SYS-CON

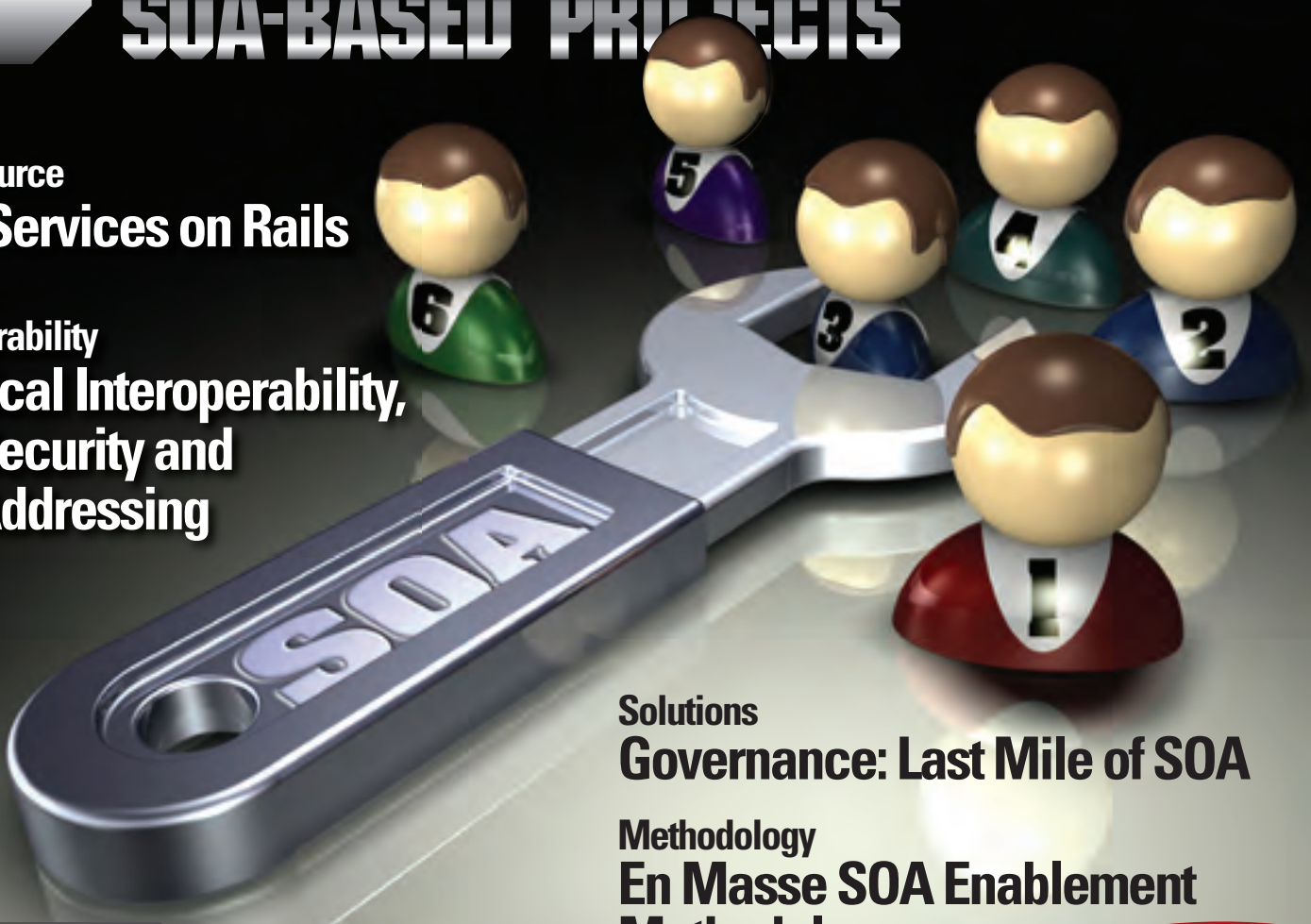
THE WORLD'S LEADING MAGAZINE DEDICATED TO WEB SERVICES TECHNOLOGIES

Visit us at www.WebServices.SYS-CON.com

SOA Web Services JOURNAL

OCTOBER 2006 / VOLUME: 6 ISSUE 10

6 BASIC RULES FOR SECURING SOA-BASED PROJECTS



Open Source
Web Services on Rails

Interoperability
Practical Interoperability, WS-Security and WS-Addressing

Solutions
Governance: Last Mile of SOA

Methodology
En Masse SOA Enablement Methodology Distilled

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

ALTOVA® xmlspy® 2006

Bring your development plans to light

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

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

XMLSpy is also available as part of the award-winning Altova XML Suite.

ALTOVA®

SOA Web Services JOURNAL
OCTOBER 2006 / VOLUME: 6 ISSUE 10

TABLE OF CONTENTS

Visit us online at WebServices.SYS-CON.com

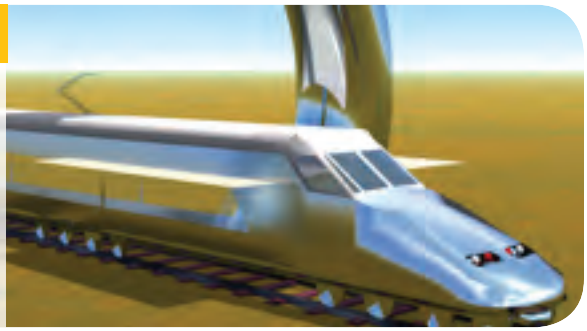
Inside This Issue

BEST-PRACTICES

Securing SOA based Projects

Common Sense SOA Security

OPEN SOURCE



Web Services on Rails

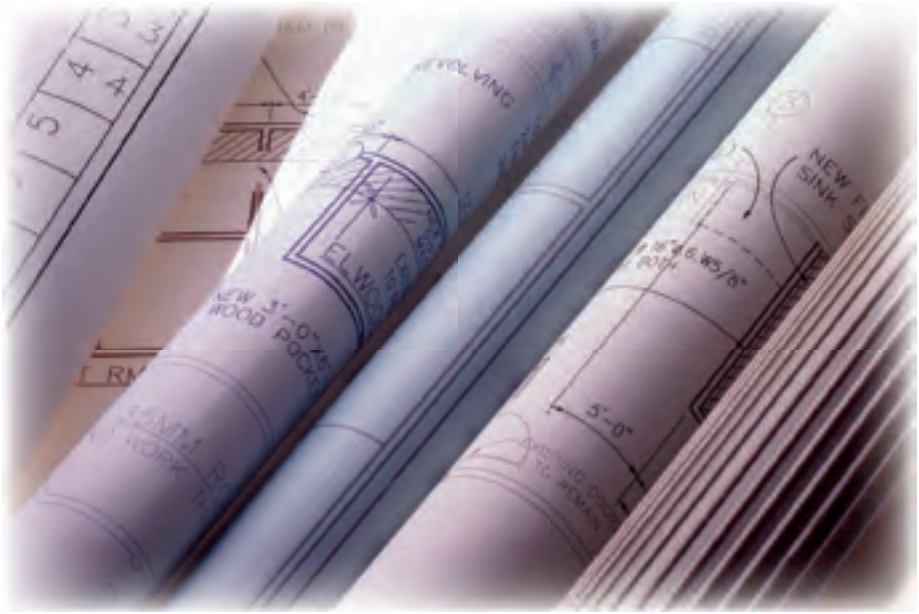
A Web 2.0 perspective for SOA

INTEROPERABILITY

Web Services and SOA

Practical Interoperability, WS-Security and WS-Addressing

Get Started with BPEL 2.0



Build next-generation SOA applications with the leader in BPEL technologies

Download BPEL tooling & server software today

activebpel.org/soa

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

activeBPEL

active endpoints

SOA Web Services JOURNAL
www.WebServices.SYS-CON.com

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.
577 Chestnut Ridge Road, Woodcliff Lake, NJ 07677

Learning the Language

WRITTEN BY **SEAN RHODY**

In some ways, an industry is like a country. It has its citizens, the corporations, its own particular nuances that serve to make one industry just that much different from another (think insurance and financial services) as to be distinct, and it has a language.

The language of an industry is unique. Words exist in some industries for concepts that have no relevance or relationship to other industries. Just like countries, who have citizens who speak multiple languages, there are groups of companies that offer support, such as IT outsourcing, to companies in any industry. To do so, they have to speak the language of the particular industry they service.

In the IT world, there has long been recognition that in order to support an industry effectively, there needs to be the technology equivalent of a language. In some cases this refers to the processes that are unique to an industry, like verbs in a sentence. In other cases it reflects the unique nature of the commodity or product that is the focus of an industry, such as a financial trade, or a container ship full of cars. Regardless, the understanding that a specialized vocabulary exists on an industry level came very early on in the natural evolution of computer systems.

That didn't mean that it was simple or straightforward to document the language of an industry. Simple things became arguing points, such as whether a customer name was one field or two, or three, or even four. As trivial as it sounds, getting two different companies to agree on data formats is a complicated task, even when the relative number of choices is limited.

