions otherwise they risk missing lucrative business opportunities.



Flat files

Many legacy enterprise systems and popular applications, including accounting, banking, CRM, and spreadsheet software, support flat-file formats. They are frequently used as an interchange format for transferring information between applications, including databases. However, flat files generally require additional processing to interoperate with common data formats such as EDI or XML and can be cumbersome when dealing with large amounts of information.

XML

In the world of information interchange, the use of XML has grown steadily, and it now plays a central role in data management, transformation, and exchange. It has gotten widespread support from leading software, server, and database vendors, and has become the language of choice for lowering the cost of processing, searching, exchanging, and reusing data and information.

The openness of XML allows it to be exchanged between virtually any hardware, software, or operating system, and allows for information interchange without restriction. XML and XML-based technologies such as XML Schema, XSL, WSDL, and SOAP are all open standards that can be used in conjunction with any programming language or platform. Thus, XML technologies can be used on and between virtually any combination of database, application runtime, and operating system – a characteristic that's essential for integration with heterogeneous systems.

Fiorano SOA™ 2006

The Quickest path to an SOA

- ✕ FioranoMQ™ 2006 – world's fastest, most scalable JMS
- ✕ Fiorano ESB™ 2006 – CAD/CAM for distributed applications
- ✕ Fiorano BPEL Server – simplifying business process orchestration
- ✕ Fiorano Tools – BPEL Studio, Mapper, FEPO, etc
- ✕ Fiorano Components – 60+ pre-built adapters



Benefits

- ✕ Adherence to popular industry standards - JMS, COM, .NET, JCA, JMX, BPEL, SOAP, etc.
- ✕ Multi-language, Multi-platform, Multi-protocol
- ✕ Unmatched Scalability and High Performance
- ✕ Quick, Measurable ROI

Download your copy of Fiorano today!

www.fiorano.com/downloadsoa

Fiorano[®]
Enabling Change at the speed of thought

Developing Data Integration Applications

Data integration applications and frameworks offer the potential to unify business data while capitalizing on the particular strengths of relational databases, EDI, flat files, and XML systems, and there are several approaches an organization can take to developing these solutions – each with its own advantages and disadvantages.

Middleware or server-based platforms, for example, tend to be proprietary, closed solutions that are extremely expensive to purchase, implement, and maintain. For some the more viable option is to build customized data integration applications that are flexible enough to adapt to changes and don't force businesses to lock into a particular system vendor. Yet despite the advantage of flexibility, customized data integration applications are often extremely complex, expensive, and time-consuming to develop.

Alternative, More Cost-Effective Approaches

Complicated ESB or EAI installations, expensive server deployments, hundreds of hours of programming, consulting and training – clearly the cost and complexity

of an effective data integration application can add up, and fast. But there are alternatives that can simplify the development of data integration applications and meet the needs of individual integration challenges.

When timeliness and budget make byzantine enterprise solutions impractical, data integration development tools like Altova MapForce can be used to build customized applications very quickly and at a fraction of the cost. Such tools are easy to use and sell for under \$1,000 – far less than an ESB or EAI solution. Most companies with specific data integration requirements will find using these tools a more manageable and cost-effective investment.

If a data integration tool is the right choice for your organization, there are certain characteristics you should look for to be certain the tool is up to the task.

Visual data integration and mapping

The advantages of a visual data integration tool can't be overstated. Because the tool reads and writes all the native file formats, from relational databases and EDI to flat files and XML, one well-rounded programmer is all it takes to accomplish what otherwise would require a specialized team of experts. A visual interface lets the developer design a data map-

ping without having to understand the specific details of how to programmatically access the data formats that are being integrated.

Multiple sources and targets

Mapping shouldn't be limited to one-to-one relationships. Look for a tool that lets developers mix multiple sources and multiple targets to map any combination of different data sources and targets in a mixed environment.

A data integration tool should also provide a comprehensive library of advanced data processing functions, and let developers specify mappings based on conditions – Boolean logic, string operations, mathematical computations, and so on. It should also let developers save complex functions for use at other stages of data processing to save time and effort.

Automatic generation of royalty-free code

The ability to auto-generate code in various languages (Java, C#, C++, XSLT, XQuery) means you get reliable code faster and with less effort. Instead developers can focus on the all-important business logic of the application while leaving the generation of low-level infrastructure code to the tool. Make sure the code that the tool generates can be used royalty-free and doesn't require any proprietary deployment adapters.

Furthermore, some tools, such as Altova MapForce, also have the capability to process transformations internally, letting you preview the output of your mapping and ensure accuracy before generating code. This feature is also useful for doing periodic or light-duty integration tasks on-the-fly.

A Way To Achieve Data Integration Today

The value of an organization's business information is directly proportional to its ability to share the information internally and externally, which is why organizations can no longer afford to let data storage and exchange systems operate in a vacuum.

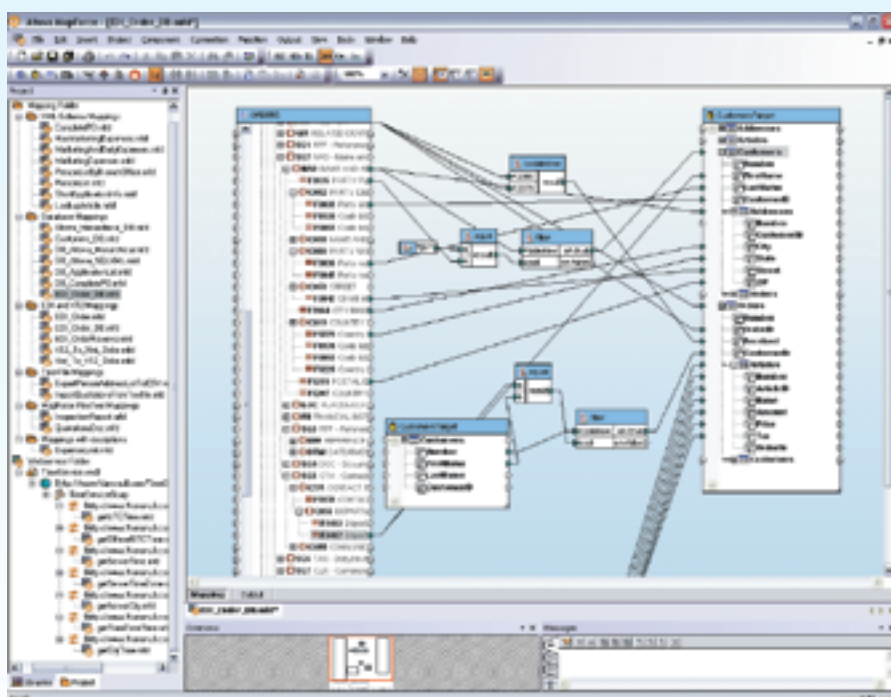
Fortunately, the arrival of simple cost-effective data integration and development tools means that sharing data with customers, partners, and business units doesn't have to cost companies a small fortune for a largely unnecessary enterprise solution. ■

About the Author

Erin Cavanaugh is product marketing manager for Altova (www.altova.com), creator of XMLSpy and other leading XML, data management, UML, and Web services tools. In this role, Erin manages Altova's XML-related line of tools. She has held product marketing, training, and technical copywriting roles at a variety of hardware and software firms.

erin.cavanaugh@altova.com

Data Mapping Can Be A Treasure



Visual data mapping tools with code-generation capabilities can simplify and accelerate the development of data integration applications and make information leverage a reality. MapForce from Altova lets you map any combination of XML, database, flat files, and EDI data. It supports all major relational databases, provides a library of data processing functions, and generates XSLT 1.0/2.0, XQuery, Java, C++, and C# code for royalty-free use in your custom data integration applications.



Bring your development plans to light

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

Revealed in XMLSpy 2006 Release 3:

- Superior error messaging with dynamic hyperlinking
- New XSLT 2.0 and XQuery profilers
- Trace points for enhanced XSLT debugging
- Innovative restriction handling in XML Schema design

Altova® XMLSpy, the industry standard XML development environment, is indispensable for modeling, editing, transforming, and debugging XML-related technologies. Illuminate your strategy with the world's leading XML editor, the original graphical schema designer, a code generator, file converters, debuggers, profilers, support for XSLT, XQuery, WSDL, SOAP, and a wealth of brilliant XML utilities and enlightened usability aides.

Become a markup mastermind!

Download XMLSpy® 2006

today: www.altova.com

Build Management Is Critical to Developing an SOA Enterprise

The potentially greater number of vulnerabilities in a SOA makes it more important

WRITTEN BY SEAN BLANTON

➤ Developing under a Service Oriented Architecture (SOA) is different from traditional development. A large set of business changes will now be funneled through a relatively small number of enterprise services. An inefficient or bad build system can impact a greater number of business changes. As services are exposed to more consumers and so to more potential threats having a robust and secure development environment is more important than ever. Centralized role-based control of builds and reporting of build activities is critical for incorporating a greater number of changes and managing the security and auditability of Web Services.

Build management is a critical operation for transitioning to an enterprise application environment modeled on SOA. Achieving a repeatable application lifecycle that is efficient and secure is strongly influenced by how well the build management system works. Instead of different client applications all connecting to different data sources to retrieve whatever bits of information they need, client or consumer apps under SOA will simply ask for a customer record from a service designed to provide a full customer record and nothing else.

The Service Model

Different consumer applications that need customer information will all request that information from the same service and that promotes great reusability. In practice, the number of these kinds of enterprise business services-facing consumer apps will be relatively small. And, compared to a single traditional application, developing these enterprise services results in a larger number of changes being made to the smaller number of services. Incorporating changes into the services is where build management comes in, and incorporating larger numbers of changes puts increasing demands on the build system.

The build process takes the application source code and translates it (compiles it) into a form suitable for running on a machine or application server. The build for an application typically involves several hundred source and resource files and dizzyingly complex dependency relationships that require certain steps in the translation process to occur in a certain order. The goals of a successful build process are that it be, in order, repeatable, auditable, standardized across applications, automated, highly performing (usually meaning as fast as possible) and able to produce reports and metrics for organizational performance improvement. A successful build process is highly dependent on good version control and is part of good overall software configuration management.

Loose Coupling

Another goal of a SOA is loose coupling between the consumer applications and the service. This means that the consumers are developed relatively independently from the services. This should drive down operational costs by creating smaller, more manageable application components. While it's everyone's dream to work independently and not be bothered by what others are do-



ing, in an enterprise environment nothing happens in a vacuum and there always has to be coordination between teams at some level. This is very different from the situation of a service exposed over the Internet for anonymous public applications to consume. This shows that implementing an SOA in an enterprise environment will be very different from other types of implementation.

Loose coupling really boils down to not having a direct build dependency between the consumer and the service. This could allow (in theory) each consumer to merrily follow its own release schedule, independent of changes being made to the service. The reality is the consumer still has to know how to talk to the service. In a Web Services implementation of SOA, there's a WSDL file (Web Services Descriptor Language) that tells the consumer how to talk with the service without having a direct build dependency.

Considerations of performance and security have meant that some organizations have not been able to achieve loose coupling with their chosen technologies. This means that the build dependencies can be as complex as traditional apps or even more so. Changing the service could mean that every consumer app that uses it

has to be rebuilt. This has an expensive impact on consumer applications' testing and release schedules, introducing risk to the production environment. As the technologies supporting SOA mature, it will be easier for organizations to move to loose coupling and simplify the enterprise build dependencies and other organizational benefits.

Security Concerns

There are a few general security concerns particular to SOA. By virtue of being a new methodology built on new technologies, security holes can easily arise in one of the technology products. One major industry analyst states that 70% of security holes addressed in enterprise environments will be reintroduced when moving to a services-based infrastructure.

One particular concern is that loosely coupling consumers and services means that tight, secure communication between the two can be a challenge. Another is that since services are exposed internally, anyone can theoretically write a consumer for that service in some implementations of SOA. For mission-critical business applications, you don't want just anyone accessing your customer record service however. A big investment in securing a SOA is required not only to thwart malicious use of the services, but to prevent an unintended impact of a legitimate consumer using a service in an innocent but inappropriate way.

Best-Practice Build Management

Before we can embark on a discussion of build and security issues of a SOA we need to cover the fundamentals that make up a good build management system. This was the stuff of articles of four or five years back, so if you're not covered here, your organization is not as competitive as it could be.

As mentioned, version control is a critical component of good build management and having a centralized source code repository as part of a version control or full software configuration management toolset is a fundamental requirement. You have to be able to control and identify precisely the application code going into the build. Centralization provides the standardization that reduces the overall number of processes and procedures and allows automation. Besides application code, you also have to manage the versions of third-party library sets that, say, provide database, messaging,



| Build Number | Build Step | Status | Result | Elapsed Time |
|--------------|--|-----------|-------------|--------------|
| 1 | Update from Perforce | Completed | Successful | 0h 0m 3s |
| 1 | Make Build Control File | Completed | Failed | 0h 0m 2s |
| 1 | CRT Dependency Analysis | Not Run | Dep In E... | |
| 1 | Build Job complete for pr-hotel | Completed | Failed | 0h 0m 1s |
| 1 | on complete for pr-hotel | Completed | Failed | 0h 0m 2s |
| 1 | BCD generation for hotel dev | Completed | Failed | 0h 0m 1s |
| 1 | Build Job complete for hotel dev | Completed | Failed | 0h 0m 1s |
| 2 | Update from Perforce | Completed | Successful | 0h 0m 1s |
| 2 | Make build control file | Completed | Successful | 0h 0m 3s |
| 2 | CRT Dependency Analysis | Completed | Successful | 0h 0m 2s |
| 2 | Build Job complete for pr-hotel | Completed | Successful | 0h 0m 2s |
| 2 | Set classpath of hotel classpath for hotel war | Completed | Successful | 0h 0m 1s |
| 2 | Ant Javac of hotel javac for hotel war | Completed | Successful | 0h 0m 3s |
| 2 | Ant War of hotel war for hotel war | Completed | Successful | 0h 0m 3s |
| 2 | on complete for pr-hotel | Completed | Successful | 0h 0m 2s |

and logging functionality.

It's vitally important to separate actively changing code from base-line versions identified by the version control system. The best way to do this is to have an isolated build environment with limited access where the only application code introduced into the environment comes from the version control system and isn't copied over by a person or some external process. This allows tight auditing of the build and you should be able to identify the exact version of every file used in the build and trace it through the version control system to the person, time, and date of the change.

Having a build coordinator responsible for maintaining the security and integrity of the build is a widely accepted best practice. The developers' responsibility is to get coding changes out the door to the test and production environments as quickly as possible and this can be at odds with best-practice build management that provides carefully controlled conditions and precise library use. The challenge is to provide a build system and process that can serve both developer and corporate needs.

Build Considerations for a Secure SOA Environment

Library control in builds is more important than ever. Using the wrong version of a third-party library set can mean incorporating a published security vulnerability in your enterprise application. Because of the exposure of services to multiple consumers, vulnerabilities that previously had no impact in a traditional application may suddenly become important for an enterprise service.

Capitalizing on best-of-breed build management tools can really be a cost-effective way to extend your build management capabilities into detailed reporting and monitoring. These capabilities will help create a more agile and controlled build

environment as demanded by the challenges of creating a service-oriented infrastructure. It's important to distinguish between tools that help you manage your existing build system (Cruise Control, AntHill, BuildForge) from tools that only control the translation process (ant, maven), and tools that can do both (Openmake, Serena Change Man Builder).

Detailed dependency reporting and analysis helps developers track down bugs, validates secure development, and provides a watertight audit trail. Dependency reporting

should include not only the file use in the build, but associate those files with specific versions of code in the source code repository that can be tied to specific business change requests.

Builds can be done "continuous integration"-style and scheduled for hourly builds or triggered by source code repository change, or simply done on-demand as the situation requires. Branching the code base could put further demands on the build system. Make sure you can handle this since developing enterprise services can involve more branching so consumers can continue to use an existing service while a new version is developed at the same time.

Summary

Having a secure build environment with tight control over the library sets used in the builds and having a full audit trail of what code was used in the build is an important part of secure development. The potentially greater number of vulnerabilities in a SOA makes it more important and starting with a tight build control system will help you avoid vulnerabilities that may be reintroduced. In addition, capitalizing on the state-of-the-art build management tools that are part of a top-line build management system introduces new capabilities for a more agile development environment while maintaining tight, secure control over code use. ■

About the Author

Sean Blanton has worked for Catalyst Systems Corporation for over eight years, has been an onsite configuration management consultant, and has taken knowledge learned from customers to help develop, support, and sell Openmake build management software. In the course of his work on the services side of Catalyst he has helped over 100 companies manage software change processes and builds (particularly). One summer Sean also programmed at the Max Planck Institut in Hamburg. He has a PhD in physics from the University of Chicago and a BS from Columbia University.

Watchfire AppScan

A simple and effective tool for assessing the security profile of Web Services applications

WRITTEN BY BRIAN BARBASH

➤ Security is a major component of application development and must be tailored to the environment and audience of the system. In many respects, the more widely available an application is, the more important security becomes. Properly testing and securing Web Services applications is a challenging task. A tool that facilitates this process and provides visibility into application vulnerabilities is the AppScan product from Watchfire.