Still more daunting is the problem of process. While members of an industry in general make the same product or provide the same service, they rarely agree wholeheartedly on how they do that. Then they have to reconcile what they do with other impacts – like how they do accounting or how they interact with their partners. One company's manufacturing process may be different from any other company's, except for the small part of it where they actually make the commodity or provide the service. Think of an insurance agent. Some work for a company; some are licensed agents. In the end, both write policies, but the process they use may be very different.

The language of an industry is really about communication between members. A company that has no external partners can ignore the language, and companies that dominate an industry can rewrite a language (think WalMart), but, otherwise, companies have to coexist within an industry. While they may compete, they also do cooperate with one another and to do that effectively and electronically they need to speak a common language, one that understands process, data, and the unique facets of the industry.

Web services and service-oriented architecture have made it easier for companies to interact with one another and exchange data. The underlying tool for that is XML. XML is the underlying structure of the language companies use to communicate electronically with one another. The transformation technologies that support XML make it possible for companies with different data formats and differing process understandings to be able to communicate effectively on a business-to-business level with one another.

The logical conclusion was to try to standardize the language across the industry so that there could be a common ground for understanding and interaction. There have been a number of different approaches, such as RosettaNet and eBXML, that have attempted to codify general principles and practices across industries. Other languages, such as FML and ACORD, have targeted specific industries and attempted with some degree of success to become the de facto language.

This issue focuses on such languages and their use in SOA and Web services implementations. We hope you find it useful in working with the language of your particular industry. ■

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*

Service Synchronicity

WRITTEN BY **AJIT SAGAR**

The entire premise behind the Web services paradigm is enabling access to loosely coupled services via the Web. In essence, Web services are based on a synchronous request-response type interaction. On the other hand, a client's interaction with a Web service can be synchronous or asynchronous.

When application functionality is exposed in the form of Web services, one of the dangers in designing the architecture of the service layer is that the designers tend to think from the perspective of service consumption only. In other words, the focus is on "who is going to call me" and not on "whom am I going to call?" Granted that the design of services needs to target the consumer, but when dealing with the asynchronous paradigm, remember that there is also the "callback" to be accounted for. Since Web services are primarily used for integrating heterogeneous platforms, it is important to also consider the constraints of the client.

Please allow me to elaborate. With synchronous services, clients invoke a request on a service and then suspend their processing while they wait for a response. With asynchronous services (also dubbed document-style Web services), clients initiate a request to a service and then resume their processing without waiting for a response. The service handles the client request and returns a response at some later point, at which time the client retrieves the response and proceeds with its processing.

Let us consider this from the design of the client. The client invokes an asynchronous Web service through the interface method exposed by the service. The result of the Web service can be obtained in one of the following ways:

- The client polls for the response. This means that the client calls the Web service interface for a response to the original request at periodic intervals.
- The service calls back the client's service with the result

Of the two approaches, the first one is non-intrusive, in the sense that the client interface doesn't need additional interfaces that are exposed to other applications. An independent "polling module" can be developed that is responsible for getting the results of the call back from the service.

Incorporating the second approach is a little more complicated. The assumption in this case is that the client is modifiable to the extent that it can be "Web service–enabled." This also leads to a greater coupling between the service interface and the client – the interface developed for this protocol is developed for the sole purpose of being able to complete the circuit. In other words, the service interface developed on the client is for the purpose of completing the functionality of the service that was developed for the application. However, this approach has the advantage of being true to the "event listener" paradigm. While designing the interfaces for the Web service, care should be taken to give the client's architecture due consideration. Several aspects need to be considered, some of which are:

- **Technology**: Is the client in a state to be Web service enabled? As mentioned earlier, the primary purpose of exposing a Web service interface is to integrate heterogeneous platforms. The client may be a legacy application, in which case, development of a Web service interface on the client may not be feasible.
- **Ownership**: Even if the client is technologically mature, it may not fall under the purview of the group or organization that is developing the Web service interface.
- **Testing**: A stable client that now exposes a Web service interface will have to be retested for "callability." This could have substantial impact.
- **Ubiquity**: Depending on the client to assume a part of the responsibility for the protocol makes the service less ubiquitous. Each client that calls the service will now have to be Web service enabled.

Although Web services offers alternatives for implementing the "synchronicity of service," architecting the solution requires careful consideration of the impact the decision will have on the development and incorporation of the paradigm, as well as the longevity of the developed service in a service-oriented architecture. ■

About the Author

Ajit Sagar is a principal architect with Infosys Technologies, Ltd., a global consulting and IT services company. Ajit has been working with Java since 1997, and has more than 15 years experience in the IT industry. During this tenure, he's been a programmer, lead architect, director of engineering, and product manager for companies from 15 to 25,000 people in size. Ajit has served as JDJ's J2EE editor, was the founding editor of XML Journal, and has been a frequent speaker at SYS-CON's Web Services Edge series of conferences, JavaOne, and international conference. He has published more than 125 articles. ajitsagar@sys-con.com

SOAWebServices JOURNAL

www.WebServices.SYS-CON.com

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
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

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

Facing a few barriers on the road to SOA?

JackBe's Rich Enterprise Application (REA) platform clears the road to SOA business benefits.

There's an abundance of products and vendors to help you *create* your SOA. Now, *consume* those SOA services with JackBe REAs to achieve the business productivity and results that led you to SOA in the first place. Our new REA platform combines the power of Ajax and SOA with reliable, secure communications to extend your SOA directly into powerful business applications.

A fully visual IDE reduces manual coding and accelerates the development process. And our lightweight, vendor-neutral platform easily integrates with existing middleware and services while maintaining server-side governance and control--unlike products that leave governance to the browser or create middleware platform dependencies.

Join over 35 industry leaders like Citigroup, the U.S. Defense Intelligence Agency, Sears, Tupperware, and Forbes who are already optimizing their business activity with JackBe solutions. Call or visit our website—let us help you remove the barriers on the road to achieving real business value from your SOA investment.

JACKBE™
Optimizing Business Activity

Web: www.jackbe.com
Phone: (240) 744-7620
Email: info@jackbe.com

Modern System Management

A service-oriented approach

WRITTEN BY **CRAIG WASSENBERG**

Since the beginning of the information technology era, IT managers have implemented various systems management applications, ranging from enterprise frameworks to open source products, in an attempt to manage the various servers, storage, and network devices currently found in their data centers. For the most part, these costly solutions have met little success because most of these devices cannot communicate with each other, as they are based on different standards and protocols depending on the vendor that created them.

Just as a Rosetta Stone was needed to translate hieroglyphics into a common language, the systems management market needs a way to unify these technologies and attain the holy grail of the automated data center. To achieve this long sought-after vision, a new platform is required that starts before servers, storage, or networking equipment are delivered to the data center. The solution, rather than introducing yet another management framework or technology, would unite all existing technologies under a common infrastructure that could glue any device to any management application.

Service Oriented Architectures are rapidly gaining in popularity for addressing the challenges of data center complexity and heterogeneity, but to date, they have not been widely used to address the infrastructure issues such as systems management. In this paper, we will look at how a service-oriented approach 1) can be applied to the issues surrounding systems management, 2) can help provide solutions that are powerful, flexible, and highly tailored to leverage existing "best-of-breed" solutions, and 3) can bring, for the first time, a solution that can unite disparate systems and systems management applications into one common infrastructure.

The Growing Challenges Faced by Data Center Managers

The life of a data center manager in most organizations is hard and destined to get harder unless some things change. According to Gartner, 85% of data center budgets are spent on operating expenses – mostly labor – and only 15% are spent on capital expenses. For most data centers, the labor is largely dedicated to simply "keeping the lights on," or ensuring that existing and newly acquired systems continue to run and that service levels don't drop. More often than not, strategic projects designed to improve service levels or better drive business improvements fail to get implemented.

For some time now, data centers have been deploying an increasing number of commodity servers with the ratio of system administrators to servers decreasing, forcing each administrator to manage more servers. Furthermore, with the growing importance of Web and networking traffic, and the increased numbers of threats (SPAM, viruses, intrusions, denial of service attacks, etc.), data centers must resort to bringing in additional network-related appliances to handle jobs related to security, routing, load balancing, traffic shaping, fire walling, etc. These appliances all serve very useful purposes but significantly contribute to the increasing complexities of managing the data center.

The challenges faced by data center managers will continue to grow. If you consider what the introduction of Service Oriented Architectures (SOAs) bring, it's obvious that SOA's make creating powerful, distributed, and inter- and intra-company applications much easier. However, SOAs bring new challenges to applications that are loosely coupled and distributed, and parts of them may be behind different company's firewalls. Additionally, some of these loosely coupled components may be running on virtual machines and can potentially be migrated from one server to another.

Because of these challenges, IT administrators are being forced to do more with less: manage more devices and support software applications that have become more granular, more distributed, and partially hidden behind firewalls.

Although SOAs might be partially to blame for an increase in management difficulties, they also offer great promise toward providing useful ways to rescue data center managers from complexity problems.

Systems Management Software that Can't Manage Systems

Before describing how SOA approaches to systems management can help, it's worth taking a look at the current state of systems management today. This will help to illustrate how much an SOA-based approach to systems management can help.

In order to effectively manage a given device such as a computer, a router, a network appliance, etc., that device has to be a good "managee." Just as in everyday life, the best manager in the world can't be fully effective if the people or things he or she is trying to manage are inherently unmanageable or only poorly manageable. Many devices sold today simply have terrible management characteristics. Some devices only offer weak management interfaces, some respond only to a very limited set of management commands, some do not have programmatically accessible management interfaces, and some do not even respond correctly to the management commands they were presumably designed to comprehend. Many devices do not speak any of the standard or de facto standard management protocols like SNMP, IPMI, or WMI. Some devices can only be controlled or queried for status via proprietary means like a proprietary command line interface (CLI) or Web interface. Just as in every day life, a manager who has to manage several "entities" with whom he or she does not share any common languages and who might not be very responsive, even if there is a shared language, can't be very effective. In the systems management world, developers refer to devices or sets of devices that can't be managed in standard ways, but only in idiosyncratic ways as "silos of management." The more silos of management there are, the tougher the system administrator's job is with more to learn, more system differences to comprehend, and more complexity in the data center overall.

Generally, a data center manager can find management software that will perform certain limited but powerful functions. However, there are entire classes of management problems that can't be managed very well today at all. Imagine for a moment that you are a data center manager. Now imagine further that you have gotten over the initial shock of being overworked and underpaid when one of the managers of the group to whom you provide service asks you, 'Why is e-mail so slow today?' On the surface the question seems direct enough; however, you will quickly realize that the root cause for e-mail being slow could be any number of things, or a combination of them such as: routers malfunctioning, a virus invading the network, an overloaded e-mail server, or an attack on your network. The problem is a distributed system problem. Data center managers have gotten used to, or perhaps numb to, the fact that the bulk of systems management software on the market today is terrible at managing a distributed system environment.

Those familiar with SOA-based applications realize that the complexities of managing distributed systems are about to get much worse. Forget about the, 'Why is e-mail slow problem' and now ask yourself how you will answer next week's question, 'Why did so many Web transactions through our new SOA loosely coupled applications fail this afternoon?' And by the way, some of the machines that those loosely coupled SOA components were running on were virtual machines (VMs) which have been moved to different physical servers, different from the ones they were on this afternoon. The SOA management problem is in many ways harder than the e-mail problem since it is more distributed, and some of the system elements involved will be or could be more transient in nature.

Companies that want to effectively manage distributed systems today, whether they are e-mail, Web serving, or SOA-based applications, are in a bind. There are so many un-integrated silos of management that companies must either implement their own very expensive means to manage them, or they must simply rely on their human administrators to understand all the relationships between devices and software components and hope they can monitor for problems and troubleshoot them in acceptable timeframes. Either way, the effort required is extremely expensive requiring costly development or costly human labor. In practice, companies will typically adopt a hybrid approach. They will make use of the management products they have, "siloed" though they may be, and write some of their own management software, perhaps incorporating some good open source management components, and rely on their people to provide whatever additional intelligence that can't easily be automated.
Overall the situation is messy and expensive to fix.

SOA Management to the Rescue

SOAs are causing some of the problems, so it is only fitting that they be the ones to provide a significant part of the solution. In fact, this trend is already starting. Some companies and players in the systems management market are drawing on SOA approaches to improve the systems management picture. Both the DMTF and OASIS standards groups are working on Web services-based management standards called WS-MAN and WSDM respectively.

The basic idea behind the standards is simple: to create service-oriented management interfaces for hardware and software resources in the network. The implications for systems builders (ODMs, OEMs, ISVs, system integrators) are huge, as are the implications for data center administrators.

Let's take a look at how the situation should play out, based on what the standards groups are doing, what many systems management companies are doing, and according to what we learned in the world of SOA-based applications when standards started to emerge.

Fewer New Unmanageable or Poorly Manageable Resources

Developers of SOA applications have made giant (and once nearly unthinkable) strides toward creating software components that can be reused and readily integrated with other software components. Software from different companies can interoperate, and loosely coupled applications can span the globe if need be.

System builders are working to make their devices manageable via WS-MAN and WSDM. The management interfaces to these devices will be formally defined using WSDL, and access to those interfaces will be via standard Web services protocols like SOAP. Today's "Towers of Babel" and management silos will start to erode.

In some cases, existing "legacy" management devices will be upgradeable to support the new WS- interfaces. However, in many cases, it will be either impractical or impossible to upgrade such devices in the field. A key question then will be: 'Can anything be done to fully integrate the management of those devices under the SOA management paradigm?' The answer, of course, is yes.

New WS-Based Ways to Manage Legacy Resources

Developers of complex SOA-based applications often want to capture the value of legacy applications rather than rewrite them. Rewrites are typically just not feasible. Capturing the value of legacy apps and integrating them into the SOA world

SOAs are causing some of the problems, so it is only fitting that they be the ones to provide a significant part of the solution

is done through encapsulation: create a formal SOA interface for the legacy application, bind that interface to the application via the most suitable means, and thereby enable other newer SOA components to interact with the services now exposed by new SOA interfaces to the legacy software. Companies like BEA Systems, IBM, and Tibco Software provide enterprise application integration (EAI) products to perform these kinds of feats, although there are many cases where the work required is not terribly complex.

The same basic encapsulate and integrate techniques can be used to bring legacy devices and software into the SOA management world. In some cases the interfaces can be added to the management "agents" running on the devices themselves. In other cases, proxy agents can be created that run off-board from a legacy device but that support a WS-based management interface to that device. The interface can be bound to the device via a number of techniques that basically require the off-board proxy agent to talk to the device's on-board management agent via whatever management language or languages the on-board agent speaks natively. In some cases the native languages are standard (e.g., SNMP, IPMI, WMI, etc.) but in others the native language may be proprietary and idiosyncratic, say a proprietary CLI or proprietary management protocol. Some systems management software vendors can provide a kind of one stop shopping for system developers interested in integrating legacy devices into the WS-management world.

Management Software That Can Manage Anything

If all hardware and software resources to be managed are managed via WS-based interfaces, then it becomes far easier to create management software that can potentially manage anything. Rather than management software having to deal with the "Tower of Babel," enough WS-based consistency (whether native or proxy based) can be introduced to make the management of heterogeneous collections of resources – say everything in a particular data center – possible. Almost by definition, management silos exist today because of the lack of common management languages that managers can use to interact with "managees."

Systems Management Software That Can Truly Manage Systems

If hardware and software resources are consistently manageable it becomes possible to truly manage systems problems like the 'Why is e-mail slow?' problem. All of the components that make up an e-mail delivery system can be comprehended by and managed by, say, extensible management appliances. More sophisticated, higher order services can be built up from lower order component services. The "e-mail performance diagnostic" service, for example, can query the routers, the network load monitors, the file servers, the mail servers, etc., to collect information necessary to resolve the problem. If one type of router is removed from the network and a replacement is installed, the benefits of loose coupling can be appreciated by simply removing consideration of the old router from the diagnostic mix and inserting consideration of the new router.

Real systems management power can be achieved by capitalizing on modularity, reusability, standardization, encapsulation, composition, layering, etc. Note that most of these descriptive terms are not typically used to describe the systems management software widely used today.

Integration of New and Legacy Management Software

When developers create systems management applications or appliances it would be foolish to ignore the value that existing management software can add. Consequently, it makes sense for new SOA management applications to employ the same encapsulate and integrate approach mentioned previously to integrate legacy management software. Consider the interesting case of open source management technology called Nagios. Nagios is an application that can be used to monitor the health of over a thousand different types of hardware and software resources, everything from servers, to routers, to databases, and other popular applications. Each different type of monitoring is performed by a different software plug-in called a (what else) monitor. Given the modular nature of the plug-ins, it is relatively easy to integrate them into a SOA application. Integrating other legacy systems management software is often more difficult but, in most cases, possible.

How Long Is the Wait?

The WS-MAN and WSDM standards are already quite far along. The DMTF has already released version 1.0.0a of its WS-MAN specification and can boast contributions from Microsoft, Intel, Dell, Sun, BEA, HP, and many other major players. The OASIS group has released a suite of 1.0 version standards documents related to WSDM, also with contributions from major players such as IBM, BEA, BMC, Dell, HP, Oracle, and others. Many system and software vendors are already starting to implement the standards.

Additional Reading

- *www.oasis-open.org*
- *www.dmtf.org. The DMTF website contains a wealth of information about systems management. The latest information about WS-MAN can be found in the "Standards" section.*
- *www.uxcomm.com*
- *http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsdm*

About the Author

Craig Wassenberg is the VP of Product Planning at uXcomm, a systems management start-up company working to bring services-oriented systems management to the masses. Prior to uXcomm, Craig managed some of the early HP OpenView architects, worked at Sequent Computer systems where he led the team that developed the world's first commercially successful NUMA multiprocess computer, and at RLX the company that pioneered the blade servers and was later sold to HP.

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

En Masse SOA Enablement Methodology Distilled

Part I: The analysis phase of the methodology

WRITTEN BY **PAUL O'CONNOR**

I never quite fathomed the software development methodology craze that was gripping enterprise computing when I came onto the scene in the early '90s. In those days development teams were managing complexity and enforcing quality via draconian software development life cycles (SDLCs).