AppScan is an application testing tool that performs security scans on Web applications and Web Services applications. In support of Web applications, AppScan can test server-side functions and vulnerabilities by interacting with the application in a client capacity. It also provides support for applications containing Flash and/or JavaScript, AppScan has the capacity to parse these components to navigate the application properly. When interacting with Web Services, AppScan acts as a SOAP client and provides tools for developers to manipulate inputs and evaluate those results. For the purposes of this review, the focus will be on AppScan's Web Services capabilities.

AppScan Approach

Application vulnerabilities are discovered using a three-phased approach: Explore, Analyze and Test. During the Explore phase, AppScan will interact with the web service like an end user (or SOAP client) by sending SOAP web services requests and receiving responses. Responses that indicate the presence of a potential vulnerability are logged for use during the Test phase. AppScan also submits multiple invalid requests to catalog the error responses. These responses are referenced during test validation.

In the Test phase, AppScan submits several requests to an application based on the results of the Explore phase. It applies a series of validation rules to the responses of each test to identify any potential security risks and rank the severity of those identified.

Finally, the Scan phase executes. From a process standpoint, the Scan phase will be based on the Explore and Analysis phases. Results from the Test phase typically supply additional application links that may be probed for security risks. The number of Scan iterations is user-configurable in AppScan.

Creating & Executing Tests

To test Web Services, AppScan must first parse the WSDL file associated with the application in question. Three sets of information are required to test Web Services:

1. The location of the WSDL file along with any applicable communications parameters including additional servers, custom error pages, explore phase parameters, and communications parameters such as proxy server credentials
2. Application authentication information, which may take the form of NTLM or HTTP authentication, or a client-side certificate
3. Testing policy information that includes the types of tests to run, the number of iterative scans to process, and the handling of application parameters and cookie data if applicable



Once configured, users have the option of saving the configuration as a Template. Templates can then be reused for future scans, useful for establishing standard testing scenarios across a corporate environment.

With the WSDL file parsed, AppScan presents the user with an Explorer-like view of the service. Included in this interface is a component to call the service with user-specified parameters. This allows unit test cases to be incorporated into the process. For each value entered and submitted to the application, AppScan records the values for use during the Test phase.

Once the configuration of the Web Service is complete, AppScan begins the process of evaluating the application. The time required to analyze the application will vary based on the complexity of the system. Using the sample application provided, AppScan completed the process in approximately five minutes. The results of the test are shown in Figure 1.

AppScan classifies its findings into



The Art of Web Services Testing



Build Your Applications on 4 Pillars of SOA Testing

- I. Automated Functional Regression
- II. Performance Profiles
- III. Interoperability Compliance
- IV. Vulnerability Risk Posture

Download SOAPSonar Enterprise Edition
<http://www.crosschecknet.com>

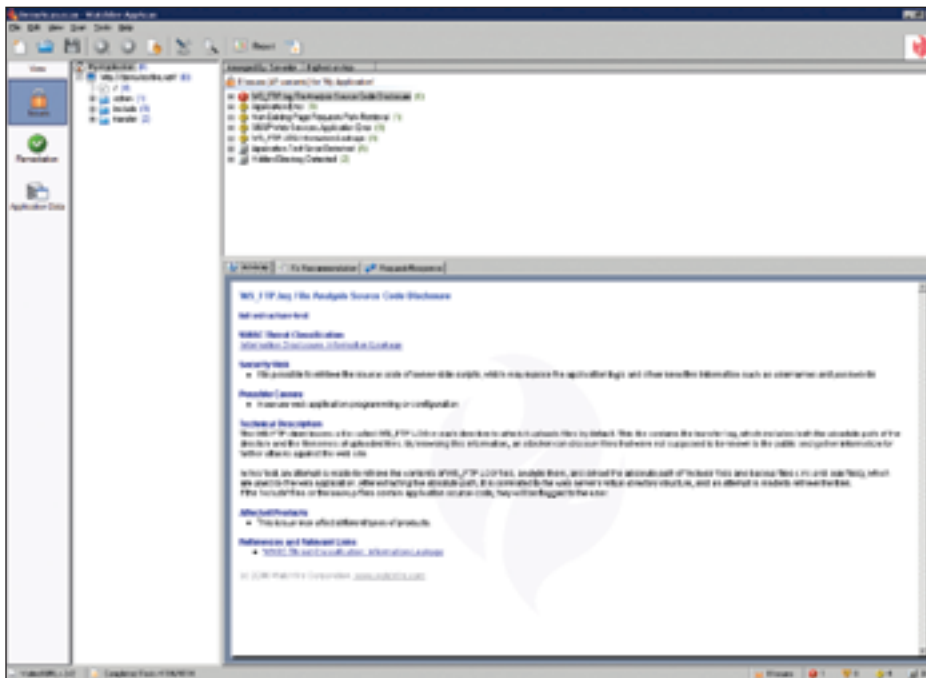


Figure 1: AppScan analysis results

high-, medium-, low-, and informational-severity levels. Each finding is described in detail and referenced to a specific Web Application Security Consortium (WASC) threat classification. Information provided includes the URL that produced the result, a detailed description of the security risk, a recommendation for addressing the issue, and the raw request/response data.

Typically in a testing situation, there are scenarios that produce results that are expected but are still reported by AppScan as an error. To accommodate this possibility, AppScan provides the ability to mark a particular issue as a false positive. Finally, AppScan provides the capability to document the issue by adding comments and capturing a screen shot of the results page.

When using AppScan as part of this review, it quickly became evident that this tool can also be highly effective in the day-to-day development process. One possibility is to incorporate AppScan tests as part of nightly and/or milestone builds, using the results to target and resolve problem areas before they reach formal testing. In the long run, this approach can lead to a more efficient development and testing process, reduce the number of test cycles, improve the quality, and establish security as a philosophy across all segments of the development lifecycle.

Test Catalog

AppScan is packaged with a number of tests to do. The following is a list of the general categories and some example tests within the category:

- **Privacy:** Unencrypted password, GET parameter sensitivity
- **Authentication:** Bypasses or exploits for ASP.NET, Lotus Domino, JRun, Netscape, PHP, and others
- **Authorization:** Token prediction, access control bypasses, session expirations
- **Client Side:** Cross-site scripting, SOAP response splitting
- **Command Execution:** SQL injection, SSI injection, buffer overflow
- **Information Disclosure:** Directory listing, log file publication, predictable location of sensitive resources/directories
- **Logical:** E-mail parameter spoofing, non-SOAP Web Service access, Denial of Service

Upon completing testing and remediation activities, AppScan can be used to generate reports that provide profiles of the application under investigation. There are many reports available categorized as follows:

- **Security Reports:** Summarizes the vulnerabilities found during the scan along with the recommended remediation steps
- **Industry Standard Reports:** Provides an analysis of the application against stan-

dards from the Open Web Application Security Project (OWASP); SysAdmin, Audit, Network, Security (SANS) institute; and the Web Application Security Consortium (WASC)

- **Regulatory Compliance Reports:** Analyzes the application against the requirements of several regulatory regimes, some of which include HIPAA, ISO, and SOX

AppScan also provides the flexibility to create user-defined report templates to fulfill any requirements not met by the existing set.

Summary

All applications that are part of any corporation's portfolio have security implications, whether they are local to a user's desktop, private to a corporate intranet, or public-facing. Care must be taken not only to protect corporate assets, but to fulfill the regulatory requirements that govern the collection, utilization, and publication of data. Therefore, security should be a part of the entire lifecycle of application development. Watchfire's AppScan product is a simple and effective tool that can be easily incorporated into each phase of the development process, helping to identify and mitigate risks before they impose significant damage. ■

About the Author

Brian R. Barbash is the product review editor for *Web Services Journal*. He is a senior consultant and technical architect for Envision Consulting, a unit of IMS Health, providing management consulting and systems integration that focuses on contracting, pricing, and account management in the pharmaceutical industry.
bbarbash@sys-con.com

System Requirements

- > Processor: Pentium III PC, 800 MHz
- > Memory: 512 MB RAM (1 GB recommended for scanning large sites)
- > Free Disk Space: 1 GB
- > Network: 1 NIC 10/100 MBPS for network communication with configured TCP/IP (100Mbps recommended)
- > Operating System: Windows 2000 with SP2 or higher, Windows XP, Windows 2003 Enterprise Edition
- > Internet Explorer 5.5 or 6.x
- > Microsoft .Net Framework version 2.0
- > JRE 5.0 (for Watchfire HTTP Proxy only)



Sound Architecture Requires Proper Planning

WEB AGE SOLUTIONS - YOUR TRAINING PARTNER FOR SOA



In all phases of SOA migration, Web Age Solutions provides training and customized services from awareness to implementation. We support vendor specific or generic SOA training tailored to your organization's needs.



Custom training for complementary SOA technologies

XML • WEB SERVICES • WSDL • SOAP • WEBSPHERE • WEBLOGIC • JBOSS • J2EE • SPRING/HIBERNATE/STRUTS • WID/WBI/WMB

Custom training plans for virtually every job role

BUSINESS ANALYST • ADMINISTRATOR • DEVELOPER • ARCHITECT • QA/TESTER • MANAGER • EXECUTIVE • SECURITY

Consulting services for all phases of SOA migration



Add Web Age Solutions to your plan & stay ahead of the competition
www.webagesolutions.com - 877.517.6540 - soa@webagesolutions.com



Load Testing Web Services

Software testing is crucial to SDLC and load testing is integral to any efficient testing scheme

WRITTEN BY BIJOY MAJUMDAR, UJVAL MYSORE, LIPIKA SAHOO, AND SUNNY SAXENA

➤ The quality of any application is determined by the robustness and scalability of the system. It's mandatory to simulate the actual environment and test the application for preparedness. Web Services-savvy applications need a different methodology for testing in a real-world scenario. The UI-less nature of Web Services presents a significant challenge in testing such applications. The whole persona of consumer stubs with different payloads dictates the planning of Web Services load-testing schemes. This paper talks about the different aspects of load testing and areas of contention that need special attention. This will be helpful in not only building a better application but also compiling a robust, high-quality enterprise architecture.

Web Services are the natural delivery mechanism to achieve SOA. While having the potential to free enterprises from the endless cycle of vendor-specific hardware/software upgrades by ensuring interoperability, they bring in integration complexities and the overhead of maintaining compatibility with the underlying EIS applications/systems. This brings in an absolutely different perspective to testing Web Services.

Web Services applications generally use a lot of data transformation, wraparounds (wrappers), translation, and abstraction to bring about the promised interoperability and portability. Their dependence on bandwidth-heavy protocols like SOAP doesn't ensure many performance benefits when compared to legacy applications (which tend to be very tightly coupled). Parameters like response time, throughput, and CPU utilization for transactions determine the viability of a real-world business application. Extensive testing of Web Services based on these parameters brings to the fore the most common performance constraints associated with them. The test results not only indicate whether the associated benchmarks are attained, but also if the service can scale to meet demands imposed by concurrent access from multiple users, simulated or otherwise.

Web Service endpoints generally also have very high visibility. They have to service multiple clients over the network simultaneously, maintaining robustness and availability at the same time. In such a situation, performance becomes even more crucial. Thus,

the significance of proper performance testing for Web Services can't be overemphasized.

A Web Service, like any other application, can be subject to a wide range of test conditions and testing strategies. Some of them being functional testing, regression testing, performance testing, stress testing, and load testing. This paper will focus only on the load testing of Web Services. The expected behavior of a Web Service will be evaluated against various performance criteria when concurrent access by multiple clients is simulated. It becomes crucial to ensure that apart from optimizing design and implementation, Web Services have to be tested for throughput, efficiency, and response simulating real-world conditions as closely as possible. This is where load testing plays a major role. A properly designed load-testing strategy can simulate real-world load and performance scenarios with minimal hassle and cost. User loads and network conditions of varying nature can be effortlessly created and replicated. Testing can be undertaken till the output charts show a performance range considered acceptable for an application of its nature. Load-testing results can hence be taken as a strong indicator of application performance in actual business environments.

To ensure optimal testing of Web Services, the test cases have been designed keeping the following parameters in mind:

- **Size of payload:** This tests the amount or size of the incoming requests. This parameter is vital in determining the threshold of data beyond which the service behaves in unexpected ways.
- **Concurrency:** The test cases have to simulate simultaneous access of the service by multiple clients to replicate real-world conditions.
- **Latency:** It can be defined as the time from issuing a request to the service from the client and the receipt of the first response. This parameter encapsulates the performance of the service, bandwidth in the network, and other communication overheads. It's important that latency be minimized as far as possible, at least up to a tolerable point.
- **System utilization:** The net CPU and memory resources consumed by the service under varying loads in normal enterprise environments should be captured by the load-testing scheme. This helps in identifying potential bottlenecks and points out areas of improvement.

These parameters shall be discussed in detail later:

Load Testing with Reference to Web Services

Load testing of Web Services is significantly different from testing of other applications since their performance is not just attributed to how robust the underlying architecture is but also to the network overheads, underlying processing involved, and the performance of the Web server that hosts the service. The behavior of the SOAP engine also invariably adds to the architecture of service provider systems. Certain major areas of contention when evaluating the Web Service performance that will be discussed here are:

- The impact of an incremental XML payload size wrapped inside a SOAP message
- The impact of a chosen style/use during the design of a Web Service
- The serialization/de-serialization involved in processing the SOAP message
- The underlying parsing model and validation schemes chosen to process the XML payload

The results of the load testing such as response time graphs will further depict the vitality of the load testing of Web Services to ascertain their conformity prior to actual deployment to enable their wide-scale adoption without compromising their performance and scalability characteristics, enabling the enforcement of stringent operational, behavioral, and non-functional requirements that are inherent in the successful realization of any business process.

Load Testing Metrics and Parameters

The results obtained by load testing Web Services can potentially be reflected in terms of the following parameters.

- **Response time:** It's the most important parameter to reflect the quality of a Web Service. Response time is the total time it takes after the client sends a request till it gets a response. This includes the time the message remains in transit on the network, which can't be measured exclusively by any load-testing tool. So we're restricted to testing Web Services deployed on a local machine. The result will be a graph measuring the average response time against the number of virtual users.
- **Number of transactions passed/failed:** This parameter simply shows the total number of transactions passed or failed.
- **Throughput:** It's measured in bytes and represents the amount of data that the virtual users receive from the server at any given second. We can compare this graph to the response-time graph to see how the throughput affects transaction performance.
- **Load size:** The number of concurrent virtual users trying to access the Web Service at any particular instance in an interval of time.
- **CPU utilization:** The amount of CPU time used by the Web Service while processing the request.
- **Memory utilization:** The amount of memory used by the Web Service while processing the request.
- **Wait Time (Average Latency):** The time it takes from when a request is sent until the first byte is received.

Performance Bottlenecks & Areas of Contention

Web Services are simply components deployed on a server. Most of the Web Services today are exposed out of existing components such as Enterprise Java Beans. Hence, in theory, we should be able to use the existing testing mechanisms and performance-enhancing strategies. But as already discussed load testing Web Services is quite different. The performance of Web Services is influenced by a lot of factors like bottlenecks in the network, processing at intermediate nodes if any, pre-processing of the SOAP message at the SOAP

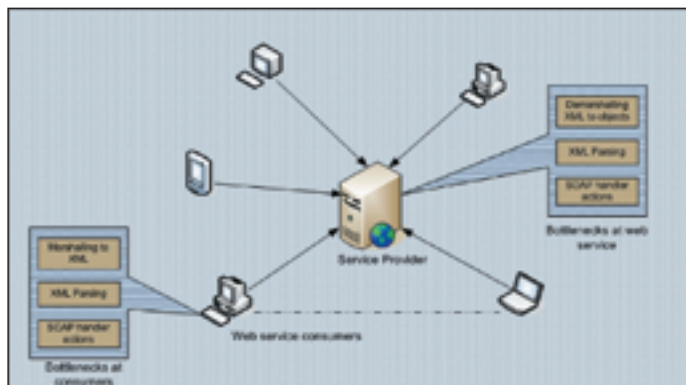


Figure 1: Performance bottlenecks

engine before it's dispatched to the service, etc. To identify the areas of contention, we'll look first at the architecture of the SOAP message processing on the service side.

A client application creates a SOAP message containing the XML payload, which can be either a SOAP-RPC-encoded request or a document-style message. The client sends this message along with the service endpoint URL to the SOAP client runtime, which in turn sends it over the network. Once the SOAP message is delivered to the SOAP runtime at the service, it passes through handlers (if any) that handle the processing of any additional tags for WS-Security, WS-Addressing, etc. Then the SOAP runtime converts the XML message into programming language-specific objects if required by the application. The Web Service processes the request message and formulates a response. The SOAP runtime on the service side takes care of creating a SOAP message and dispatching it back to the client.