A lot of big enterprises were using SDLCs that they had purchased as mindshare from system integrators or tool vendors. The ones people paid for came to be known as big-M methodologies, to denote their hegemony in the field. There was even a meta-process to measure how refined your methodology was. It was called CMM – the "Capability Maturity Model." CMM had five levels of "maturity" to which enterprises aspired, with each level adding an increasing number of artifacts, expensive tools, consultants, and documentation reams. If you ever toiled as a developer under a CMM-focused methodology, you would have gotten the sense that your job was to produce CMM artifacts, not software, in search of that elusive level-5 competency certification.

Stringent methodologies are perfectly suitable for the design of systems that never change, like an airplane flight control system. Since airplanes don't grow new wings, or add extra pilots over the course of time, or have to talk to other airplanes much, a software development methodology that makes it next to impossible to change the system makes sense. In fact, I would say it is highly desirable once the system works. The fact that it takes 10 years to develop new systems in this way is not a problem... it takes longer than that to create the plane's hardware anyway. This is not the case with systems predicated on integration patterns or having anything to do with the Internet. Change is the order of the day in these systems, and CMM-happy enterprises did not fare well in this climate. If a dollar figure could be placed on the cost of system development (actual cost and missed opportunity cost) to business because such methodologies were used, I am sure it would be astounding.

When I decided I wanted to be an SOA architect a few years ago, the old '90s developer in me wanted some type of methodology to follow, while the skeptic in me demurred. Further confusing the issue was the fact that SOA, the business-focused concept, has not been well defined for that long. It took a while for the dots to connect between the Web services and EAI platforms that we were using a few years back, all the way through to a business-focused agile execution paradigm. Along the way we were cautioned by analysts that there was great risk in doing SOA, that the business would undoubtedly be skeptical – especially when we asked for money to retool – and that we should do small pilots to validate ROI and assumptions about reuse in our enterprises. "Don't try to boil the ocean," they said. "Limit the scope of the problem domain or you will fail." Then along came 2006, which has been a year in which en masse SOA projects seem to be appearing with more regularity. As usual, those of us in IT do not have the luxury of scoping the problem domain. The business drives our projects, and the promise of efficiency and competitive advantage of SOA has derived the need for an all-encompassing way to proceed.

In thinking about the problem, it occurred to me that the ability of a services-based architecture to manage complexity (and thus engender agility and efficiency) stems from how architects can divide and conquer problem domains by using its precepts. The divide-and-conquer approach is the oldest method known to man for addressing complex problems. The scale of the problem is irrelevant as far as the method is concerned – the larger and more complex the problem is, the more dividing and conquering you do. With the big-M methodologies, when the problem got more complex, you were in even more trouble. So I wondered if a methodology (little-m) could be distilled for en masse SOA enablement by dividing enterprise development aspects along SOA boundaries as dictated by the fundamental concepts of SOA, including business and organizational aspects. I use the term "enablement" to denote the fact that we are not concerning ourselves with implementing an application (or set of applications). We are focused on creating an infrastructure that supports creation and assembly of services and related enterprise aspects via metadata management at sufficient change velocity to support our vision of business agility, as well as repurposing existing components and services for use in SOA. Specific applications and service compositions then leverage the SOA-enabled enterprise to meet specific business requirements.

Conventional wisdom has dictated that SOA projects start with and focus primarily on all things services – service creation, lifecycle management, and governance. Not so much has been put forth on repurposing an existing enterprise for a large-scale SOA project. Such an SOA enablement effort necessarily focuses more on the build out of an SOA infrastructure and leverages the well-worn service delivery and management methods axiomatically. In Part 1 I will focus on analysis for en masse SOA

enablement, with design/build/test being addressed in Part 2. It is the purpose of the analysis phase to elicit requirements for our SOA infrastructure and its associated change management facilities, thereby providing to the design phase what it needs to produce a detailed infrastructure design. Though it has been said that we are building an SOA enterprise and not a specific application, the fact remains that the problem domain will not be expressed in the abstract; it will be the usual mix of actors, process flows, interactions, use cases, etc. As such, separate requirements sets must be derived – system requirements that define specific process flows, services that support them, and consumers that consume them, and an infrastructure (non-functional) requirements set that will drive the design of all of the non-functional aspects of the enterprise SOA, i.e., the SOA "enablement" of the enterprise. The trick (and focus of Part I of this series) is to glean enough from the analysis of the problem domain to drive design of the SOA infrastructure such that it can serve the business's notion of agility and efficiency for as long as possible without too much modification. A series of broad work steps will be proposed, moving from the more abstract to the more specific, which should allow the infrastructure requirements set to be completed if applied to a specific enterprise.

First, Develop an Overarching Agility Vision

It has long been understood that the business value proposition of SOA is that it engenders agility and efficiency in enterprises. SOA agility is defined as the measure of the cost of change in IT systems: dollar cost and opportunity cost. If your enterprise can accommodate change easily, then changes demanded by business are going to be cheaper. In addition, accommodating more change allows business to offer a higher quality of service to customers and likely to achieve competitive advantage against less agile competitors. Conventional wisdom for piecemeal SOA rollouts has told us that to be successful in our efforts, we needed to establish success criterion and even detail ROI metrics by which we can chart our progress. In an en masse SOA enablement effort, we take this to the nth degree and paint an overarching agility vision. The analysis phase of the project must create this vision. It should be grandiose – idealistic in fact. It should be simple; easily understood by everyone on the team within a couple of minutes. It should make people ask, "How could we possibly get there from here?" SOA offers the ability to deliver extreme agility in due course of time (likely years) and project architects and business analysts would be remiss not to shoot for it. In keeping with the spirit of divide and conquer, the vision can well be broken out by sub-domain, like lines of business. By staying true to such a futuristic vision of your enterprise, downstream analysis and design tasks become easier. Start your analysis by creating an overarching agility vision for your enterprise.

Reengineer the Business Domain

Any kind of system analysis is going to focus on the business problem domain. Detailed domain decomposition is critical for accurate development and should be conducted by a team of business analysts with representation from the IT architecture team. If the business cannot accurately detail the task at hand, there is no hope for IT to deliver. With en masse SOA, it is essential to ensure that the business domain is taking advantage of the new capabilities it is being offered – the business domain should be reengineered. For example, if it adds competi-

The analysis phase of a project must create a vision of extreme agility. It should be grandiose – idealistic in fact. And it should be simple, easily understood by everyone on the team in a couple of minutes.

tive advantage to the business to be able to interact with 100 new business partners per year electronically, each having a different access policy profile, then demand it up front. "Business re-engineering" is a loaded phrase but an important concept here – an enterprise recast as an SOA can offer business a degree of agility that may be unexpected, such that it may well need changing itself. In en masse SOA enablement, the business is basically telling IT to reinvent the way that it enables business. Hence, a virtuous cycle is possible whereby business re-engineers and IT war-games enterprise changes. This cycle should play out throughout the entire analysis phase. The result should be a highly focused set of system and infrastructure requirements for your "new" enterprise that supports your overarching agility vision.

Inventory Existing IT Assets

Every enterprise making a shift to SOA is going to have lots of assets that can (and should) be reused in the new effort, without compromising the vision of the new SOA. This has been recognized previously as a key SOA analysis step, most notably as a key element of the IBM SOMA process. In en masse SOA analysis the boundaries are extended beyond services and enterprise aspects are considered as well – we are trying to repurpose enterprise aspects for SOA. Any database, service (Web service or otherwise), workflow engine, mainframe, identity store, policy repository, and even client applications that trap rules or policies at the client tier are fair game. The mining process can be quite complex, and this is where the divide/conquer approach comes in handy. Create a team with representation from each group that owns relevant assets and give them a tool with which they can collaborate. There are some great standalone asset management tools out there that can also be used down the road for semantics sharing and for governance.

The goal of this endeavor, from the infrastructure requirements point of view, is to derive requirements for repurposing existing assets into the SOA. Some things are going to be readily reusable, like existing Web services. Others are going to have value but be less ready to participate in an SOA, like old EAI platform services or business rules trapped in client code. There will be a few pieces that cannot be a part of an SOA, particularly things that are not consistent with your agility vision. There will undoubtedly be wide consensus about things that cannot be included, like that poor-performing DAO code that no one wants to see bog down the new system. As such, it should become clear what is needed to support reuse, e.g., a way to do data access in SOA instead of using that brittle DAO code, and perhaps a way to repurpose EAI platform services that are not currently exposed as compliant Web services. Likewise, a need to reuse an enterprise business rule management system may be elicited. Capture these types of requirements in your infrastructure requirements set. In the design phase we will systematically repurpose all of these elements for use in the SOA.

Derive a Semantic Model for the Enterprise

SOA architects have long recognized that semantic interoperability between service producers and consumers is fundamental for meaningful SOA. Service semantics include message-exchange vocabularies and meaning as well as expressions of service policies, constraints, and quality of service concerns. SOA agility requires that human intervention to translate between data dictionaries, which underlie services at the component level, not be required. Techniques and patterns for deriving SOA semantics are well beyond the scope of this article. But it can be noted that en masse SOA enablement makes the effort more complex by giving it enterprise scale.

In keeping with the premise of divide-and-conquer as a means to deal with complexity, it can be postulated that a federation model, supported in the design phase by transformation and composition capabilities, is a great way to get at enterprise domain data modeling for en masse SOA. If the enterprise has existing data silos with associated data models, there is no harm in respecting these boundaries if vocabularies and associated metadata jibes with the business domain model, expressed as a canonical data format. If the business model for an investment house defines customer as both a customer profile and also includes all financial holdings, don't worry that three systems of record (perhaps managed by different groups) may be needed to fulfill a "getCustomer" component service request. SOA infrastructure components can be used to abstract transformation and composition details from service creators, but you must elicit and capture these infrastructure requirements in the analysis phase.

It is essential to codify domain semantics in the analysis phase of the project. It is a natural outgrowth of business domain analysis and reengineering. If you employed a tool to do your IT asset mining, it will likely be able to help your team collaborate on semantics as well. The end result will be both a collective understanding of semantics, as well as associated vocabularies expressed in terms of XML schemas, WS-Policy, and XACML policies (which depend on data semantics and vocabularies), as well as other standardized formats. Transforms and composition points needed to support reuse and integration will fall out of this exercise along with associated infrastructure requirements.

Depending on your industry, you may be urged to adopt a canonical data format from an industry standards body. This is a good way to seed the semantics development process, but it will likely not get you all the way there. For one thing, it might not jibe with your existing enterprise data model. It is much more important that your semantics fit your business domain and that your team understands it implicitly. As such, a modified form of the standard form can be used. You can always transform data at the perimeter to a standard form to support federation as needed and capture such infrastructure requirements.

Focus on the Service Consumers

How new services are called and used by consumers must be well understood before SOA enablement can ensue. Consumers

CROSSXCHECK

networks

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

include user interface clients as well as internal and external service federation participants. New services and a transformed business will undoubtedly cause some impedance mismatches with consumers as they adapt. New consumers may ultimately be constructed, such as new user interfaces.

Traditionally, user interface clients suffer from a great deal of "hackitis" in big applications. It is very common for business rules and policies to be either hard-coded or trapped in a portal repository. This causes a twofold problem for en masse SOA. First, we would like to limit the number of rules and policy repositories that need to be managed. Large enterprises can have many portals, for example, and it would be quite counterproductive to manage rules and policies for each, especially in an environment where agility is the goal. Second and more generally, if consuming new services at the UI consists of potentially painful changes to backing code, hope of enterprise agility is lost. At the end of the day, it's the service consumers that present the services to the customers, partners, etc., and deliver the business value we are working so hard to create. It will likely be necessary to create infrastructure elements to support transition-of-service consumers to the new SOA. A clear understanding of the roadmap for these consumers to move to the SOA should be created, and infrastructure requirements is needed to support the roadmap captured. In the design/build/test phase (Part 2 of this series) I will also detail some portal design patterns that leverage the SOA infrastructure to drive agility in the UI tier while enabling central administration of business rules and policies. Even though it is usually pretty ugly, paint a picture of your service consumers' ability to interact with the emerging SOA.

Connect the Dots

Having defined the business flows/services (top-down approach), existing enterprise assets that will be repurposed for SOA (bottom-up approach), and the data and metadata semantics that drive service interactions, there comes a time to connect the dots and perform a gap analysis to determine a roadmap for design. IBM SOMA suggests performing "goal-service modeling" to complete the services set and prioritize services against specific business goals. Swimlane modeling can also be performed against all business flows, in priority order, to further this prioritization. This will reveal dependencies on micro services, components, and infrastructure elements such as transformation, mediation, and policy enforcement. By way of example, business analysts for a brokerage may place extremely high value on a new customer net asset value calculation. Such a composite service might include data from two internal domains – customer account and financial advisor management – as well as an external source in the form of a stock quote Web service. The process flow might be to gather all customer account data, for all clients of an advisor, compose the data into a higher-order domain entity (Advisor Customers), and calculate a net asset value for each customer based on data from the stock quote service. In this case, two internal data domains, one external service, and SOA infrastructure elements, including transformation, composition, and security policy administration and enforcement, will be involved. As such, an infrastructure design roadmap would include these elements at the beginning of the design process. In this way, the infrastructure requirements set can be made to include a prioritization of elements that will maximize business value with respect to development time.

Develop a System Change Taxonomy

Perhaps the most difficult aspect of en masse SOA analysis to get at is the development of a system change taxonomy; a classification of what has to change in the system, and how fast, to support the business agility vision. A major element of infrastructure design will be to create facilities for metadata change management, i.e., rules, policies, service contracts, service compositions, transforms, etc. Without a specific classification and prioritization of system change, effective design for change is not achievable. The infrastructure requirements set must capture this as completely as possible. Some metadata change requirements will be obvious, like service compositions, while others will be less so, such as changes of all the types of business rules in the enterprise. Care must be taken to expose dependencies between metadata under threat of change, e.g., schema changes that impact policy resource expressions defined in XPath. Create as detailed of a classification (and sub-classifications if need be) of changes in your system as you possibly can and codify it in the infrastructure requirements set. Don't forget to include aspects that may not even exist in your current enterprise, e.g., fine-grained access control policies, federation policies, and compliance policies. Development of metadata change facilities and repository management will be a key infrastructure design step.

En Masse SOA Analysis Is Iterative, Expansive, and Time Consuming

En masse SOA analysis may well begin with a very small team, even for a large enterprise. Broad goals may be painted at first without much specificity at all. Analysis artifacts such as business process flows, policies, and IT assets are derived by iterating through the activities that create them, dividing the problem domain further with each pass and including new team members as needed. At some point in the process, it will become apparent which infrastructure elements are likely needed to support the system. At this point the infrastructure design process can begin and join the analysis iterations. As the infrastructure design becomes apparent, service and application design can also join the iterations. When final requirements are signed off, system design can complete and implementation can begin.

Summary

2006 has been a year in which some enterprises are launching en masse SOA enablement efforts in an attempt to engender efficiency in their IT operations and achieve competitive advantage through agile business process management. As such, this article attempts to distill an en masse SOA enablement analysis method as an adjunct to well-known service development methodologies, as well as by keeping the method true to the divide and conquer precept of SOA. The goal of seeding an infrastructure design process with an infrastructure requirements set was espoused. Emphasis was placed on getting the vision of agility set and recasting the business domain in its light. It was then advised to elicit infrastructure requirements through the processes of mining for IT assets, developing domain semantics, focusing on service consumer channels, and creating a system change taxonomy. Prioritization of infrastructure design elements can be elicited when the dots are connected between the future state vision of the business and existing IT assets. An iterative process of analysis was espoused that seeds an iterative design process. Part 2 of this series will focus on the design/build/test aspects of the methodology. ■

About the Author

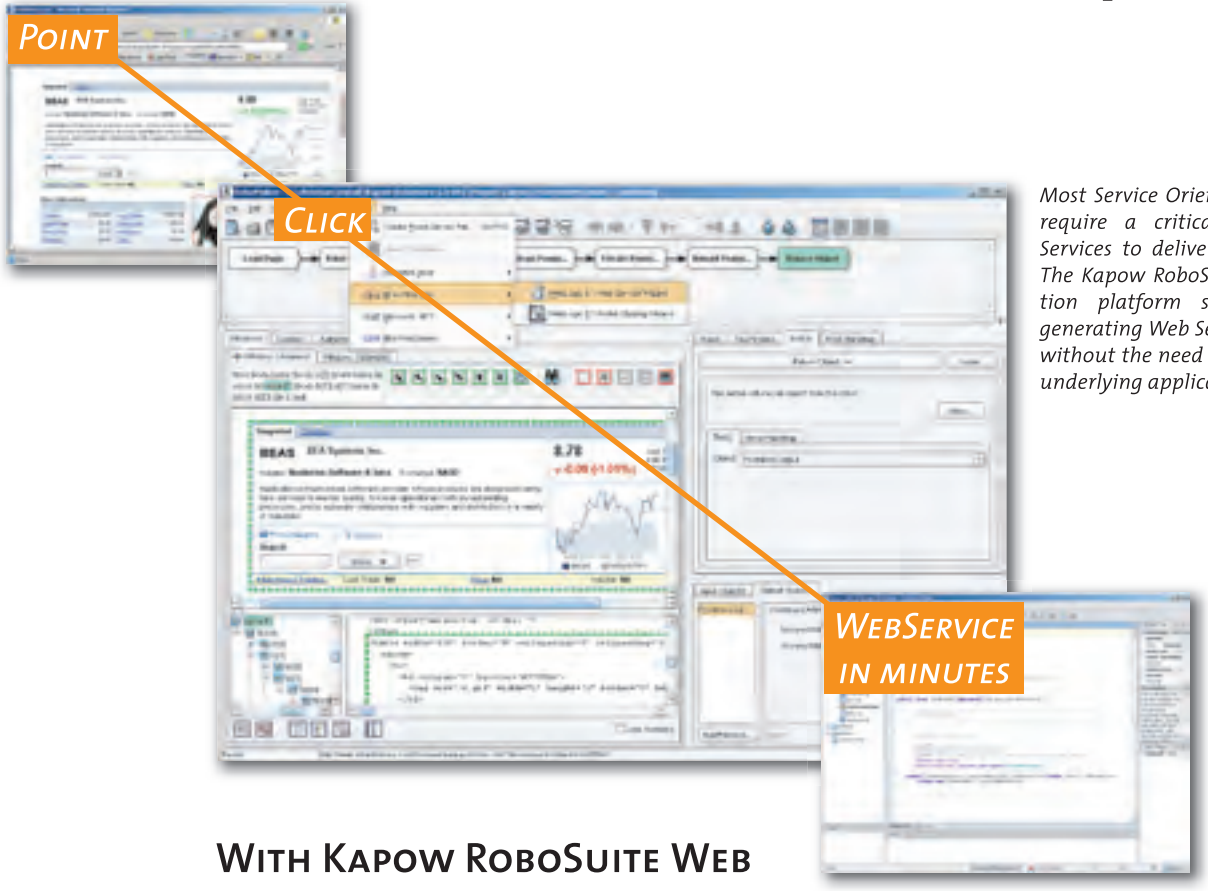
Paul O'Connor is chief architect and SOA Practice managing principle for e-brilliance LLC (a leading NE SOA consultancy), and is currently doing major SOA architecture and implementations for Fortune 100 clients across the US. Previously he was chief architect for Damascus Road Systems, specializing in security architecture.