So, apart from the actual processing of the Web Service, there's some additional processing involved before and after the Web Service builds a response. Let's identify the bottlenecks involved in invoking a Web Service:

- **XML processing and overheads:** Since XML data is verbose and contains lot of metadata information, processing of XML is a major performance bottleneck. Processing XML involves parsing, validating the data against schema, and marshalling/unmarshalling. It's quite memory- and CPU-intensive and the response time takes a hit if the proper strategies aren't followed.
- **Parsing of XML:** The larger the SOAP message, the longer it takes to parse it. Parsing SOAP messages is a major contributor to performance issues with Web Services. A memory-efficient parser like StAX (a pull parser) should be used in place of a memory-intensive parser like DOM.
- **Serialization/de-serialization:** When the SOAP engine on the service side receives a SOAP request, it de-serializes the XML data according to the encoding format mentioned, i.e., extracts the payload out of it, and creates objects that are used by the Web Service. After the Web Service executes the business logic, the SOAP engine takes care of serializing the response back to XML and sends the data to the client. For huge XML documents, the serialization/de-serialization takes a performance hit if a proper mechanism isn't selected.
- **Select a proper style for your Web Service:** The two most predominantly used styles of Web Services are document/literal and RPC/encoded. The SOAP message of the RPC-encoded style of Web Service contains the type-encoding information, which is an overhead on the SOAP engine and degrades the throughput performance whereas the document/literal SOAP message contains no such type-encoding information. The XML payload can be easily validated against the schema included in the WSDL. Also, the data binding specific to a SOAP engine can be switched off in the case of a document/literal-style Web Service. This enables one to use a custom binding framework like XMLBeans, castor, JAXB, etc. This is especially useful when the application uses a large number of complex custom data types.
- **Processing by handlers:** The SOAP engine first dispatches the SOAP message to the handlers. The handlers may be responsible for performing additional processing like authentication, encryption/decryption of the XML payload, parsing the SOAP message for any information like WS-Security, WS-Addressing, etc.

Case Study

To do load testing, we've created an environment similar to a real-world scenario or that emulates the scenario to a high degree. A real-world scenario will contain different payloads and a varying number of users accessing the same Web Service simultaneously.

Our test environment setup is described below:

Type of Web Service: Document/literal

| Load size or no. of concurrent users | 25 | 50 | |
|--------------------------------------|------|-------|-------|
| Payload size | 10KB | 100KB | 500KB |

A document-style Web Service has different payloads being passed on as SOAP message elements. These documents vary in size to measure the response time given by a Web Service invocation. Network congestion or the time spent in the communication pipeline distorts the Web Service's actual response time. To measure the true response time of a Web Service, the service is locally hosted, eliminating any network-related bottlenecks.

The Web Services are hosted on the JBoss application server that resides on a machine with the following setup: A Dell server PE 1600SC with an Intel 2.8GHz Xeon CPU and 1GB of RAM. The various performance parameters like response time, throughput, number of transactions passed/failed, and load size are measured against different payloads for RPC and document styles of Web Services and the results are shown on graphs.

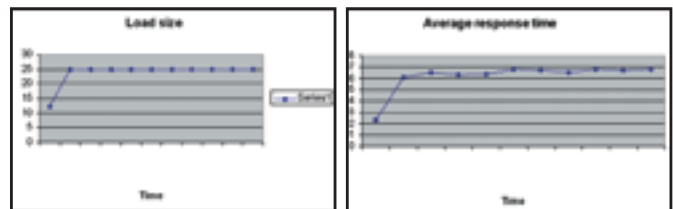
Results

Load testing summary of a document/literal-style Web Service

Payload size: 10KB

Number of concurrent users: 25

Total test duration: 10 minutes

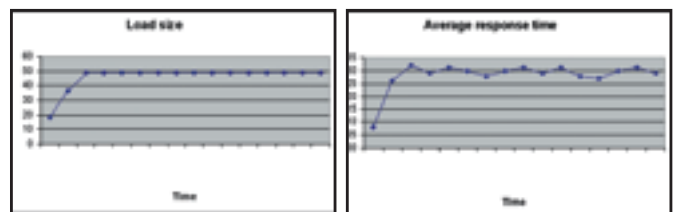


The graphs above depict the variation of the average response time of a document/literal-style Web Services at a constant payload size of 10KB. Note that the average response time is significantly nominal and the performance of the service remains stable over the given period of time.

Payload size: 100KB

Number of concurrent users: 50

Total test duration: 15 minutes



Does Your Application Change Management Process Provide You the Visibility You Need?

Don't Drive Blind

Forty percent of critical business “disruptions” are caused by application change management failures. Metalllect helps enterprises reduce risk, accelerate cycle-times and reduce costs related to application change management.

IQ Server uses advanced semantic inferencing and metamodeling to automatically map dependencies between business services and the underlying logic that execute these services, as well as the relationships within and across application logic and databases throughout the enterprise.

Whether you are changing existing applications to:

- Extend or enhance existing applications in response to changing business needs,
- Modernize and adopt SOA to increase reuse and agility while eliminating duplicative functionality, or
- Meet IT risk management and compliance initiatives

IQ Server provides you and your stakeholders with actionable insights throughout the application change management lifecycle.

For more information visit us at www.metallect.com and sign up for one of our upcoming webinars on *Adding Visibility to the Application Change Management Process*.



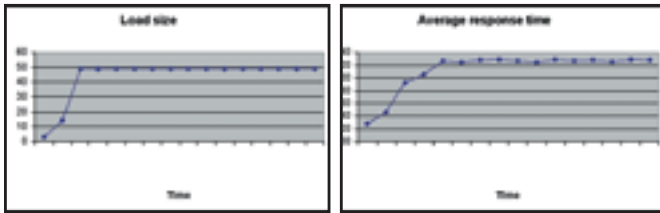
metalllect®

The same document-style Web Service when evaluated for a medium payload of 100KB performed the same as the graphs depict.

Payload size: 500KB

Number of concurrent users: 50

Total test duration: 15 minutes



When tested for a high payload of 500KB and 50 concurrent users, the document-style Web Services remain stable. The average response time remains significantly low at around 1.2 seconds. None of the transactions failed.

Load Testing Tools

There are commercial tools like Mercury's LoadRunner and Radview's Webload that are very efficient and detailed for load testing Web Services. There are various Open Source alternatives to LoadRunner that can serve our purpose of load testing to varying degrees. Some of the more popular tools include the soapUI 1.6 beta, the Grinder 3 beta, and OpenSTA. soapUI provides basic functionality to create test cases, execute them, create sample SOAP clients, etc. Grinder uses a highly detailed language called Jython to write the test scripts.

Conclusion

Software testing is a crucial phase of the SDLC and load testing is an integral part of any efficient testing scheme. This paper highlighted the importance of load testing with specific reference to Web Services. The design principles entailed attempted to bring about a proper plan for testing, the parameters to be looked for, and the expected results. The strategies contained in this paper can be implemented regardless of the platform on which the application is deployed and the tools used for testing. ■

References

- Web Services Architecture <http://www.w3.org/TR/2002/WD-ws-arch-20021114>
- soapUI 1.6 Beta <http://soapui.org/>
- Grinder 3 Beta <http://grinder.sourceforge.net/>
- OpenSTA <http://www.opensta.org/>
- Apache Jakarta's Jmeter <http://jakarta.apache.org/jmeter/index.html>
- Radview's WEBLOAD <http://www.radview.com/products/WebLOAD.asp>

About the Authors

Bijoy Majumdar is a member of the Web Services COE (Center of Excellence) of Infosys Technologies, a global IT consulting firm, and has substantial experience in publishing papers, presenting papers at conferences, and defining standards for SOA and Web Services. Prior to Infosys, he worked as an IT analyst and was a member of the GE Center of Excellence (e-center) under the E-Business Practice of Tata Consultancy Services. bijoy_majumdar@infosys.com

Ujval Mysore is a member of the Web Services COE (Center of Excellence) in SETLabs, the research arm of Infosys Technologies, a global IT consulting firm. He has substantial experience in publishing papers, presenting papers at conferences, and defining standards for SOA and Web Services. He is currently involved in the design and development of an Enterprise Service Bus. ujval_mysore@infosys.com

Lipika Sahoo currently works with the Web Services Center of Excellence in SETLabs, the technology research division at Infosys Technologies, India. Her core area of work involves dynamic adaptability and management of Web Services. She is currently involved in various research activities within the group relating to REST-based Web Services. lipika_sahoo@infosys.com

Sunny Saxena currently works with the Web Services Center of Excellence in SETLabs, the technology research division at Infosys Technologies, India. His interests range from Web Service security platforms to aspect-oriented development models. sunny_saxena@infosys.com

The Challenges of SOA (continued from page 9)

tion relayed a story of how they had built a service that had five authorized consumers (each of which had been issued a special consumer key so that the service owner could track them), but it turned out there were 34 different consumers. What happened was one of the five authorized consumers had built the use of the service into a jar file. The jar file embedded the consumer key for simplicity. Twenty-nine other project teams reused this jar file without knowing that it happened to use an external service – so they unwittingly reused the service. And these service uses didn't get approved; they were rogue service uses.

How did this organization find out about these other uses? It turned out that, to find a performance problem with their service they deployed a runtime governance product (one of the common capabilities of runtime governance products is service-level measurement) – since they thought there were only five consumers, they didn't

understand why it wasn't performing as expected. The runtime governance product they deployed could also automatically discover new services and new service consumers. This product automatically discovered all 34 consumers. By interfacing with the company's registry of approved services, the product determined that 29 of these consumers were actually rogue consumers and immediately flagged these for approval. The most advanced runtime governance products can even automatically quarantine rogue services and service uses until they're approved – eliminating the risk of rogue services.

Bringing It All Together

To implement a complete approach to SOA governance, you have to consider the roles of development, deployment, and runtime governance. Taking a holistic view of governance across the lifecycle will automate as much of the governance burden as possible, while pro-

viding a backstop to catch the rogue services and service uses that your human-centric processes don't catch. Of course, there's no perfect solution – the human element still plays a key role. To reduce risk, you have to reduce the complexity of the manual processes – so remember to think strategically about which rules are really necessary, and which are just “nice to have.” ■

About the Author

Dan Foody, CTO of Sonic and Actional Products, leverages his extensive experience in enterprise systems software toward designing robust and manageable Service Oriented Architectures. Foody's experience with distributed systems technologies including middleware, integration and Web Services, gives him a broad knowledge of the complexities and requirements for managing real-world enterprise software deployments. He is the author of various standards, and has contributed significantly to the OMG standard for COM/CORBA interworking. Recently Foody received InfoWorld's 2005 CTO 25 award. Foody holds a BSEE and MSEE from Cornell University.

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

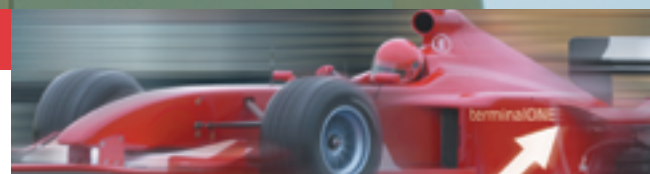


The future of EDI is B2B over IP

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



ChoicePay: Rising above the SOA Testing Challenge

**SOA testing doesn't
stand alone**

WRITTEN BY SALMAN AKHTAR

➤ ChoicePay has embarked on a strategic Web Services-based SOA initiative as part of its ongoing effort to improve customer and partner service. To meet its service improvement objectives, ChoicePay builds reliable and robust Web Services. It continues to enhance its service objectives without compromising overall quality through short, iterative, and demanding Web Services testing cycles.

Background

ChoicePay handles electronic bill payments for some of America's largest companies. Starting operations in 1996, it was a pioneer in the Electronic Bill Presentment and Payment (EBPP) industry and is recognized today as providing one of the most comprehensive suites of payment channels and options in its class.

ChoicePay's adoption of Service Oriented Architecture (SOA) stems from its continuous effort to improve service for its clients. One of the many SOA-based Web Services launched by ChoicePay enables centralized account number validation across all of its bill payment clients, encapsulating hundreds of masks, algorithms, and rules intelligently. As with other systems it has developed, this Web Service went through a meticulous design and implementation process. The intended goal was to provide a service that would cause minimal friction when integrated into a client's IT infrastructure. To reach this goal, the developers had to design a flexible service that would be readily consumed by multiple companies with diverse hardware and software platforms, operating systems, and programming languages. The SOA developers also had to ensure that the services were robust and reliable, which meant that the QA team had to ensure that all operations exposed by the service were thoroughly tested with a large set of data permutations. And for ChoicePay's Web Service, these rigorous and broad testing requirements had to be addressed in a span of only a month.

How ChoicePay Did It

"Testing a new function based on Web Services with global business visibility and produced on a tight schedule, posed a twin challenge for us," ChoicePay CTO Keith Fulton explains. "Throwing more bodies at the problem wasn't a viable solution. Our implementation required intelligent tools and techniques."

When Fulton initiated the SOA testing plan with ChoicePay's software QA team, he first identified the key objectives in order to roll out the service:

- Verify correctness and an exact one-for-one match with the functionality the service was replacing and centralizing, which involved testing several thousand business cases and validating the results
- Boundary conditions were tested to ensure the robustness of the Web Service
- Each operation exposed by the service had to meet specific performance metrics
- Ensure that the service met interoperability criteria since it was to be consumed by multiple companies

The recognition of these objectives introduced several issues:

- All of the objectives had to be met with limited human resources
- Creating repeatable baseline tests was required so iterative regression testing could be done
- The bug reports that were generated had to be precise and easily understood to ensure that developers could create quick fixes to make the project timeline as short as possible

Meeting the Challenge

The initial task was to select the right testing technology to quickly meet the objectives set by Fulton and overcome any hindrances. This included rapid deployment and a limited time for testers to come up to speed with the new product. The criteria to select the right testing technology was based on it meeting current testing objectives, being easy to use, cost-effective, and extensible for future testing scenarios. From an extensibility perspective, ChoicePay

wanted to select a testing tool that would fulfill the following core functional requirements:

| Requirement | Description |
|------------------------------------|---|
| Functional Regression | Validate the integrity of the Web Services and allow baseline regression suites to be created |
| Performance | Ensure that throughput requirements are met |
| Interoperability Compliance | Maximize client interoperability for the Web Services |
| Vulnerability Assessment | Validate the error-handling robustness of the Web Services |

The goal behind these requirements was to find a unified SOA testing technology that would be reusable across multiple IT groups during pre-deployment and post-deployment. Also, given the tight testing schedule, it was important for ChoicePay to have an easy-to-use technology that didn't require a significant time investment for training. The testing team didn't have time for boot camp-type training.

After evaluating SOA testing technologies from several companies, ChoicePay selected Crosscheck Networks SOAPSonar Enterprise. ChoicePay was most impressed by the ease-of-use and rich functionality that not only addressed ChoicePay's stringent testing criteria, but also gave the team immediate productivity improvements. It took less than an hour for the ChoicePay testing team to become effective using the product.

Besides the ease of use of SOAPSonar, the support team at Crosscheck Networks was responsive. Each issue raised by testers was immediately discussed and alternatives given without impacting the project timeline. This allowed full and complete testing during a short time frame for the implementation. "We rely on agile competitive partners in technology and Crosscheck Networks proved to be exceptional," Fulton said.

As the testing plan was taking shape, ChoicePay continued developing effective techniques to maximize their utilization of SOAPSonar. The first technique focused on test-case usability. Borrowing from SOA's emphasis on reuse, ChoicePay's QA team created a set of test cases that could be reused across multiple SOA testing functions such as functional regression, performance, and interoperability.

The second technique adopted by ChoicePay was to leverage external data sources to dynamically parameterize SOAP requests as well as expected SOAP-response success criteria. This technique enabled the QA team to conduct full regression tests across thousands of test cases. Figure 1 illustrates the basic setup and highlights different components that played a role during rollout.

In Figure 1, SOAPSonar enabled the testing team to create coarse-grained and fine-grained filters to verify the integrity of SOAP responses from the target Web Service. The team was able to create

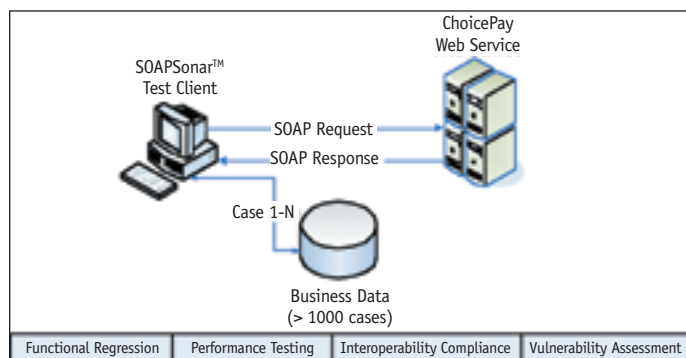


Figure 1: Setup to perform functional regression tests on over 1,000 business cases.

filters based on both HTTP response codes and specific XML elements within SOAP responses. The ability to create fine-grained filters enabled the testing team to quickly identify successful and erroneous responses. Sorting data issues from application issues was simplified by using these filters. Rich reporting also played a key role in facilitating rapid bug fixes by development. Without accurate reporting, the team wouldn't have been able to convey the bugs to the development team and identify whether the issue was code-related or database-specific.