poconnor@e-brilliance.com

Leverage existing IT to generate Web Services

[in minutes]

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



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

With Kapow RoboSuite Web Integration Platform you get

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

kapowtech.com

kapow TECHNOLOGIES

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

6 Basic Rules for Securing SOA based Projects

Common Sense SOA Security

WRITTEN BY **KEVIN SMITH**

Over the past five years, an "alphabet soup" of new Web Services Security specifications, standards, and buzzwords has been thrust upon the technology scene. As we have watched the evolution of many Web services security specifications, it has sometimes been difficult to wade through the murky and dangerous waters of implementation patent issues, vendor wars, competing specifications, and challenges of interoperability. These waters have thankfully become clearer over the past few years, due to vendor agreement and some diligent work in standards organizations such as OASIS and the W3C. Mature specifications have emerged and have become (or are now on their way to becoming) standards. As a result, many standards-based Web service security toolkits and implementations have been shipped that allow developers to build solutions quickly. That's the good news!

Is there bad news? The bad news isn't tragic, but it must be stated. Because it is relatively easy to use these new products to build security solutions without a security background and without looking at the big picture, it's easy to develop a security solution that is over-engineered, complex, poor-performing, and possibly even insecure. Because our current tools are focused on ease of integration, the use of these tools may not require any background in information assurance or information security techniques. As a result, bad solutions can and will be developed. In meeting this challenge, we must underscore the need to focus on the "A" in SOA, the essential tenets of information security, and a few common sense rules.

This article will build on six basic, common sense rules and best-practices that you can follow as you are building your SOA security solutions.

1) Plan Ahead

A major pitfall occurs when architects wait until the last minute to plan for security. This happens more than we would like to admit. Either requirements aren't plainly stated and the team is surprised at the last minute, or someone makes a deliberate choice to worry about security later. Many times, our focus is on designing, developing, and testing core functionality of our services, and a security focus is not scheduled until a few weeks before deployment. The standard model of Web Services Security SOAP messaging appeals to designers and developers because of the logical separation of security data in the SOAP header and business data in the SOAP body - and therefore, the physical separation of security logic in security handlers and business logic in the main Web service code. Unfortunately, this separation also provides an unacceptable excuse to worry about security later.

For your project, it is absolutely vital to determine your security requirements very early in the project. Based on those requirements, you can then delve into how you are going to satisfy them. A good requirements analysis phase is absolutely necessary because most customers may simply say "Just make it secure!" From the beginning, you will need to determine what requirements may exist for authentication, authorization, integrity, non-repudiation, auditing, and confidentiality. Talk to your customers and end users and find out who will be levying security requirements. Once you get with the right people, either work with them to gather security requirements or help them determine what their security needs are. The key point here is to make sure that you find out the true requirements at the beginning of your projects in order to make the appropriate design decisions early.

2) Know the Enterprise Infrastructure

As many of us are developing our applications, we often forget about the environment in which we will deploy our solutions. As we architect our security solutions, we must never architect in a vacuum or assume anything about the existing enterprise security infrastructure. Security-wise, there will undoubtedly be systems such as LDAP directory servers, policy servers, and Public Key Infrastructure (PKI), with which you will have to integrate. This enterprise infrastructure may dictate much about how your system will operate. Issues regarding connectivity, credential management, policy and privilege management, user authentication and authorization, and auditing may be affected by the existing infrastructure. This gets back to our first point – plan ahead by asking the right questions as you begin your project. Do not assume anything about the infrastructure; make sure to know your enterprise integration points.

3) Stick To The Standards

For those of us who had to develop Web services security solutions in the caveman days (three to five years ago), many times we had to play the role of amateur cryptographer because the secure messaging standards had not yet been developed. As a result, many of us have grown accustomed to tinkering and creating our own "secure" messaging concoctions. This may not have been the best thing to do then, but it is definitely the wrong thing to do now.

Now that there are accepted standards - such as WS-Security and its associated token profiles used for identity propagation (WS-Security SAML Token Profile, WS-Security X.509 Token Profile, WS-Security Username Token Profile) - as well as emerging specifications in standards bodies (WS-SecureConversation, etc.), there should no longer be any reason to create a home-grown security messaging syntax. Certainly, you must be able to understand the purpose and use of these standards and security specifications in order to meet your security requirements, but you need to avoid creating home-grown protocols. In addition to the problems that you will have down the road involving lack of interoperability with other systems, any non-standard solution created by wannabe cryptographers will most likely have security vulnerabilities that could come back to haunt you in very ugly ways. We have standards for a reason - embrace them.

4) Think Like a Security Guy (or Hire One!)

If you are developing a security architecture for a SOA that involves sensitive, proprietary, or highly confidential information, you will either need to be well-versed in information security and the current standards, or you will need to hire a security professional. Understanding the fundamentals of information security will be necessary not only in determining the security requirements at the beginning of your project, but also for designing the architecture, testing, and - if necessary- getting through the proper certification and accreditation processes.

Although you can lean on the WS-Security standards and toolkits to do the hard work for you, many of these standards have multiple options that you will have to study and select, based on the security requirements of your system. Most Web services security solutions must be able to thwart message replay attacks, protect availability, and, depending on the requirements, provide some or all of the classical security goals: authentication, authorization, confidentiality, integrity, and non-repudiation. Remember that the Web services security standards may not be intuitive for those who have not had an information security background, so make sure your people are educated in information security.

The double-edged sword is that cryptography - when done right - is a friend of security, but cryptography always has a negative impact on performance.

5) Remember the Nature of Web Services

In a large SOA pilot four years ago, initial tests showed that all Web services were functional and operating normally, so the developers were relieved. When there were errors related to access control, a Web service would throw a SOAP fault. The problem? In this particular environment where much Web service chaining was happening, it wasn't always quite apparent which service threw the SOAP fault, because the SOAP fault contained very little information. I remember calling the administrator of the first Web service on the phone, asking him if he could look in his logs to see why the SOAP fault was thrown. Then he told me that his Web service didn't really throw it, but that his Web service programmatically called another Web service, which must have thrown the fault. I then called the administrator of the next Web service on the phone to get more information. It goes without saying that the process got old quickly.

We can still have similar problems today if we do not understand that Web services are essentially "black boxes" with published interfaces, and that they will be called and used together in ways we may not always anticipate. Web service chaining and orchestration should therefore be expected, so you must understand the distributed and dynamic nature of Web services. In planning for this from a security perspective, you need to focus on Web services transaction management, centralized auditing, and detailed, descriptive error handling. This relates not only to Web services security in regards to auditing, but also to Web service management.

6) Understand the Double-Edged Sword of Cryptography

Cryptography is a mechanism that we use to achieve most, if not all, of our security goals. It is with cryptography that we achieve authentication, confidentiality, integrity, and non-repudiation. At the same time, we must acknowledge that cryptography is performance-intensive, which could slow down systems and lead to problems with availability and denial of service. The double-edged sword is that cryptography - when done right - is a friend of security, but cryptography always has a negative impact on performance. Think of a seesaw with your security goals on one side, and your goals of performance and availability on the other. The key thing is to keep things balanced.

Sometimes engineers forget about the high cost of cryptography when they choose to secure services that don't need securing, or when they apply unnecessary cryptography. I once saw a solution where the body of every Web services call was signed by the caller with XML Signature, encrypted to the recipient with XML Encryption, and each Web service client initiated an SSL connection to each Web service for every request. What was the biggest problem (other than the redundancy of encryption for each connection)? The back end Web services were reading bytes from a public Internet Web page, so the developers used a lot of cryptography to encrypt, decrypt, sign, and eventually validate the signature of publicly available data that had no assurance of integrity. Essentially, they created a very slow SOAP-based alternative to HTTP!

In planning your SOA security solution, it is absolutely essential to be wise and deliberate about the cryptography you use. Going back to the first rule (Plan Ahead), it is important to truly understand your security requirements. If there is no requirement for confidentiality, then you shouldn't use encryption! If there is only one part of your message that is sensitive, then perhaps only that part of your message should be encrypted. If you need to encrypt everything between two points only, consider using SSL. If you have a requirement for technical non-repudiation, then using XML Signature in your WS-Security messaging is a valid choice. Realize, however, that all cryptography will have a performance cost and will affect the scalability of your system. If you have serious cryptographic needs in solving your security requirements, consider using hardware acceleration products that do the cryptography and XML processing in the hardware. The use of cryptography should be intentional, based on your requirements, and you must plan ahead for its impact on performance.

Conclusion

Diving head-first into detailed implementation issues involved in SOA security, it is truly easy to make mistakes or become overwhelmed if you don't initially focus on the big picture, beginning with understanding your true security requirements, understanding the fundamentals of information security, and understanding the proper use of Web services security standards. This article has attempted to capture this big picture by providing six common sense rules to use when planning and developing SOA security solutions. Above all else, it is important to focus on basic project management, security architecture fundamentals, and common sense!

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 has been an invited speaker at many conferences, such as JavaOne, OMG Web Services, Association for Enterprise Integration (AFEI), ApacheCon, and Net-Centric Warfare.

kevintsmith@comcast.net

Other companies in this magazine spent a lot of time on pretty ads. As you can see, we did not. We spent our time hiring the best people and training them to deliver outstanding support for your website. We spent our time building a state of the art datacenter and staffing it with people who care about your website like it's their own. Compassion, respect, credibility, ownership, reliability, "never say no," and exceed expectations are words that describe our service philosophy. From the first time you interact with us, you'll see what a difference it really makes. And you'll also forgive us for not having a pretty ad.

The Core Four:

A Service-Based Approach to Security Management

WRITTEN BY **VADIM LANDER**

I recently attended a security conference where thousands of security products from hundreds of vendors were all vying for attention. While most of these products filled a legitimate need, the array of products reminded me of an orchestra warming up. Each instrument may sound good by itself, but together they would be cacophonous without a conductor.

Companies need to develop a coherent strategy to align security investments with the real risks faced by the business. How can a business leverage identity, role, and policy information to improve governance and security? How can an enterprise tie together the knowledge gathered by disparate security systems to help intelligently manage security services to effectively protect the business and increase IT effectiveness?

This challenge mirrors what's happening in our data centers. Too many companies have overspent on silos of solutions that are nearly impossible to manage holistically and leave the business at risk. Many enterprises have deployed dozens of identity management and threat solutions to thousands of desktops and servers. Yet the endless onslaught of unmanaged worms, viruses, and other attacks (both inside and outside) prove that these solutions aren't providing the necessary protection when implemented as "one-offs." The security management task - to make sense of it all - isn't getting any easier either. If companies can't get a handle on their security information, the entire corporation could be exposed to accident or attack.

From the CEO and CIO on down, it is imperative to align security and risk management with the business. To ensure business agility, IT must be able to react quickly to ever-changing business and security requirements. When a new threat or risk is discovered, the IT infrastructure must be able to mitigate and control the exposures before excessive damage can be done. Aligning security with business also means being able to instantly support a new business application that requires a strong degree of authentication. Another example is ensuring that every application meets strict audit requirements imposed by business or government regulations. Businesses need to create an agile infrastructure that treats security as if it were a volume button - one that can be adjusted as needed without disruptive retrofitting or architectural upheaval. Making this happen requires identity and security management tools that can deploy policies on the fly.

Making the Vision a Reality

Enterprises require a broad, contextualized view of their networks and security stance in much the same way that Google Earth lets users zoom down to the street level and out to the country level with the touch of a button. Such a comprehensive and intelligent view would help make it easier to understand dollar value impact to the business when problems occur or when new applications are being assessed or deployed.

So how can the technologists help make that vision real? One important part of the solution is to implement a cohesive approach to IT and security management—one that has been architected to create a cross-enterprise, platform independent view. The key to success is approaching the solution in a modular, service-oriented manner that supports the rapidly changing needs of the business.

One of the failings of today's disjointed, piecemeal approach to security is the myriad of disconnected solutions that have been rolled out in silos of ownership and management. These silos are brittle and restrictive. Service-oriented architectures (SOAs) have been developed to provide businesses with the ability to deliver information and services to an organization's ecosystem of customers and partners and then to manage these services to the appropriate risk level. This eliminates restrictions or roadblocks based on platform restraints, legacy deployments, or political or business unit boundaries. By liberating the intelligence from the silos, the promise of SOA is that enterprises can use both network and identity information to increase the focus and granularity of policy decisions.

Consider the earlier example of a business application that requires strong authentication. An enterprise creates an authentication service that all of the application developers call from within new applications. The service provides options such as password or biometric authentication, as well as control parameters such as password length restrictions and cryptographic key length. Using an authentication service shortens the development process and ensures that the resulting application is already in compliance with the corporate controls and infrastructure. If the developers had to create a new authentication service for each application, the process would be slowed as the service was developed and implemented. In addition, the missing connection to an approved authentication service would require extensive testing, approval from security organizations, redundant development, and possible changes to the architecture.

This takes time and introduces additional risk. Using the services model as part of the development process allows organizations to create new user experiences and services on the fly - not just a personalized experience but one that is aligned with business and security needs as well. This intelligence needs to be managed by a heterogeneous policy management layer that provides the management process described above.

Instead of isolating (and duplicating) security information within the various silos throughout the enterprise, information silos must be broken down and replaced by service-savvy models that help enterprise technologists turn data into knowledge that serves the business. To support the service-oriented architecture, four cores of security building blocks are needed — users, assets, services, and policy. Each core component can be leveraged to realize the vision of advanced, intelligent security management services.

Users

Users are actors on a network. When looking at how a business runs, one of the most critical considerations is the profile and unique characteristics of the users accessing systems and services. In a service-oriented world, services also can be "users" of other services. To strategically approach the task of breaking down the security information silos, it is essential to plan for management of data related to users. This is because user data is a critical piece of information that enables security management services to function intelligently. Managing the entire user lifecycle - such as creation of the user ID, provisioning a profile and access for the user, and real-time awareness of what the user is doing (be it changing a password or accessing a service) - enables IT to know what is occurring on the network and how the services are being used.

If an organization approaches the user management function from a services approach, the vision becomes more focused. Each application no longer acts as a separate silo of user management— the user management itself is abstracted into a service. Now, instead of leaving the user data as breadcrumbs attached to unique applications, the applications interact with the user management service as needed. Federation takes this concept one step further by liberating the user from the repository. User access is managed and monitored from a central console that serves as an intelligent, policy-aware gatekeeper for all the applications and devices on the network. Concerns around user experience, customization, access control, and audit can be alleviated because centralized policies are called through a user service to which baseline controls can be applied. In addition, this information can be fed back into the security management service to provide risk data and usage data to the business for audit, compliance, and risk control.

Assets

Users need to access resources, such as a portion of the network, a device on the network, or an application. Users also have devices to access resources, and the network has devices that are accessed. Information about these assets includes the type of device, the operating system on the device, configuration information, and current patch levels. This information is essential for creating a complete vision of current vulnerabilities and risks on the network, as well as the overall risk level for the business.

Armed with information about what is on the network, a security management service can begin to make intelligent decisions about those devices. For example, a critical application that resides on a device that can't be patched due to operational restrictions (such as a healthcare device that would no longer be considered approved for medical use if configuration was changed) can be flagged as a high priority risk. If vulnerabilities on that system are discovered, they would need to be protected via other mitigating controls such as being placed in a protected zone or via enhanced auditing. Coordination of the information about this asset, as well as workflow regarding how the controls were approved and implemented, would be handled by the intelligent security management service.

Many enterprises already have some type of asset management investment in place, so the challenge is often how to find a way to share existing data with the security management service. This can be accomplished by creating a service model overlay from which security management services can consume legacy asset data, and translating that data into usable information within the new model.

Services

Services provide distinct functions that can be used and reused by users, applications, and other services. The service approach decouples functionality and data from proprietary applications and makes

Each instrument may sound good by itself, but together they would be cacophonous without a conductor

them available to a wider ecosystem.

To work efficiently, these services need to be embedded in the infrastructure and be accessible to a variety of systems, applications, and users. An example can be seen in identity and access management (IAM). Embedding an entitlement service into an application platform provides hooks that give new applications and services access to existing policies and user repositories. The services themselves can be self-protecting. That is, they carry with them information and data that protects the transaction or usage. For example, say that a company wishes to provide a partner with access to data for only one day so the partner can check out the offering. An IAM service would issue a temporary user identity with a time to live of 24 hours. The service maintains the kill time on that identity so that it is deleted at the end of its lifecycle. The applications that interact with this service do not need awareness of the kill date, because it is already contained in the self-protected service.

Ultimately, the service approach abstracts complexity from the application or service developer, the administrator, and the user. It also supports federation. This is powerful because it allows the business to centralize the policies governing that self-protection into a single place - while also enabling the business to gather data about service usage. This data can be aggregated into the security management service to provide full visibility into what is really happening within the business.