Next Steps

After its initial success, ChoicePay will continue performing real-time integration with clients where critical business processes will be interdependent. "We will use SOAPSonar not only internally, but also to certify our clients' Web Services on a daily basis to ensure there aren't changes or upgrades on their side that impact our ability to do business," explains Sandee Wagner, QA lead at ChoicePay. The plan here is to consume multiple Web Service Definition Language (WSDL) files into SOAPSonar and schedule a test on a daily basis where multiple test cases would be run against remote client Web Services, and reports would be compared from a previous day's run. Figure 2 illustrates the technique to test the customer's Web Services.

Summary

After successfully implementing the test plan, ChoicePay's QA team was able to achieve the project's goals and provide confidence that the Web Services were adhering to functional, performance, interoperability, and security requirements.

Fulton is confident they have gained considerable experience in the last few weeks with timely Web Services-based rollouts, saying, "Our experience with SOAPSonar has given us the ability to provide best-in-class reliability in highly complex, real-time distributed systems across multiple enterprises." Some of the valuable lessons learned during ChoicePay's development and release exercise include:

- Minimizing the impact of Web Service complexities through the effective use of good testing tools
- Developing test cases that are readily extensible through the use of parameters and external data sources
- Reusing key validation criteria iteratively through a project's lifecycle
- Create ongoing controls to monitor live Web Services where dependencies exist

For ChoicePay, SOA testing doesn't stand alone. SOA testing spans complex internal and external services in different phases and versions of the service lifecycle. For ensuring business growth through customer and partner integration, ChoicePay has made reliable and robust SOA a core blueprint of its corporate strategy. ■

About the Author

Salman Akhtar is the CEO and Co-Founder of Techlogix and has over 15 years of industry experience. Salman holds two degrees in Electrical Engineering from MIT.

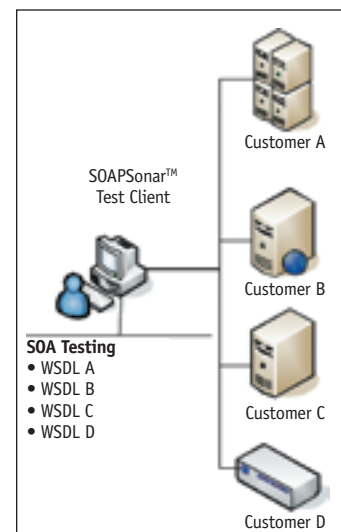


Figure 2: Loading WSDLs from multiple customers for regression testing

SOA Access Control

Policy Management

Approaches, Common Pitfalls, and Best Practices



WRITTEN BY KEVIN SMITH

➤ When SOAP-based Web Services solutions began appearing five years ago, one of the major challenges was securely propagating end-user identity in Web Service chaining scenarios. Certainly a user could authenticate to a portal, and that portal could talk to a Web Service that talks to another Web Service that talks to another Web Service (and so on), but the big question was – how could each point in the Web Service chain be assured who the requesting end user really was?

Initial trials of identity propagation solutions became like the “Kevin Bacon game.” The assurance of the end user’s identity would be based on trusting each connection in the chain and, as we know, there’s no limit to the number of Web Service hops that could occur before the lifecycle of a SOAP request reaches its final destination. Much like the game of “Pass The Secret” that we play in kindergarten, where a secret is whispered in one ear and passed around in the room, there was little assurance that the final recipient had the right information.

Years later, SOA security architects now have blueprints for propagating end-user identity and attribute credentials in a multi-tiered SOA environment. Leveraging mature standards such as XML Signature and SAML (the Security Assertion Markup Language), the WS-Security SAML Token Profile provides a mechanism for trust propagation in Web Services. This standard, along with other

similar token-based standards, give us opportunities and choices related to access control management and enforcement in the enterprise.

Because of these Web Service security standards, we've moved from the problem of asking, "How do we know who the user really is?" to "How do we enforce access control policy for this user?" Leveraging identity propagation standards in Web Services, there are usually two common approaches of SOA access control policy management:

- 1) **Centralized Approach.** Using a "Yes/No" policy server, a handler or component enforcing access control for a Web Service looks at the propagated identity and must call a central server essentially saying, "Can user X access this Web Service?" This model, shown in Figure 1, is completely centralized, as it involves centralized management of all access control policy, and also centralized decision-making.
- 2) **Decentralized Approach.** Completely different from a "Yes/No" policy server, Web Service containers express, manage, and enforce local policy based on global identities and attributes propagated to the Web Services. In this model, shown in Figure 2, a handler or an enforcement component inspects the identity and attributes propagated to the Web Service, does a local lookup on the Web Service policy, and makes an enforcement decision. This model is completely decentralized because it uses decentralized management (management expressed by each Web Service container) and decentralized decision making.

Pros and Cons of the Centralized Approach

The centralized approach is the most common as anyone can see by looking at the enterprise policy server market. There are several benefits to this approach:

- **Information Hiding.** From a security perspective, a completely centralized approach leads to information hiding, which can be a very good thing if you want your reasons for access control decisions to be secret. (That is, if you ask the question, "Can person X access resource Y," you get a response, but you don't really know the reason for the response.) This is also beneficial if you only want identities to be propagated and not necessarily the attributes about users to be propagated, if any of these attributes are confidential.
- **Complete Control.** Finally, if you have a central point in your enterprise that makes all the decisions, the administrator has centralized control of everything, where access control changes for all enterprise Web Services can be made with the touch of a button, which is a great thing. A centralized policy server making all access control decisions means that you also have centralized auditing, which is a lot better than having to look through the logs on the machines throughout your enterprise.

Having listed these benefits of the centralized approach, there are also potential pitfalls:

- **Poor Scalability.** If an enforcement point for every Web Service in your enterprise needs to make a call to your policy server for every access control decision, there will be a large load on that server, and you'd better hope it never crashes. This is a potential Denial of Service (DoS) attack waiting to happen. If your policy server does go down, you need to ask yourself – "Should I give access to everyone, or should I deny access to everyone?" Believe me, you don't want to go there.

- **Poor Performance.** If policy enforcement points for your Web Services have to make network calls to a policy server for every access control decision, it will slow your applications down. Calls to policy servers must be cryptographically protected, since the enforcement point will need assurance that it's talking to the policy server and that there's message integrity in the message response. The result of this added network call and cryptography will be poor performance. Certainly your local enforcement points can cache these policy decisions for a certain amount of time to avoid repeated calls to the policy server for the same user, but the cryptographically protected initial calls to the policy server – combined with what will surely be a heavy request load on that policy server itself – will lead to your applications slowing down.
- **Potential Management Burden.** If you choose a centralized policy server model, this means that there will be one point in your enterprise that makes all access control policy decisions. And this means that either your policy server connects to other servers to get those policies for every Web Service in your enterprise, or your central policy server will have to manage policy for every enterprise application, which could be a heavy burden.

Pros & Cons of the Decentralized Approach

As a result of seeing many completely centralized approaches fail, another model that's used is a more decentralized approach, where local policy is expressed based on end-user credentials that are propagated to the Web Service and local policy decision points

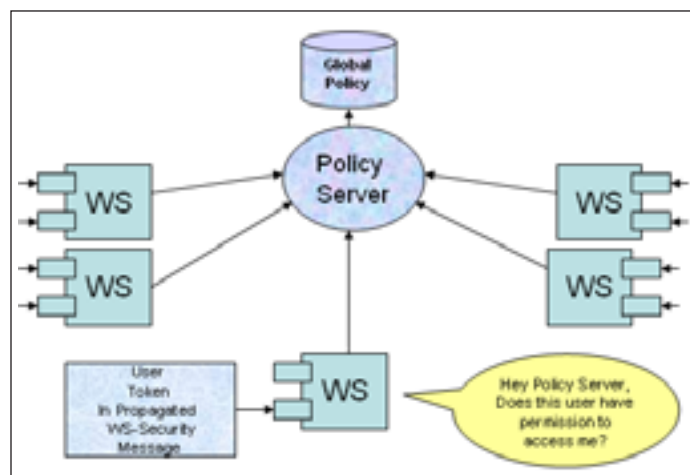


Figure 1: Centralized approach

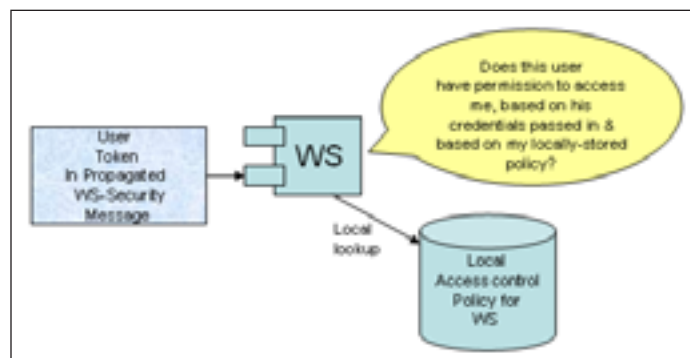


Figure 2: Decentralized approach

make decisions based on that local policy and the end-user's credentials.

The benefits of this approach are obvious – by doing this, we eliminate all of the pitfalls of the centralized management approach. Instead of having to ask “Mother may I?” to a policy server every time a decision has to be made, a local Web Service handler is empowered to make a decision based on the identity propagated in and based on the local policy expressed by the Web Service itself. No policy server needs to be running using this model, eliminating the concern of the policy server being a bottleneck or a performance burden. In this model, each Web Service container manages its own policy – eliminating a potential centralized management burden.

Having said this, there are also potential issues with this approach. By moving to a completely decentralized model, we lose the benefits that the centralized model adds. Most importantly, there is an issue here of control. In an emergency situation, where a policy change must be made to deny access to security violators, how could we quickly lock down and protect every Web Service from those security violators when using a completely decentralized model? Would the enterprise administrator have to call up every Web Service provider, asking him to change his policy?

The lack of control that a completely decentralized model brings is problematic. As a result, there's a need for a different model that leverages the pros of each of these approaches.

Best Practices

At this point, I've discussed two common models – completely centralized and completely decentralized. There are benefits to each, but there are serious pitfalls associated with each model. These are, in fact, two extremes, and yet they are still the most common models for SOA security access control policy management. The dilemma we face is that neither model offers us the best solution.

The answer to this dilemma involves creating an architecture that merges the best of centralized policy management and the best of delegated decision-making eliminating those architectural items that lead to pitfalls. This article proposes the following:

- 1) End-user credentials are propagated in WS-Security messaging using one of the token profiles

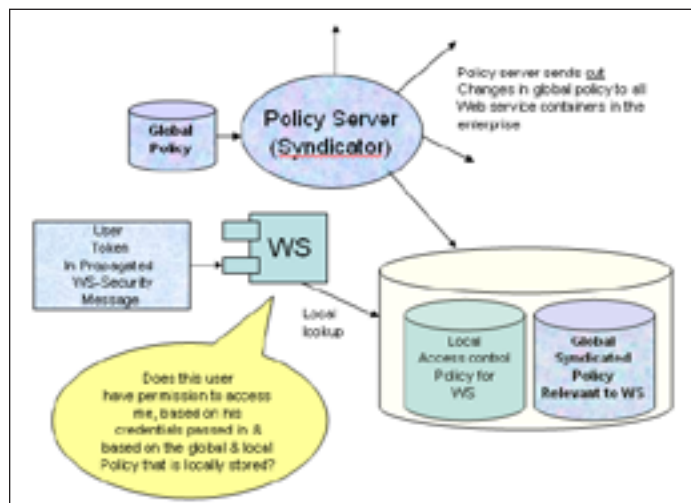


Figure 2: A hybrid approach

- 2) Global Security Policy is managed by a central authority and *syndicated* to local policy decision points
- 3) Local Security Policy is managed by Web Service owners (optional)
- 4) Local Policy Decision Points enforce locally stored local and global security policy based on credentials propagated in Web Service calls

Figure 3 shows a diagram of this model that in most cases will represent the best of both worlds – taking some things from the centralized approach, and some from the decentralized approach.

This model answers many of the dilemmas discussed so far here by using a *policy syndication server*. If we can avoid using a “yes/no” request/response policy server and instead have a policy server *syndicate* global policy that Web Service security handlers in the enterprise enforce, there can now be centralized access control enforcement. This provides the benefit of total control that the decentralized model was lacking and eliminates the availability threat, the enterprise bottleneck, and the performance concerns that were inherent in the centralized model.

Allowing Web Services to express their own “local” access control policies removes the potential management burden of having to dictate policy for the entire enterprise. The trick, however, will be conflict resolution between syndicated global policy and local policy, since global policy must always trump local policy.

Centralized auditing can be handled by using network logging and a central auditing server, where all access control events from local Web Service handlers are sent to a central auditing or Web Service management server.

What this model lacks, however, is the benefit of information hiding that's present in a completely centralized security policy model. As we addressed before, the yes/no policy server abstracts the reason that decisions are made, which can be a very good benefit in situations where the security policies themselves are extremely sensitive. This is very uncommon, and if this requirement exists, it may only pertain to some components in your enterprise. If this is the case, you can complement the solution I'm describing with a centralized policy server used only when absolutely necessary.

Conclusion

This article has provided an overview of some of the issues that organizations face regarding SOA access control policy management and enforcement, looking at the benefits and pitfalls of two common methodologies. We presented a best-practice approach that can be used in your enterprise. As with any security solution, it's important to focus on both your short-term, long-term, and even potential security requirements to plan the most scalable approach. ■

About the Author

Kevin T. Smith is a technical director at McDonald Bradley, where he leads the SOA & Semantics Security Team (S3T) focusing on information assurance initiatives for multiple projects. An author of several technology books on XML, Web Services, Java development, and the Semantic Web, he is a frequent speaker at many conferences such as JavaOne, OMG Web Services, the Association for Enterprise Integration (AFEI), and Net-Centric Warfare.

kevintsmith@comcast.net

Take C++ with Your Java?

HydraSDO™ from Rogue Wave Software

C++

```
class A  
{  
public:  
    A() {m_value=0;  
    int getValue()  
}
```

Shared
Memory
Access

Java


ROGUE WAVE
SOFTWARE
A QUOVADIX DIVISION

Free Evaluation

www.roguewave.com/developer/downloads/

The Optimization Appliance

A field guide to distributed processing in a Service Oriented Architecture

WRITTEN BY TOM YOHE

➤ An efficient Service Oriented Architecture (SOA) implementation distributes as much processing as possible to trusted appliances in the nearer tiers, where intelligent content-based routing decisions made by highly efficient processors can also perform caching, transformations, and other functions. This article will present a detailed example of a “Las Vegas Casino” that has been implemented as a set of distributed Web Services and provide a step-by-step guide for delivering these services. The implementation of this virtual casino extends from the farthest tier of the central database engine all the way out to client, where acceleration has been transparently injected into the browser for an optimal user experience.

The Las Vegas Casino manifests itself to the user as an Asynchronous Java and XML (AJAX) application, with a rich GUI of slot machines, roulette wheels, Texas hold'em, and of course blackjack. Each of these is supported by a highly scalable set of Web Services. The XML traffic between the client and data center is mostly Simple Object Access Protocol (SOAP) request/responses transported over an optimized HTTP/S protocol with unique features such as bi-directional compression, “TurboStreaming,” and XML document differencing. The XML “front gate” that is situated at the nearest tier of the data center analyzes the traffic and classifies the user based on an authorization realm upon which sophisticated decisions can be made based on application policies.

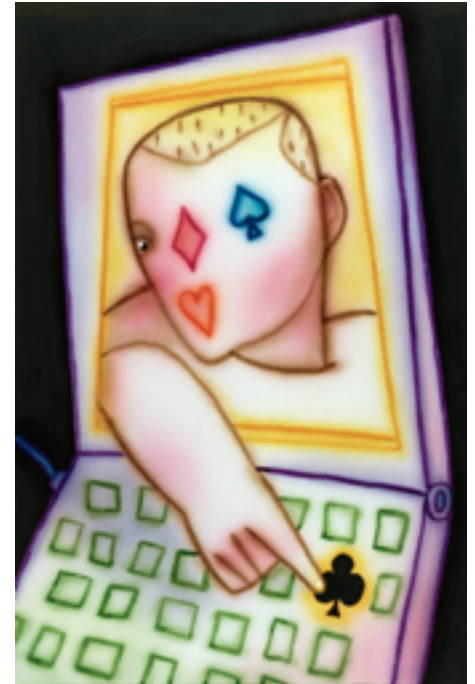
The application policies control how XML content processing should be performed. Foremost is protecting the virtual casino from malicious XML-borne threats and informing the casino bosses that threats have been encountered and averted. Incoming XML requests are also validated to ensure that they conform to one of the virtual casino's schemas. Each request is then analyzed against a set of XPath statements that gov-

ern how the request should be transformed and then a different set of XPath statements that determine which enterprise application server in the farther tiers should handle the now validated and transformed request. When possible a response is handled from a cache located at a nearer tier.

This article will also highlight the performance measurement techniques employed to measure the response time of the various services of the virtual casino. Service level agreements are established and alerts are sent out when response time falls below the compliance threshold.

Backdrop

The patrons of the virtual casino enjoy a robust graphical user interface that is presented by their browser. These users find comfort in the padlock shown on the status bar that proves that all traffic is flowing over an encrypted tunnel. The rich user interface seems to effortlessly convey the sights and sounds of a casino atmosphere along with a vivid portrayal of their account status. Back at the data center, the database servers, which are the ultimate source of this presentation, operate smoothly and securely processing a steady flow of transactions.



The owners of this enterprise have designed a business model where a small but fixed percentage of all wagers flow directly to the bottom line. This lucrative business is the result of hiring top-notch service-oriented architects who understood how to make effective use of optimization appliances to deliver an exciting product to the customers in a completely secure fashion.

The enterprise architects were tasked with meeting several important objectives:

- The data center had to be completely safe from malicious attacks.
- Customer confidentiality had to be protected.
- The customer experience had to be vividly rich and minimize consumption of I/O bandwidth.
- The system had to scale and be impervious to single points of failure.
- Response time to customer transactions had to appear as instantaneous as it would in a real casino.

To meet the above objectives, the architects decided to implement a set of Web Services, each with a very clearly defined interface. The following services were implemented:

- Account Registration – Establish user ID, password, credit card.
- Account crediting/debiting – The other “gaming” services interface with this service as games are won or lost.
- Gaming Services
 - BlackJack
 - Slots
 - Texas Hold'em
 - Roulette

An AJAX paradigm was used to develop the rich graphical user interface. This model allows graphical objects to be manipulated by the client processor while transaction updates are communicated to the data center by posting SOAP requests.

The back-end database servers and Web Service processors are insulated from threats by employing an Optimization Appliance (OA). The judicious use of XML content processing appliances was the key to a successful build-out of this SOA. The OA takes care of the following:

- User authentication
- SSL encryption
- WAN optimization
- XML threat protection
- XML content-based routing, transformation, and schema validation

The enterprise architects were thrilled that the optimization appliance's “Acceleration on Demand” (AOD) feature would inject bi-directional optimization of all the AJAX/SOAP traffic without impacting the development efforts of the AJAX application. AJAX programming is a tough enough field; by transparently injecting AOD into the application the AJAX programmers were free to concentrate on object-oriented development, knowing that WAN optimization will be taken care of by AOD.

Now that the backdrop has been painted, the remainder of this article will discuss the steps taken to integrate scalable optimization appliance into the SOA implementation.

Designating the XML Front Gate

The first challenge is to ensure that all of the external HTTP traffic is directed to the optimization appliance. This is accomplished by having the DNS of the server portion of the URL resolve to the OA. The OA typically has “external” (public) ports that are protected with intrusion detection and other basic Internet attacks and “internal” (private) ports that interface to the

other services of the SOA implementation. The software that runs on the OA functions as an important insulator between the wilds of the Internet and the well-behaved Web Services of the data center.

Intelligent Port Definitions

The OA is configured to securely insulate the data center by only listening for incoming TCP sessions on predefined port definitions that associate the external IP address/port pairs with SSL encryption certificates. The SSL encryption certificates are text documents that have been “signed” by a certificate authority and provide credentials to the end user that they are securely connected. The certificate documents are uploaded into the OA and stored into a tamper-proof key store. In the case of the “virtual casino,” only one SSL encryption certificate document is needed because all SOAP requests are directed to the same URL. The certificate makes it possible to conduct SSL sessions between the OA and the customers, this SSL traffic is terminated at the OA and the OA in turn communicates to the back-end servers over unencrypted channels.

Signing In

All new TCP connections are expected to be HTTP/S and any other protocol is rejected. After completing the HTTP/S, the AJAX application is retrieved by the browser. The first operation of this application is to “sign-in” with the casino's account registration Web Service. The sign-in operation generates SOAP requests that ultimately result in a “cookie” being obtained from the accounting service. The OA insures that

without this dynamically generated cookie, which is cryptographically impossible to guess, no other operations are possible to the other casino Web Services.

The sign-in process also entails the assignment of the user to an authorization realm that's defined on the OA. When subsequent SOAP requests are received by the OA, it bases application policy decisions (such as preferential treatment for a user in the “high-roller” group) on the authorization realm that's assigned.

Acceleration Injection

A very interesting phenomenon occurs when the AJAX application is downloaded. The OA “injects” a powerful ActiveX control called “AOD” into the application that extends the optimization capabilities of the OA all the way out to the client. All subsequent traffic between the OA and AJAX application flows through this optimization engine. This engine does bi-directional compression and TCP session aggregation, which is important because it lets the AJAX application perform its “object-oriented” functions without generating costly new SSL session establishments. This AOD feature is integral to meeting the objective of limiting the WAN bandwidth consumption of the application.

Threat Protection

Now that we've nailed the delivery of an AOD-injected AJAX application we can move on to configuring the XML threat management capabilities of the OA. Fortunately this is easy to do. A simple checkbox (on the default) causes all inbound XML SOAP requests to be screened against a new



Figure 1: Load balancing->edit screen

breed of XML-based attacks. These threats operate on a higher level than the attacks of yesteryear (e.g., SYN-FLOOD), which is effectively defended with intrusion detection devices that operate at the IP packet level. The virtual casino's OA is hardware-assisted by Tarari's unique "XTM" XML threat protection engine. Tarari's patent-pending XML anomaly detection "learns" to recognize threat-bearing messages. The Tarari XTM recognizes dozens of well-known XML XDoS attacks like recursive payload, attribute explosion, and dangling XML, and can also flag traffic that represents previously unknown threats – often on the first message.

Server Pool

The next step in configuring the OA is to define "server pools" that map Web Service requests to a member of a cluster of servers that can all perform the same Web Service requests. The screen below shows an example of a typical server pool definition for the blackjack Web Service.

In the Figure 1 example load balancing->edit screen, you can see how multiple back-end servers are designated by the IP address. The OA will then balance the load between the servers in the pool by routing requests to the host that has the fewest pending responses. The OA can handle the configuration of hundreds of server pool definitions. By defining server pools, we've set the stage for intelligent routing of the Web Service requests.

Schema Validation

The OA also provides the important function of schema validation. To configure this, the administrator uploads XML schema documents, with each schema defined by a target namespace, into the appliance. Subsequently, the OA validates that these XML documents are well formed (correct XML per the W3C XML spec) and then checks that the document conforms to a schema known and approved by the casino. XML schemas ensure that the casino's applications can process the input documents without difficulty. The OA has a massive scalability schema validation capability because it leverages the Tarari hardware acceleration that can parse XML documents in parallel. The casino has defined namespaces for its gaming and registration/authentication Web Services. Any request that's processed by the OA from an external user will have already had its schema vali-

ACCELERATION

Appliance Administration Index Module Index Help..

suse326.dayton.stampede.com : Configuration->SOA Optimization->Content Based Routing->Edit

Action Characteristics

Action Name:

Comment:

Enable: ☒ Enable ☐ Disable

Action Class

☐ Filter

HTTP response code:

Message:

☐ Transform

☒ Reprocess transformed document

☒ Route

Target server pool:

Action Conditions

☐ If any

☒ If all

☐ If none

of the following built-in conditions or XPath statements are TRUE

Built-in Conditions

☐ Validate properly formed SOAP requests

XPath Statements

```
#
# This XPath identifies the message as a Gambling soap request
#
/*/*[local-name()='Body' and namespace-uri()='http://schemas.xmlsoap.org/soap/envelope/']/Gamble
#
# This XPath identifies the Gamble type to be poker.
#
/*/*[local-name()='Body' and namespace-uri()='http://schemas.xmlsoap.org/soap/envelope/']/Gamble/Poker
```

Figure 2: Content-based routing->edit screen

dated, freeing up cycles for the back-end Web Service so that it can focus better on performing the service.

Transformations

Again, with a hardware assist, the OA provides the ability to perform stylesheet transformations on the SOAP requests before they're presented to the back-end server. Since the casino has an international presence, its account crediting/debiting Web Service uses an XSLT to ensure that euros

are converted into dollars. Another transformation example would be if, after rolling out the AJAX application, the casino bought a new roulette service that was expecting data in a format different from the previous one; a stylesheet could be used to transform the requests into the new format as well as transforming the responses back into the old format.

The first step in configuring the XSLT transformation is to simply upload the stylesheets into the OA's cache. The

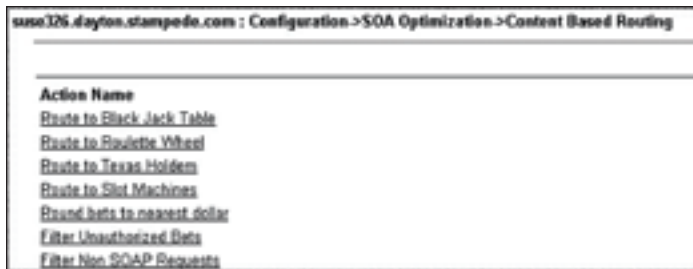


Figure 3: Content-based routing action pick-list

uploaded XSLTs can be referenced later when defining content-based routing actions.

Content-Based Routing

The heart of the OA's XML content processing power is harnessed by defining content-based routing actions. The conditions that predicate these actions are specified using XPath statements. XPath is a W3C standard for querying XML documents, that is, they can test for specific content in XML documents. Entering XPath statements into a form configures these rules. As many XPath statements as needed may be entered. There are three classes of content-based routing actions:

1. **Filter** – Reject the request by sending back an HTTP response code.
2. **Transform** – Process the request by performing a stylesheet transformation. The document may then be reprocessed against other XPath statements after the transformation occurs.
3. **Route** – Send the request to one of the members of the server pool.

Action will be taken based on the results of a "truth-table" that is generated when parsing the document against the XPath rules associated with the action. The criteria can be "any," "all," or "none" of the conditions. If the hardware-assisted content processor determines that the criteria have been met then the action will be taken.

In the content based-routing->edit example screen above, two simple XPath statements have been entered that dictate that if the SOAP request contains both the "Gamble" and "Gamble/Poker" namespaces in the "Body" then the request should be routed to the least busy server in the Texas Hold'em server pool. After saving the definition it will show up in a pick-list as shown in the content-based routing pick-list screen.



Figure 4

As many XPath statements as required may be entered without incurring a significant performance penalty because the XML content processing silicon that analyzes the message can process the statements in parallel. The OA's ability to perform this type of analysis on each message allows the AJAX application programmer to concentrate on converting the operation of the rich GUI into SOAP requests without having to worry about directing the request to a specific server URL.

Application Policies – Tying It All Together

The virtual casino uses application policies to tie the raw capabilities of the optimization appliance together. It's actually the application policy that instructed the OA to inject the AOD and to do the threat management checks. The application policy is what determines the selection of content-based routing actions that should be performed on the XML SOAP requests. Multiple content-based routing actions can be selected in a single policy as well as the order in which the decision analysis will be done.

The OA selects the application policy that should be applied to a request based on a combination of the URL in the HTTP request and the authorization realm that was established during the sign-in stage.

An example excerpt of an application policy that ties it all together for the virtual casino is shown in figure 4.

SLA Compliance

Since the OA is at the heart of it all and has extended its reach by injecting an AOD component out to the client, it's in a unique position to measure the response time of the application. Embedded in the Web Service Description Language (WSDL) that describes each of the virtual casino's ser-

vices is a statement that sets goals for an acceptable response time for each transaction definition. These goals represent service level agreement (SLA) thresholds. The AOD injected into the virtual casino's main AJAX application is aware of these thresholds and embeds statistics into every request from the requester to the OA at the data center that indicate continued service-level agreement compliance.

Conclusion

This article has illustrated how an optimization appliance can easily be configured to become a strategic component of a SOA by offloading XML content processing and encryption, intelligently routing SOAP requests, thoroughly optimizing the data flow out to the consumer, providing threat protection, and monitoring for adherence to service level agreements. ■

References

- *XML Programming with PHP and Ajax* by Hardeep Singh DB2 Magazine, 3 Quarter 2006.
<http://www-128.ibm.com/developerworks/db2/library/techarticle/dm-0511singh/>
- 2005: *The Year Mainstream Networking Embraced XML* Michael Leventhal, Tarari, Inc. XML 2005 Conference Proceedings
<http://www.idealliance.org/proceedings/xml05/abstracts/paper252.HTML>
- *Stampede Web 2.0 Performance Series Product Description* <http://www.stampede.com/web-2-0-performance-series.html>

About the Author

As VP of engineering at Stampede Technologies, Inc., Tom Yohe currently leads one of the world's most elite enterprise optimization engineering teams. He has been delivering award-winning enterprise products for more than 25 years, and has been granted numerous patents for unique data communications optimization techniques. Tom has a computer science degree from Penn State.

Did You Know There's a "C" in SOA?

Don't Forget the Consumer in Service Oriented Architecture

WRITTEN BY GUS BJORKLUND

➤ When designing your SOA and services, keeping the service consumer in mind will make the job easier. Consumers must conform to the interfaces of each service they use and invoke them with the right data in the right format. The more similarity there is among services, the less coding and translation your consumers will have to do. Using the techniques of transformation, semantic data modeling, and a conceptual data model can make your job much easier – both during initial design and testing and when making changes later.

The interface specifications, protocols, and data formats used in SOA and Web Services are designed to create services with “loose coupling” between the service consumer and service provider. The service provides a consumer with an abstraction representing some business function so that the consumer doesn't have to be concerned with the details of how the function is implemented or how and where its data is stored. The only thing that the consumer has to know is how to call the service using standards like WSDL, SOAP, and HTTP.

Once you've implemented some services you can then compose higher-level business functions and so-called “composite applications” by invoking multiple services and orchestrating interactions among them. If you build all your services from scratch for a project you can design interfaces and data models that interact smoothly and consistently. Opportunities for such green-field designs are rare and today's new system is tomorrow's legacy system anyway. More than likely, you'll be using existing as well as new services and applications that were bought or built long ago and have or will have service interfaces added to them.

The Fly in the Ointment

An application's service interface may be new, but it probably won't be clean or consistent with your other services since an application's functionality and exist-

ing interfaces usually dictate what can be provided as a service, as well as what data it can accept and provide. Interacting with multiple services can become complicated – even though they use standard protocols and formats for interface definitions and invocation, these protocols and formats only deal with the syntax of the data and not with their vocabulary or meaning. Each application or service is likely to use different names for the same things and the same name for different things. Such disagreements and mismatches are commonplace and not limited to Service Oriented Architectures. Here are a few examples:

- Many of us have an address book on our computers, another on our cell phones, and software that synchronizes them. Unfortunately the definition of an entry in each is different, with variations in field names and the number of fields. One may have fields for home, work, cell, fax, and modem numbers and an e-mail address as well as the other home, work, mobile, pager, assistant, and main numbers and multiple e-mail addresses. The synchronization software knows how to map the different names but it can't map all the fields one-to-one. Still it's a good enough compromise because most people don't use all the fields.
- Many companies provide access to an external CRM system – like salesforce.com – for salespeople and use an internal ERP system for processing orders. Those



two systems keep different information about customers so using the salesforce.com Web Service interface to get customer information and then correlating it with data from ERP system may not be straightforward. Like the address book, the data formats are different, but so is the intended use and content of the data. One keeps track of things like prospects and their interactions with the sales rep and the other keeps track of things like what someone has bought, how much they paid, and when.

The two systems both have a notion of “customer” but one may call the customer identifier “cust_num” and the other something like “csidno” and one may contain a number while the other contains both letters and numbers. One may have “orders” represented as an XML document with elements for “orderlines” and the other an “orders” XML document with the elements “orderheaders” and “orderdetail.” Other data elements will present similar issues. Reconciling or mediating the differences in data models and service interfaces for applications like these is more difficult than reconciling differences in address books.

The consumer bears responsibility for providing the correct data elements to the service in accordance with its interface definition. In many situations, the service designer gets to decide what it wants to use. When you design service interfaces, you

should try to make it as easy as possible for the consumers to use. That isn't always possible, especially if you use services provided by someone else like another department or an external organization like salesforce.com, UPS, or Google.

Transformation to the Rescue!

To mediate the differences among services' data formats you can introduce a data transformation or mediation layer into the architecture. This layer becomes responsible for mediating data formats between service invocations, performing conversions and translations for the consumer. The consumer then has the data in the right format when it wants to invoke a service using data supplied to it by another service.

You can construct the transformation operators using XQuery, XSLT, or even hand-coded Java functions to translate one data layout and vocabulary to another with relative ease. If you're using an Enterprise Service Bus (ESB) as a backbone for reliable communication and service management you can put your transformation services on the bus. The ESB can then automatically invoke the transformations as needed.

When you build transformations one

by one as you need them, the result is a set of transformations for converting data received from service A into the right form for sending to service B, service A to service C, service B to service C, and so on. Each transformation does one specific job in isolation from the others. Now everything will run smoothly, right?

Transformation Isn't Enough

While transformation is necessary and helps to make service integration easier by itself it's not enough. Even when you use effective tools like DataDirect's Stylus Studio or Altova's XMLSpy to create them, having many such point-to-point transformations can make things complicated:

- It reintroduces the tight coupling SOA was supposed to eliminate in the first place. Instead of providing agility your SOA will become brittle and difficult to change.
- We've spent more than 50 years inventing new applications and their data models so there can be many complicated mismatches between applications. Every organization has a lot in their systems. Dealing with them can keep you busy for a long time.

- It's pretty likely that you'll end up with duplicate sections and duplicate operations in your transformations perhaps with unintentional variations, especially if different people make them.
- Maintaining hundreds or thousands of transformations is a big burden and keeping them all correct, consistent, and current will be next to impossible.
- Individual transformations encapsulate knowledge of the data semantics in many different places. That knowledge can't be found easily or reused effectively.

Transformations certainly help but they aren't enough.

Conceptual Data Models to the Rescue!

Instead of many individually designed point-to-point translations, if you define mappings from each service to a common *conceptual data model* then it will be possible to generate the transformations you need automatically and use them where needed. This can greatly reduce the amount of work you have to do because instead of defining transformations for pairs of services, you define one mapping for each



Looking to Stay Ahead of the i-Technology Curve?

Subscribe to these FREE Newsletters >

Get the latest information on the most innovative products, new releases, interviews, industry developments, and i-technology news

Targeted to meet your professional needs, each newsletter is informative, insightful, and to the point. And best of all – they're FREE!

Your subscription is just a mouse-click away at www.sys-con.com

| | |
|--|----------------------------|
| IT solutions | NET JOURNAL |
| JDJ | WebServices JOURNAL |
| Information STORAGE+ SECURITY JOURNAL | LinuxWORLD JOURNAL |
| LINUX BUSINESS WEEK | wireless JOURNAL |
| MX developer's JOURNAL | XML JOURNAL |
| wldj | WebSphere JOURNAL |
| | COLOFUSION |

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

service. At this point, the necessary point-to-point transformations are generated from the mappings from service to conceptual model.

Note that there's no need to actually transform to and from the conceptual model (that's why it's conceptual, not real). With the right tool for defining and extending the conceptual data model and service mappings and storing them, this approach solves a number of the problems we mentioned previously.

As you define and extend the conceptual model, you can use clear and understandable names for all the data elements no matter how cryptic they may be in the service interfaces. When the same data element is used by multiple services, the modeling tool can show them together enabling you to see where an element is being used even when the names are different.

Data Validation

The service interface standards we have now don't include much ability to express the rules or constraints that must be applied to the data presented to the service. For example:

- A retail customer may not be allowed to use purchase orders, while commercial customers in good standing can;
- There may be several related fields that must be validated together, such as street address, city, state, and ZIP code, and so all of them must be present;
- A shipper can't be assigned to an order until a shipping address has been specified; and
- Two addresses – current and future – are required in a move order for relocating broadband cable service.

A service always has to validate its inputs but, in loosely coupled systems, it will often be necessary to do data validation of consumers as well – especially in service consumers that interact with users who want early feedback in data entry, for example, before an order is submitted.

The common model can be used to capture such constraints so they can be applied everywhere they're needed in a uniform manner. When the validation logic is generated from these rules you can be sure they're the same everywhere.

Calculations

Often a data element used in one service may have a different data format than another and the values may be expressed in

different units (inches versus millimeters, Fahrenheit versus Celsius, pounds versus kilograms, and so on). One service may define a value with three decimal places (e.g., 3.602) and another as a whole number (e.g., 3). Should the decimal value be rounded or truncated when converting?

The conceptual data modeling tools should be able to define such commonly used conversions as well as application-specific conversion or calculations so that they can be defined once and used where needed. This will make the conversions easier to change and improve overall system reliability.

Industry-Standard Data Models

Standard data models have been defined by many different industry-specific groups such as insurance (ACORD), healthcare (HL7), and telecommunications (SID). Most of these data models are large and complex, with 1,000 or more classes. If you can use one of these data models, it can save you considerable effort and ease conformance with your industry's standards.

To use these standards, you'll have to add them to your conceptual data model. Since they're usually large and complex, the modeling tool has to have a means to import them. Once imported, you'll also want to be able to add layers on top of the industry model without changing it. For example, since these data models are usually abstract, attribute names tend to be generic, such as a set of "ContactMediums" instead of "main telephone number" and "home telephone number." You'll want to add mappings to the names used by your own data and applications.

The Metadata Repository

The metadata (the mappings, transformation rules, data format conversions, validation rules, and all the other things we've discussed already) have to be stored somewhere where they can be easily used and where the tools can get and save their artifacts. This is the role of the metadata repository. The repository isn't a big shoebox that you can just throw all the metadata and related artifacts into. It should give you a way to organize and keep track of everything.

When you use repository-based modeling tools, besides easier implementation, ongoing maintenance costs will be lower and you can achieve the following benefits:

- **Automated impact analysis** – As you plan changes to your SOA, the tools can provide reports about the interdependencies

between services and consumers and the impact of any proposed changes.

- **Reusable components** – When all the data definitions, transformations, validation rules, conversion formats, and the generated code are stored in the repository, you'll be able to find and reuse elements that have already been developed.
- **Correctness** – By using the elements defined in the repository where they're needed, overall system and application correctness, consistency, and reliability will be enhanced. You have the assurance that a change can be propagated to all the places it needs to go. Your SOA will have fewer bugs.

Conclusion

Take care of the needs of your service consumers to be sure you can achieve the SOA benefits of simpler and faster application and data integration, agility, and the ability to respond to the needs of your business more effectively. Modeling tools, a common conceptual data model, and a metadata repository will promote reuse, reduce errors, and control the effects of change. Then you can bask in the glorious sunshine of your SOA. ■

References

- "Semantic Interoperability of Web Services – Challenges and Experiences" lstdis.cs.uga.edu/library/download/techRep2-15-06.pdf.
- Jim Gabriel. "Best Practices in Integrating Data Models for SOA." *Web Services Journal*, Vol. 5, Issue 2, February 2005. <http://www.sys-con.com/story/?storyid=48031>.
- "Scaling SOA Through EAI Enhancement and Use of Model-based Standards" Pantero Corporation White Paper <http://www.pantero.com/Downloads/Details.asp?DownloadID=15&DownloadTypeID=2>.
- XSL Transformations (XSLT) Version 1.0. World Wide Web Consortium Recommendation. <http://www.w3.org/TR/xslt>
- Stylus Studio XSLT Editor: http://www.stylusstudio.com/xslt_editor.html.

About the Author

Gus Bjorklund is vice president of technology for Progress Software Corporation and works with technical and business leaders across the company's product units to clarify technical strategies, identify technical synergies, evaluate new technology directions, and coordinate cross-unit technology efforts, particularly in the area of data management. Gus joined Progress in 1989 and has over 30 years of experience in domain-specific programming languages, data communications, industrial automation, real-time control, manufacturing quality control, operating systems, and financial systems.



Visit the *New*

www.SYS-CON.com

Website Today!

The World's Leading *i*-Technology News and Information Source

24/7

FREE NEWSLETTERS

Stay ahead of the *i*-Technology curve with E-mail updates on what's happening in your industry

SYS-CON.TV

Watch video of breaking news, interviews with industry leaders, and how-to tutorials

BLOG-N-PLAY!

Read web logs from the movers and shakers or create your own blog to be read by millions

WEBCAST

Streaming video on today's *i*-Technology news, events, and webinars

EDUCATION

The world's leading online *i*-Technology university

RESEARCH

i-Technology data "and" analysis for business decision-makers

MAGAZINES

View the current issue and past archives of your favorite *i*-Technology journal

INTERNATIONAL SITES

Get all the news and information happening in other countries worldwide

JUMP TO THE LEADING *i*-TECHNOLOGY WEBSITES:

IT Solutions Guide

Information Storage+Security Journal

JDJ

Web Services Journal

.NET Developer's Journal

LinuxWorld Magazine

Linux Business News

Eclipse Developer's Journal

MX Developer's Journal

ColdFusion Developer's Journal

XML Journal

Wireless Business & Technology

Symbian Developer's Journal

WebSphere Journal

WLDJ

PowerBuilder Developer's Journal

Get Rich Applications with WorcsNet IAB Studio

Its feature set is too rich to let a few glitches stop you from trying it out

REVIEWED BY PAUL KAISER

➤ IAB Studio is a tightly integrated set of development and runtime tools you can use to easily create browser-based applications, reports, and workflows with rich user interface controls. It makes liberal use of client-side event processing and AJAX-based communication to a J2EE server to provide a more responsive user experience with lighter server loads.

IAB Studio is broken down into three components: Application Builder, Reporting Suite, and Workflow Suite. Application Builder provides all of the containers and controls you need to assemble rich user interfaces complete with data grids, entry forms, and tree-view controls. Reporting Suite adds several capabilities that greatly simplify report and chart production. Workflow Suite lets you use a graphical UI to create workflow models using a variety of tasks types. It executes these models in its own engine.

Application Builder

The terms “rich client” and “rich user interface” have been used for several years to refer to software that make use of a set of controls that exhibit robust behavior (e.g., drag-and-drop) and/or provide more control properties than HTML to construct a UI. These include tree-view controls, tab pages, and multi-page editable data grids. The important difference between rich controls and the standard HTML controls is the degree of interaction the user can have with it and the amount of application state that can be represented with one instance of the control.

WorcsNet uses the term “Rich Internet Application” or RIA to refer to an application built with a rich user interface that runs in a Web browser. Application Builder is the primary tool used in constructing these applications.

The WAB Web App

The Application Builder tool is itself a Web application. It runs as a standard J2EE Web application and is compatible with a variety

of relational databases including Oracle, MS SQL Server, Sybase SQL Server, Sybase SQL Anywhere, MySQL, and SAP DB. You can download a copy from Worcsnet (<http://www.worcsnet.com>). There are three installation methods you can use. I chose the simplest: the bundle consisting of a single ZIP file with IAB Studio, JBoss, and MySQL already set up and ready to go. Just expand the ZIP, point JBoss to your Java home, and start them up.

Workspace

Once you have Application Builder running, point your browser to <http://localhost:19080/WAB> to get started. Follow the instructions to log in as the admin user. The sample project includes several pages demonstrating the various types of controls available. It was nice to realize that the controls used to construct the Application Builder IDE are the same ones available to build your own.

Projects

Projects are simple hierarchical structures that you use to organize the artifacts you create with IAB Studio. At the top level, you can create instances of any object type. These include folders, pages, URL references, data-entry wizard objects, reports, workflows, and files. Folders can contain any of the object types too including other folders. Pages, as you might expect, are the primary container object.

To create your own project, click on User Projects in the left-hand navigation panel, and click the New Item button at the top. Enter a name and description of your project.

Pages

IAB pages let you bring together the set of user interface controls you need to create the user experience desired. Pages are synonymous with JSP (as they are implemented with JSP technology) and can contain any number of controls. Figure 1 shows one of the sample pages open in the workspace. Besides the controls you'd find supported in HTML, Application Builder provides tree-views, paginating data grids, combo boxes, tab controls, RTF editor, menus, toolbars, and special controls for data management.

Create a new page by selecting the New Item button at the top of the Navigator panel on the left. Complete the pop-up form and the page is added as a JSP to the WAB web application. Adding controls is very simple: click on the control-type button along the top of the page in the workspace then click inside the body of the page where you want the control. Positioning and sizing are done in absolute terms so there should be no surprises.

Data Entry Wizard

The Data Entry Wizard (DEW) lets you quickly create a sequence of data-entry forms that will be presented to the user. The user can move forward or backward in the sequence filling in fields on the forms that may be bound to columns (from one or more tables) in the database. The DEW control handles generating SQL statements and interacting with the database. The wizard is functionally complete (client, J2EE server, and database interactions are managed) without any client- or server-side coding, although you can write event handlers if needed. Certain validation rules are available for selection as you assemble each form in the wizard. You can add as many forms as the user experience requires. When finished, the DEW object is saved then referenced on a page by adding a DEW control to the page and selecting the DEW object.

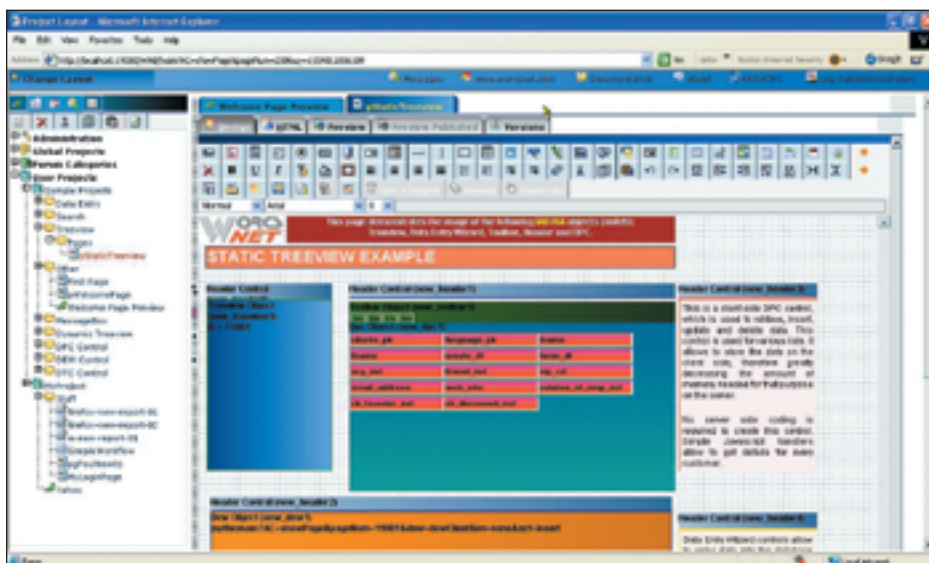


Figure 1

Data Presentation Control

The Data Presentation Control (DPC) is another powerful control that makes it easy to present and manipulate data in a page. The DPC is added directly to a page and configured in place. It uses a grid display style. You select the table and columns you want to show in the grid. For each column selected, you can control whether the column is visible, updatable, can be sorted, and which are the key columns. You also specify the column heading. Without any other work, the DPC can now be used to retrieve and view the specified data.

Data Transformation Control

The Data Transformation Control (DTC) is a non-visual control that lets client-side code interact with the database via the J2EE server. Since the control is non-visual, you simply write JavaScript to instantiate the DTC (also called DataStore), set query properties, and call the LoadData method to execute the query. You specify a callback function to the LoadData method to receive the data. Your method can do whatever you need to modify the DOM with the data.

Reports

Reports are very easy to set up. They are created as project artifacts in a folder. Once created, open the report up in the workspace. You must select the report style from one of the Free-form, Graph, Grid, Group, N-Up, or Tabular styles. You also can produce HTML, PDF, XML, tab-delimited, or CSV formats. The tool produces a canonical form for the XML output based on the query output. The graph style allows for various types (pie, line, bar, X-Y, etc.).

In Addition...

IAB Studio is also an execution environment and as such provides basic security and entitlement management functions. It supports user authentication and role-based authorization. Access to pages is granted to roles.

And if you'd rather develop in your favorite IDE, IAB Studio comes with an Eclipse plug-in so you can develop your IAB objects in familiar surroundings.

Workflow Builder

The Workflow Suite is a workflow model builder and execution engine included with IAB Studio. It lets you create a model of a workflow by connecting various types of tasks together in a flow. All tasks are implemented via a Java class. While IAB provides default implementations, they do nothing; you must provide an implementation to do anything.

There are tasks to execute your logic synchronously or asynchronously to the workflow. Other tasks provide classic flow control (if, loop), synchronization, HTTP URL retrieval, e-mail, FTP transfers, a simple delay, and user activity. Each kind of task defines parameters specific to it.

Concerns

While IAB Studio has a lot of great features, there are a few things about it that concern me.

I encountered a few minor glitches with the Web IDE. One time I had a pane pop up, seemingly out of nowhere. I had to log out to close it and couldn't reproduce it. The report designer shows three nested scrollbars; which makes navigation a bit challenging. You have to deal with scroll bars

on the browser window, scroll bars on the page, and scroll bars on the report control. Finally, there are mislabeled and duplicate buttons. Worcsnet is aware of the last two items and is planning to address them.

IAB doesn't connect to any external source code repository, though it has an internal versioning scheme. Since the latest version of all project artifacts are saved on the file system (under the WAB context root), it's possible to use an SCM outside the tool. However, this means that any files the SCM tool puts in its workspace (such as CVS and Subversion) will become part of the Web application file set and must be removed as part of promoting the application out of development.

IAB Studio acts as a development tool, framework, and runtime engine for the application. It stores metadata in the database regarding the application artifacts. It's designed to use a "develop and transplant" approach to deployment rather than the commonly used "checkout and build" approach. IAB Studio provides an export facility that's part of a promotion process to move the application from development to QA and production. The feature wasn't functional and the developer is expected to include the IAB metadata as part of the application database promotion scripts. A promotion/deployment model like this may also make some people question the veracity and repeatability of a CM process based on it. While this may not be an issue, it should be reviewed before committing to build an application that may be subject to SOX audits.

Conclusion

Despite these concerns, I'd recommend that developers looking to step into the rich Web application arena take a look at IAB Studio. It provides UI development features reminiscent of client/server tools like PowerBuilder and Visual Basic and includes basic workflow capabilities for your application. This feature set is too rich to let a few glitches stop you from trying it out. ■

About the Reviewer

Paul Kaiser is an application architect in New Jersey, where he develops Java-based e-commerce systems.
paulkaiser@yahoo.com

Information

WorcsNet, Inc.
4 Viewmark Drive, Richmond Hill
Ontario L4S 1C9 CANADA
www.worcsnet.com

Pricing: Structure varies; contact WorcsNet.

The Missing 'Discovery' Link to Successful Business Process Management

Process improvement based on user behavior, not estimates or anecdotes

WRITTEN BY **STUART BURRIS**

➤ Every organization is under pressure to deliver tangible business benefit through its IT projects. This point is illustrated by the fact that almost all IT projects are justified based on the ROI they will deliver. However, very few organizations follow up and review all projects based on the ROI they actually provided. While the reasons for this aren't clear, one compelling, well-documented statistic is that on average only one out of three IT projects will be successfully completed and deliver the ROI they promised.

If we dig deeper into understanding why two-thirds of projects fail, the good news is the root cause is not due to the deficiencies in technical capability. The tools the IT staff has on-hand become very robust and well understood over the last 30 years. Instead, the root cause of these failures is simply due to a lack of understanding of the business process that the project was supposed to improve. In spite of the fact that on average, 70% of an IT project's budget is dedicated to understanding and documenting the business process through requirements, analysis, design, and testing the process. Just like a sick patient at the hospital, you can't begin treatment without an accurate, effective diagnosis.

The traditional IT methodology for documenting the business process has several significant limitations. First, it never captures all of the nuances of the business process. The old saying "The devil is in the details" is true, and every process has a seemingly endless set of exceptions, variations, and deviations that only become apparent as they occur, hopefully in testing, invariably after the system is put into production.

Second, the process is often assumed to be repeatable across all users. Unfortunately, this is never the case. Best case, users just have different habits; worst case, different user groups have significant differences

stemming from the process's core requirements.

Third, the business process is a snapshot in time of the process. With the speed of business today, business processes are in a continuous state of transition. One key reason that shorter, small projects succeed more than longer projects is that the business process doesn't have time to evolve from the assumed, analyzed process.

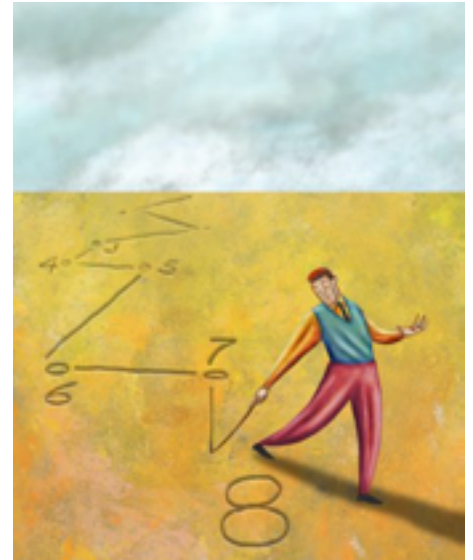
Most business processes today are technology-enabled and users leave electronic footprints of the process through the IT assets supporting the process. By examining the footprints the users leave as they execute their process, the business process can be automatically discovered and documented in near real-time.

Business Process Discovery (BPD) is an emerging field that discovers the business processes based on examining the activities of the users on the IT assets that support the business processes. Coming at the business process "bottom-up" from the detailed facts of instances of the process provides a detailed depiction of the business process, complete with all the nuances of that process, including detailed statistical information on how often different variations of the process are executed, how long it takes, what data conditions give rise to process variations, what variations there are between different users or groups, etc. And,

since BPD is an automated process, the business process can also be kept updated on a perpetual basis.

To illustrate BPD, consider a mainframe-enabled business process of order management. Order management in itself is a fairly large and ambiguous process; there are multiple aspects and organizations that could be involved ranging from order entry, call centers, credit management, etc. A traditional approach would interview representatives from each area and paint a process map that everyone could agree on. Alternatively, with BPD we'd watch how each user in the organization interacted with the mainframe and let the business process emerge from the data.

If we were to create a session file of every keystroke and every screen for every user in the organization and weave those files together into a process map we'd be able to see exactly how users are interacting with the system to achieve the process. We'd see what screens transaction users are using, how often they are using them, and how consistent the process is between users. Out of this picture, we'd quickly see the different sub-processes of order entry, call center, and credit check. We'd see the various data conditions that trigger different process points, e.g., order greater than credit limit creates a credit approval process. And, perhaps most importantly, we'd see the wall



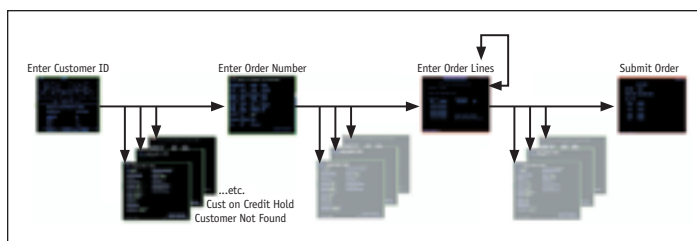


Figure 1

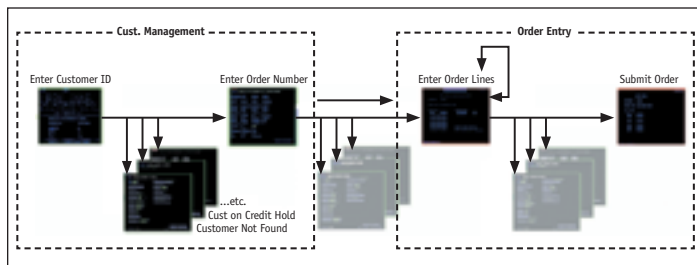


Figure 2

clock time and the process errors that you can't get from the interview process. Those are the places in the process that require the most user time, where users are struggling with the existing system.

The BPD approach provides a near real-time, complete, and accurate view of the existing business process. This fundamentally changes the economics of IT projects, no longer is 70% of the budget required for analysis and design. This leads to shorter, cheaper projects.

This also leads to a different way to approach business process improvement. The traditional approach of an enormous, risky business process re-engineering effort requires that extensive work be done on process mapping, process analysis, and making large one-time changes. BPD, however, creates an environment where it's natural to make smaller incremental process changes that yield immediate return, and repeat the process in a continuous improvement cycle. This creates an environment where the business users are continuously seeing benefits and most IT spending is on creating benefit, not on documenting business processes.

Traditionally, small incremental projects were difficult to execute not just because of the challenges of traditional process analysis but also because of the inflexible IT architectures that made it difficult to make incremental improvements. Service Oriented Architectures (SOA), however, are fundamentally changing that inflexible paradigm. SOA lets organizations rapidly adapt and incrementally improve their processes. This architectural change in the way IT systems are organized and deployed removes the technological constraints to an iterative, incremental improvement cycle.

However, the key to success with SOA is to be able to effectively define the appropriate granularity of the business services. BPD complements a SOA approach to solve this problem by having services emerge from this bottom-up analysis of the business processes and implementing those services that are required incrementally as part of process improvement initiatives.

Having a near real-time view of your current business processes yields dramatic returns beyond improving application development. Once the typical business process is understood, being able to monitor that business process for deviations introduces a proactive business process monitor that has immediate applicability for IT operations management as well as compliance and security. A

self-documenting business process allows IT staff and business users to be on the same page and help-desk support staff can immediately understand the business process or the deviation from the business process of a specific user. This allows a proactive investigation and response to the deviation, whether it's a user-training issue, systems-availability issue, or just an unusual business situation driving the workflow.

Perhaps the best thing about BPD is that getting started is easy and extremely low risk. The promise and benefits of BPD is to be able to derive business processes quickly, easily and automatically based on an examination of business user behavior. In our experience, we find that in a couple of weeks, we can target several business process improvement opportunities based on the detailed process maps that our BPD tool provides. This creates quantified process improvement opportunity based not on estimates or anecdotes, but on the solid facts of user behavior. ■

About the Author

As chief technology officer, Stuart Burris is responsible for the overall management of OpenConnect as well as its technology vision and strategy. Stuart joined OpenConnect in 1990 and has held a variety of research and product development roles, most recently as vice-president, research and development. During his tenure at OpenConnect, he has been instrumental in the development of the architecture for the company's industry-leading mainframe-to-Web products used by thousands of companies worldwide. Prior to OpenConnect, Stuart served as vice-president of technical services at Smartech Systems, Inc. Before that, he held a variety of development positions, including vice-president of R&D at Mtech.

SOA WSJ Advertiser Index

| Advertising Partner | Web Site URL | Phone # | Page |
|-------------------------|---------------------------------------|-----------------------|-------|
| ACTIVE ENDPOINTS | ACTIVEPEL.ORG/SOA | | 4 |
| AJAXWORLD | WWW.AJAXWORLDXPO.COM | 201-802-3022 | 44-45 |
| ALTOVA | WWW.ALTOVA.COM | 203-929-9400 | 2, 14 |
| CROSSCHECK NETWORKS | WWW.CROSSCHECKNET.COM | 888 276 7725 | 17 |
| FIORANO | HTTP:WWW.FIORANO.COM/DOWNLOADS | | 11 |
| FORUM SYSTEMS | WWW.FORUMSYSTEMS.COM | 801-313-4400 | 51 |
| IBM | IBM.COM/TAKEBACKCONTROL/SOA | | 6-7 |
| METALLECT | WWW.METALLECT.COM | 972-801-4350 | 23 |
| PARASOFT | WWW.PARASOFT.COM/WSJMAGAZINE | 888-305-0041 (X-3501) | 52 |
| ROGUEWAVE | WWW.ROGUEWAVE.COM/DEVELOPER/DOWNLOADS | | 31 |
| SOA WEBSERVICES JOURNAL | WWW.WSJ2.COM | 1-888-303-5252 | 39 |
| WEB AGE SOLUTIONS | WWW.WEBAGESOLUTIONS.COM | 1-877-517-654 | 19 |
| XENOS | WWW.XENOS.COM/VAN | 1-888-242-0695 | 25 |

General Conditions: The Publisher reserves the right to refuse any advertising not meeting the standards that are set to protect the high editorial quality of. All advertising is subject to approval by the Publisher. The Publisher assumes no liability for any costs or damages incurred if for any reason the Publisher fails to publish an advertisement. In no event shall the Publisher be liable for any costs or damages in excess of the cost of the advertisement as a result of a mistake in the advertisement or for any other reason. The Advertiser is fully responsible for all financial liability and terms of the contract executed by the agents or agencies who are acting on behalf of the Advertiser. Conditions set in this document (except the rates) are subject to change by the Publisher without notice. No conditions other than those set forth in this "General Conditions Document" shall be binding upon the Publisher. Advertisers (and their agencies) are fully responsible for the content of their advertisements printed in ColdFusion Developer's Journal. Advertisements are to be printed at the discretion of the Publisher. This discretion includes the positioning of the advertisement, except for "preferred positions" described in the rate table. Cancellations and changes to advertisements must be made in writing before the closing date. "Publisher" in this "General Conditions Document" refers to SYS-CON Publications, Inc. This index is provided as an additional service to our readers. The publisher does not assume any liability for errors or omissions.

Rich Internet Applications: AJAX,

www.AjaxWorldExpo.com

AJAXWORLD™ **CONFERENCE & EXPO**

SANTA CLARA SILICON VALLEY

**SYS-CON Events is proud to announce the first-ever
AjaxWorld Conference & Expo 2006!**

**The world-beating Conference program will provide developers and IT managers alike
with comprehensive information and insight into the biggest paradigm shift in website design,
development, and deployment since the invention of the World Wide Web itself a decade ago.**

The terms on everyone's lips this year include "AJAX," "Web 2.0" and "Rich Internet Applications." All of these themes play an integral role at AjaxWorld. So, anyone involved with business-critical web applications that recognize the importance of the user experience needs to attend this uniquely timely conference – especially the web designers and developers building those experiences, and those who manage them.

CALL FOR PAPERS NOW OPEN!

We are interested in receiving original speaking proposals for this event from i-Technology professionals. Speakers will be chosen from the co-existing worlds of both commercial software and open source. Delegates will be interested learn about a wide range of RIA topics that can help them achieve business value.

Flash, Web 2.0 and Beyond...

REGISTER TODAY AND SAVE!

→ **October 3-4, 2006**

→ **Santa Clara Convention Center**
Hyatt Regency Silicon Valley
Santa Clara, CA

→ **To Register**
Call 201-802-3020 or
Visit www.AjaxWorldExpo.com

→ **May 7-8, 2007**
First International AjaxWorld Europe
Amsterdam, Netherlands

“Over the two information-packed days, delegates will receive four days’ worth of education!”

Early Bird*

(Register Before August 31, 2006)

\$1,495**

See website or call for group discounts

Special Discounts*

(Register a Second Person)

\$1,395**

See website or call for group discounts

(5 Delegates from same Company)

\$1,295/ea.**

See website or call for group discounts

On-Demand Online Access

(Any Event)

\$695

*Golden Pass access includes Breakfast, Lunch and Coffee Breaks, Conference T-Shirt, Collectible Lap-Top Bag and Complete On-Demand Archives of sessions in 7 DVDs!

**OFFER SUBJECT TO CHANGE WITHOUT NOTICE.
PLEASE SEE WEBSITE FOR UP-TO-DATE PRICING

“It Was The Best AJAX Education Opportunity Anywhere in the World!” —John Hamilton

Topics Include...

Themes:

- > Improving Web-based Customer Interaction
- > AJAX for the Enterprise
- > RIA Best Practices
- > Web 2.0 – Why Does It Matter?
- > Emerging Standards
- > Open Source RIA Libraries
- > Leveraging Streaming Video

Technologies:

- > AJAX
- > The Flash Platform
- > The Flex 2 Framework & Flex Builder 2
- > Microsoft's approaches: ASP.NET, Atlas, XAML with Avalon
- > JackBe, openLaszlo
- > JavaServer Faces and AJAX
- > Nexaweb
- > TIBCO General Interface

Verticals:

- > Education
- > Transport
- > Retail
- > Entertainment
- > Financial Sector
- > Homeland Security

GROUP DISCOUNTS AVAILABLE:
— 5 Delegates from Same Company —
for only \$995 (each)
— Register a Second Person —
for only \$1195

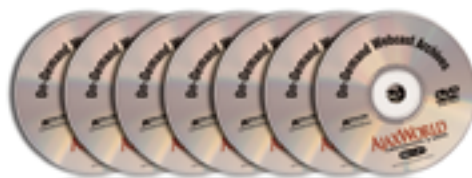
**Hurry! Limited Seating
This Conference Will Sell-Out!**



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

Receive **FREE**
WebCast Archives
of Entire Conference!

The best news for this year's conference delegates is that your "Golden Pass" registration now gives you full access to all 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

HYATT
HYATT REGENCY SILICON VALLEY



SYSCON
EVENTS

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

VISIT WWW.AJAXWORLDDEXPO.COM FOR THE MOST COMPLETE UP-TO-DATE INFORMATION



Commission Junction Announces Web Services Offering

(Westlake Village, CA) - ValueClick, Inc.'s, Commission Junction, a global provider of advanced affiliate marketing solutions, has announced the release of its Web services for advertisers and publishers. By offering a comprehensive Web services solution, Commission Junction is enabling its clients and third parties to develop their own applications to create unique products or consumer experiences in the CJ Marketplace. Advertisers can now offer publishers enhanced access to their product catalog data feeds, enabling publishers to present the most up-to-date product information to their visitors in the most desirable manner. In addition, advertisers benefit from the ability to customize affiliate program sign-up and login areas, allowing for the creation of a truly branded experience for the publishers in their affiliate marketing programs. Publisher functionality includes direct access to advertisers' Product Catalog data in real time and the ability to perform searches based on keyword, UPC, manufacturer, model number, advertiser, SKU and more.

www.valueclick.com



StrikeIron and Xignite Expand Partnership to Resell StrikeIron's OnDemand Web Services for Excel

(Research Triangle Park, NC) -StrikeIron Inc., provider of the Web Services Marketplace, has announced that it has partnered with Xignite which will resell StrikeIron's OnDemand Web Services for Excel. As part of the agreement, Xignite will use this Microsoft Office add-in to create live customizable Excel workbooks connecting to Xignite Financial Web services. These workbooks will be available to integrate with any financial or Web application in order to build customized, real-time applications quickly and cost-efficiently. In addition, the OnDemand Web Services for Excel workbooks will enable users to create compelling Service-Oriented Architecture (SOA)-driven applications to leverage the existing investments organizations are making in SOA infrastructure technology.

www.xignite.com www.strikeiron.com



Thailand and IBM to Create Nationwide SOA Center of Excellence

(Bangkok, Thailand) - Thailand's Office of Computer Clustering Promotion (CCP), under the National Science and Technology Development Agency (NSTDA), has announced it will create an SOA Excellence Center with IBM. The SOA Excellence Center will train and develop Thailand's IT resources around service-oriented architecture (SOA), a way of reusing a company's existing technology to more closely align with business goals helping to result in greater efficiencies, cost savings and productivity.

The new SOA Excellence Center will be located at the Thailand Science Park. The Center's mission is to develop skilled IT resources in Thailand in two tracks:

- **Academic Initiative Track:** The establishment of an IBM Academic Initiative to enable the IT workforce to be competent in high value SOA skills and related IBM's industry-leading WebSphere software, using CCP and IBM's curriculum and courseware.
- **DeveloperWorks Track:** The establishment of an SOA Community of local IT developers to assist the local IT industry to adopt SOA technology and to nurture local industries to develop innovative SOA-based products, applications and services. IBM's software will be used where applicable at the Center.

www.ibm.com/soa



Ubiquity Software Announces Enhancements to Its SIP Application Server and SOOF SOA Architecture

(Boston) - Ubiquity Software, creators of the standards-based SIP (Session Initiation Protocol) deployment platform, has announced the release of version 7.1 of its Session Initiation Protocol (SIP) Application Server (SIP A/S), which provides enhancements for service providers running SIP applications in a production environment. In conjunction with SIP A/S 7.1, Ubiquity has also announced enhancements to its standards-based SOOF Service Oriented Architecture (SOA) with the introduction of Open Web Services, a set of SOOF Service Components that provide support for Parlay X. Open Web Services Add Pack 1.0. is a new set of pre-built SOA service components that deliver capabilities defined by the ETSI Parlay X 2.0 specification, including: Third Party Call, Call Notification, Call Handling, Audio Call, and Multimedia Conference. With Ubiquity's SOOF SOA and Open Web Services, customers and application developers can now create advanced SIP applications with minimal knowledge of SIP and the underlying protocol, and can extend existing applications with the addition of real-time communications features like Instant Messaging (IM), click-to-call, click-to-conference, and more.

www.ubiquitysoftware.com



Fujitsu FlexFrame for mySAP Business Suite Supports Full SOA Deployment for SAP Applications

(Sunnyvale, CA) - Fujitsu Computer Systems Corporation has announced the availability of FlexFrame for mySAP Business Suite 3.2B, an integrated solution using virtualization technology that enables companies to run mySAP applications using a true service-oriented architecture (SOA). Companies are moving to SOAs in order to be more responsive to customers and react more quickly to changing business needs. Achieving the full benefits of shared services requires both SAP Enterprise Services Architecture (ESA), SAP's blueprint for putting SOA to work, and a flexible hardware platform capable of responding rapidly to dynamic business changes. With the latest version of FlexFrame for mySAP Business Suite, companies can achieve a complete, top-to-bottom SOA deployment of mySAP applications based on SAP's Adaptive Computing Initiative, resulting in more agility and lower total cost of ownership (TCO)

www.fujitsu.com



webMethods to Acquire Infravio

(Fairfax, VA) - webMethods, Inc., a business integration and optimization software company, has announced that it has entered into a definitive agreement to acquire privately held Infravio, Inc. for approximately \$38 million in cash. The deal is expected to close during the month of September 2006 and it is anticipated to be accretive to webMethods' earnings per share (EPS) beginning with the quarter ending March 31, 2007.

Infravio is a pure-play provider of service-oriented architecture (SOA) registry and governance solutions. SOA governance enforces the policies and procedures that determine how developers, IT staff, and business users leverage and utilize services throughout the entire SOA lifecycle, from initial design and run-time to ongoing changes in the system. SOA governance creates alignment and enables collaboration across these disparate groups while allowing each participant to maintain their own distinct view of services and policies.

www.webmethods.com



Acsera Changes Company Name to ClearApp, Inc.; Launches ClearApp QuickVision 6.0

(Sunnyvale, CA) - Acsera Corporation, a provider of model-driven application performance management (APM) solutions for portal, J2EE and SOA applications, has announced its name change to ClearApp, Inc., effective immediately. The company name change is accompanied by an updated logo and a new Website at <http://www.clearapp.com> redesigned to reflect the speed, visibility and control ClearApp delivers to customers managing complex, composite application environments. In addition, ClearApp announced today the launch of a new version of its flagship APM solution, now called ClearApp QuickVision 6.0 (formerly Acsera Manager).

www.clearapp.com



BEA Announces SOA for Executives Services

(San Jose, CA) - BEA Systems, a provider of enterprise infrastructure software, has announced the availability of Service-Oriented Architecture (SOA) for Executives, the first BEA suite of SOA services for senior IT executives. This suite is designed to help empower executives to invest in SOA and lead change within their organizations by conveying the business value of SOA through BEA's sensible approach. The new SOA for Executives suite of services is a set of professional services that are built for executives from the ground up and based on BEA's experience with SOA at Global 2400 companies. According to a recent IDG Research Services survey of 500 IT and business professionals, the top inhibitors to SOA deployment are a lack of skill or training, building a new governance model, internal IT and organization barriers, and defining metrics to measure SOA.

www.bea.com



SAP Helps Customers and Partners Map Easier Path to Enterprise SOA

(Las Vegas) - SAP AG has announced the availability of SAP Discovery System software for enterprise service-oriented architecture (enterprise SOA), designed to help customers and partners map their paths to a successful adoption of enterprise SOA. With SAP Discovery System, developers and enterprise architects have a clear risk-free first step in experimenting with enterprise SOA, enabling them to test-drive the simplicity and flexibility of composing new business processes using enterprise services in a standalone SOA environment. This release delivers a pre-configured SOA landscape, giving customers and partners immediate access to the latest software and tools available from SAP as well as a comprehensive set of sample business scenarios.

www.sap.com

Best Practices for Building SOA Applications

Seven Steps to SOA Adoption - Part Two: Rich GUIs, monitoring, security, and performance

WRITTEN BY DAVE SHAFFER

➤ This article is the second part of a two-part series covering best practices for building Service Oriented Architecture (SOA) applications. The following are the seven key steps for effective SOA adoption:

1. Create a portfolio of services
2. Define connectivity and messaging interfaces
3. Process orchestration, workflow, and rules
4. Rich user interfaces
5. Business activity monitoring
6. Security and management
7. Performance and scalability

In the first article, we described why adopting an SOA is valuable but can be difficult. We also looked in detail at the first three of the seven steps outlined above. In this article we'll focus on the final four steps and look at some "worst practices" — common errors in SOA design and how to avoid them.

Rich User Interfaces

We've seen several generations of UI evolution since the emergence of the Web as an application interface paradigm. Initially, HTTP and HTML provided many benefits for administrators through a thin-client approach, but users were faced with GUIs that were much more primitive than what could be done with thick-client interfaces. With the emergence of rich Asynchronous Java And XML (Ajax)-style interfaces, we are now seeing a truly mature thin-client paradigm.

However, developers often find the complex JavaScript code for user interfaces to be cumbersome, hard-to-debug, and repetitive. In this area, the emergence of Java Server Faces (JSF) frameworks that encapsulate rich dynamic GUI capabilities in reusable components has given developers

some new tools to make the development of rich Web GUIs easier.

As Web GUI paradigms evolved, developers were faced with more choices. In our first article, BPEL was discussed as the standard for business process orchestration, and GUI page flows are sometimes considered "orchestrated" interface components. However, BPEL is usually not the right abstraction for page flows. We see JSF and its predecessor, Struts, as being the best way to implement user interface control flow in the Java/J2EE world. BPEL is best for structured flows, but page flows are typically semi-structured or unstructured. Although BPEL is also particularly important when you need to maintain audit trails and when the process strictly controls the order of execution of activities, but GUI flows usually don't require these.

Of course, applications often connect their GUIs to business processes through human worklist interfaces, custom Web interfaces, and portals. BPEL's ability to support Web Services interfaces and transactional interfaces via adapters and WSIF bindings makes it easy to integrate J2EE GUIs and portals with BPEL processes. Standards like WS-Remote portlets and JSR-



168 mean that vendors can publish process portlets, such as a worklist editor, in a way that's easy for developers to integrate into a portal of their choice.

Business Activity Monitoring

A common complaint in organizations is that they have lots of data but not enough information. For example, we have a client who described his problem as having "14 terabytes of data but no unified view of our customer." One of the best ways to avoid this problem is to define key performance indicators (KPIs) as early in the SOA design process as possible. KPIs are pieces of information that the organization wants to track, such as the number of business transactions that are processed a day, the number of exceptions that are raised, and the amount of the time it takes to process each step. Because KPIs can change over time, the most effective approach for gathering this information is to instrument processes and IT events with "sensors" that monitor the business transactions. The events can then be fed to business activity monitoring (BAM) dashboards and custom reporting channels without requiring that process logic be changed.

Once the events are identified, correlated, aggregated, and fed to rich real-time dashboards, an organization achieves what we call the “fusion effect.” This occurs when actionable information informs an organization how to improve its processes, and its agile IT environment lets these changes be implemented efficiently.

Security and Management

Security has become increasingly critical as the perfect storm of information proliferation, regulatory change, and identity theft disclosures have come to pass. Industries such as healthcare and financial services require an unrelenting focus on security as information such as an individual's medical and financial data is passed over a network. Addressing these challenges is particularly complex in a heterogeneous and fast-changing technology environment. Conveniently (and not coincidentally), key standards such as WS-Security have emerged to enable the secure exchange of information between processes and services, even across different technology stacks such as J2EE and Microsoft .NET.

WS-Security specifically provides a standard mechanism for authentication and access control for services, as well as full or partial encryption of message data. WS-Security support is available in Microsoft .NET services, Open Source Web Services frameworks such as Apache Axis, and commercial J2EE toolkits such as Oracle, BEA, and IBM's application servers. It's easy to find information describing how this interoperability works. For example, Microsoft MVP Jesus Rodriguez has code examples on his blog demonstrating WS-Security interoperability between Microsoft WSE 3.0 and Oracle BPEL Process Manager (<http://weblogs.asp.net/gsusx/archive/2006/03/22/440881.aspx>). Likewise, Security Assertion Markup Language (SAML) provides a standard mechanism for role-based access control and federated identity. Standardizing on WS-Security and SAML for service interfaces gives an organization much more flexibility in its future technology choices and for secure Web Service interactions with trading partners.

It's also important to extract security requirements out of core services and clients and implement them in a policy-oriented fashion. This results in systems that are dynamic, secure, and auditable. Organizations implementing this approach are able to define external security policies and



Figure 1: Rich thin-client GUI-PeopleSoft real-time BAM dashboard

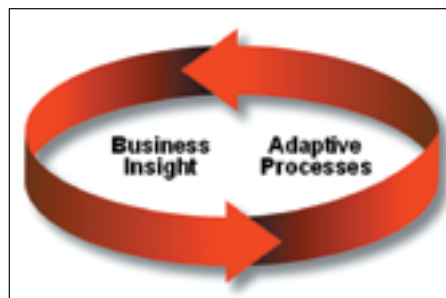


Figure 2: The fusion effect

change them dynamically, without needing to modify services or the clients that call them. This approach is supported by leading Web Services management (WSM) products.

Performance and Scalability

Once security policies are in place, the next step to effective SOA is to focus on the performance and scalability requirements in detail. As ever, the principle of “a stitch in time saves nine” applies. For example, we've seen project teams that used all asynchronous interfaces for their services because the toolkit they were using made that very easy to do. After developing sophisticated processes for handling registration for consumer credit services, the project team did stress tests late in the development lifecycle. They discovered that the overhead of the asynchronous interfaces, which required frequent persistence of the overall process, was such that their anticipated load could only be supported by an unaffordably



Figure 3: Securing services in a standard fashion

large number of CPUs. When such information is discovered so late in the process, the choices are bleak: either increase the budget significantly or re-engineer all the services to use different interfaces — which is a change that will propagate painfully throughout the project.

The best way to avoid this scenario is to do a performance POC early in the development process (even at the design stage) and get some real numbers regarding the size of the systems that are needed to achieve expected loads. By doing this during early prototyping and design stages, potential performance bottlenecks will be uncovered while there's still time to change key design decisions.

Another best practice is to choose carefully among synchronous and asynchronous service interfaces, standards such as WS-Addressing, and custom correlation mechanisms for correlating asynchronous messages. WS-Addressing provides a standard mechanism for correlating asynchro-

nous messages so that system A can send a request to system B, and system B can call back to system A when a response is ready. This kind of asynchronous interface does have a performance cost, but you gain reliability and flexibility because the two systems no longer have to be tightly coupled to each other. Of course, projects have been built on top of asynchronous message-oriented middleware such as IBM MQ Series, TIBCO, and JMS messaging for years. What's new is that the benefits of asynchronous interfaces are now available through standards such as WS-Addressing over protocols such as SOAP over HTTP so that such implementations can cross technology and vendor boundaries more easily.

When considering Web Services as an integration approach, people sometimes worry about XML as a performance bottleneck, and it can be when used inappropriately. However, in general, we don't believe that XML in and of itself presents performance overhead sufficient to rule it out, even for very large load requirements, especially given its many benefits. As when Java emerged to replace C and C++ as a preferred programming language, it takes a little time for design-time and runtime tools to evolve to optimal performance for the latest development approaches. We're now starting to see toolkits for XML processing. These toolkits, such as Oracle XDK, allow operations such as dehydration, XSLT transformations, and BPEL assign activities to be applied while the data remains in an optimized binary format. This avoids the most expensive part of XML processing — serialization and deserialization. For external gateway-style transformations or WS-Security support, hardware devices such as the one created by DataPower (recently acquired by IBM) and software tools such as Forum Vantage XML Accelerator can be useful.

However, there are ways to misuse XML. For example, passing very large documents between services via SOAP requires large amounts of bandwidth, processing time, and memory to serialize and deserialize the documents, even if you can minimize these steps. A preferred approach is to store the documents in a central location (a file system, database, or document management system) and then pass references to the document.

When processing very large documents comprised a large number of records, batching and debatching the documents can also make a huge difference in perfor-

mance and memory use. The tricky part of batch processing large documents is often coordinating the processing. For example, when a master process has to do some post-processing after all the child records have been processed. This is an area where developers can look to the commercial toolkits they're working with to provide explicit support. Finally, another useful strategy for processing large documents is to stream them. We recently found that a streaming implementation increased the size of documents we could exchange and perform XSLT transforms on in our BPEL engine from 10MB to more than 1GB, without encountering out-of-memory errors.

Worst Practices

There is often more to learn from negative experiences than positive ones. Here are ways in which SOA development projects can go awry and the lessons that can be learned from other people's mistakes.

Choosing the Wrong Language or Abstraction

We occasionally see developers using Web Services and BPEL instead of a programming language such as Java or C# for large iterative computational loops. Some of the issues with a Web Services orchestration language for this kind of problem include the fact that audit trails will get very long when a process has thousands of activities — which could easily be the case with a large number of loop iterations. There can also be a problem with transaction timeouts. Because computational loops are typically synchronous, a BPEL engine would try to fit them all into a single transaction for efficiency purposes. However, most application servers will timeout synchronous J2EE transactions after 30 or 60 seconds, a time period that could easily be exceeded with many iterations invoking Web Services in each loop. A much better approach is to do all the calculations in pure Java and then invoke that external Java logic from a BPEL process, ideally via a Java or EJB WSIF binding for optimal performance.

Embedding Security in Components or Services

Embedding security implementations in each component or service that needs to be secured makes the policies more difficult to change. But it also decreases the chance that a service will be reusable, because clients that require a different security

policy will necessitate that the service itself be modified. These clients will have to use copies of the service, entailing additional management issues over time as the number of versions of a service increases. We recommend externalizing security policies so they can be changed dynamically.

Developing Large Projects with a Bottom-Up, Big Bang Approach

A large initiative that touches or replaces many layers of IT infrastructure can be very valuable; however, implementing such a project with a pure bottom-up approach all at once is tremendously risky. The main problem is that the business benefits of such a project (which get a lot of attention from the business people funding such a project) are very much back-end loaded. The duration of such projects can be measured in months or years — but they can also often be measured in "CIOs." It's not uncommon to see a bottom-up overhaul of IT infrastructure try a CEO's patience to the point of replacing the CIO who initiated the project because of all the money and time spent without any perceivable value.

A much better approach is a stepping stone project model with rapidly iterating development cycles. In this model, each individual project brings an improvement in the IT infrastructure as well as a business benefit that can be used to measure the ROI of the overall initiative.

Summary

SOA can bring great value to an enterprise, but there are many potential pitfalls and hurdles along the way. Because SOA is now moving into the mainstream, we strongly believe that organizations adopting it should think through why they are doing it and how to achieve their goals through the promise of SOA, while minimizing the risk. Our hope is that these articles will impart some useful tidbits of information that we have gained from our experience with many SOA projects and spur an ongoing discussion in the IT community regarding how best to adopt and implement SOA. ■

About the Author

Dave Shaffer is Sr. Director of Product Management at Oracle for the Oracle SOA Suite, overseeing BPEL, BAM, ESB and other products. Prior to Oracle, he has held consulting, product management and software development roles at a wide-range of technology companies including Collaxa, Apple Computer, NeXT Software and Integrated Computer Solutions.
david.shaffer@oracle.com

SOA
MAKE YOUR ^ 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

W W W . F O R U M S Y S T E M S . C O M



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™

Parasoft Corporation, 101 E. Huntington Dr., Monrovia, CA 91016. For information, call 888-305-0041 (x-3501). Copyright ©2006 Parasoft Corporation. All rights reserved.
All Parasoft product names are trademarks or registered trademarks of Parasoft Corporation in the United States and other countries. All other marks are the property of their respective owners.