Policy

With the previous three components in place, the nirvana for security management is nearly achieved. However, one key ingredient is still missing: policy. Policy is the glue that ties assets, users, and services together to enable organizations to adjust security procedures to meet business requirements. Policy infrastructure has to be heterogeneous and standards-based, and offer lifecycle management capabilities. It needs to support different policies such as authentication, authorization, auditing administration, privacy, and governance. By implementing a service-based approach to policy management, organizations can integrate and consolidate individual policy silos into an enterprise policy tier that governs enterprise IT security services. This policy service tier enables organizations to fine-tune alignment since it allows decisions to be made centrally and implemented en masse across the enterprise without the need for costly individual testing and approval.

To illustrate how a policy service can improve the alignment of security with the business, consider the case of a new business partner in a typical enterprise. For most IT managers and application architects, this means long hours of integration work, testing and retesting of rules, roles and permissions, and potentially, weeks of effort cataloguing existing system security mechanisms. If, in addition, this partner has contractually negotiated a higher level of response time for security services, many IT organizations may find the challenge just too great to bear and might even be tempted to compromise security in order to fulfill the obligations of the business. In a policy-driven enterprise, however, individual systems would not have to be catalogued and mapped to ensure compliance with the business goals. An appropriate change to the high-level policy is all that's required to make the necessary changes at the application or system level.

Presence is the perfect example of a policy decision point that can be applied to a variety of policies. The location of a user, device, or service is a telling piece of information that affects risk. Presence information includes data about not only when a user or device is active, but also where it is in the context of the overall environment. This is especially true in today's mobile environment, where users and services are accessed from points all over the globe, including kiosks, un-trusted networks, and a variety of devices such as PDAs and handhelds.

Policy decisions about access might need to be changed based on information related to presence. If we consider the data points of physical presence and integrate these into the security management service, organizations can identity possible policy violations. Say, for example, a user accesses the corporate headquarters in New York using a physical access badge card at 9:00 AM and then the same user ID attempts to login from a laptop located in California at 9:05 AM. This presence disconnect could have deep security implications for the firm. It could represent an active attack or an unauthorized use of system resources. Another example is the set of services that a user may be approved to access but only under certain presence conditions. For example, managers may be able to access payroll applications when connecting from a trusted network, but will be blocked when connecting from a wireless connection at a coffee house.

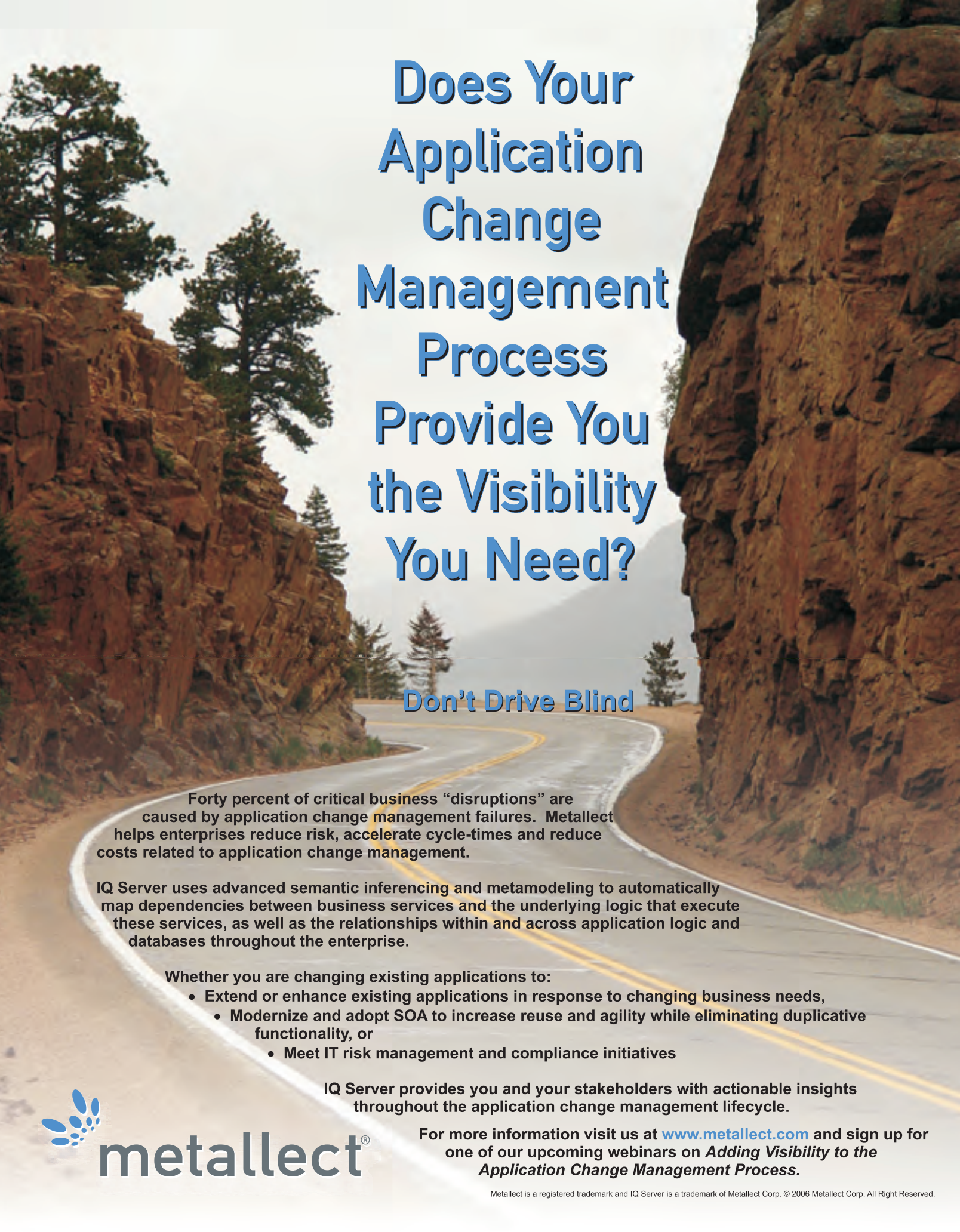
Using advanced analytics within the security management service, such violations can be identified quickly and prevented. Such a model empowers the business with the intelligent flexibility to deliver (or revoke) services from a central point based on presence information. It is the security management service that intelligently ties this information together and provides the full view of risk from which policy decisions can be made.

Bringing Security into Harmony

By taking a service-based approach to security we can help enterprises understand and quantify their security controls in a way that effectively manages risk while promoting more efficient business operations. Like the conductor of an orchestra, the service-level approach brings the disparate sounds of the instruments into alignment and allows the melody to appear. The best way to bring security into harmony is to implement the "core four" - users, assets, services, and policy - as services and then to feed the data into a security management service that is built on top of a flexible policy management infrastructure. In this way, the disparate bits and bytes of security data can be transformed into the intelligent, contextualized knowledge necessary to protect the enterprise in an increasingly complex environment.

About the Author

As Chief Security Architect, Vadim Lander advises the Company on key security technology trends, further develops CA's strategic direction, and works with the engineering teams.



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. Metallect 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.

metallect®

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.*

Web Services on Rails

A Web 2.0 perspective for SOA

WRITTEN BY **ABHISHEK MALAY CHATTERJEE**

The term Web Services refers to loosely coupled, executable application components linked dynamically over the network with open standards. Typically, they are software components that employ one or a combination of the following XML standards: SOAP, WSDL, and UDDI. Ruby on Rails is an open source Web application framework written in Ruby that closely follows the Model View Controller (MVC) architecture. It strives for simplicity, allowing real-world applications to be developed in less code than other frameworks and with a minimum of configuration. To define Rails - the Web Services or a Web Framework - we have to say that Rails is a full-stack, open source Web development framework that requires comparatively less time and effort to code XML interfaces than most other frameworks.

Implementation

Rails has a vast library of plug-ins, and similar plug-ins can also be developed easily. The Action Pack splits the response into a controller part (performing the logic) and view part (to render the template). Action Pack implements the CRUD (create, read, update, delete) actions as public methods on Action controllers that are responsible for handling all of the actions relating to a certain part of the application. Action views use templates that are written using embedded Ruby code in tags along with HTML similar to what ASP does.

```
BlogController < ActionController::Base
    def display
      @customer = find_customer
    end

    def update
      @customer = find_customer
      @customer.attributes = params[:customer]
      @customer.save ?
        redirect_to(:action => "display") :
        render(:action => "edit")
    end

    private
      def find_customer() Customer.find(params[:id]) end
end
```

Actions grouped in controller as methods

```
<% for post in @posts %>
    Title: <%= post.title %>
<% end %>
```

Embedded Ruby for templates

Choosing a Web Server

WEBrick, the default server for Rails, is written entirely in Ruby. It supports the standards you'll need: HTTP for communications, HTML for Web pages, and RHTML for embedding Ruby code into Web pages for dynamic content. WEBrick has some important advantages:

- It comes with Ruby, so it's free and always available for use or packaging of projects.
- It's built into Rails, so we don't have to go through any special effort for integration.
- It can make direct calls to a Rails application because they are both written in Ruby.
- It's simple to use.

Apache

Although WEBrick is the most convenient choice, it's not the most scalable or flexible

choice. The Apache Web server is the most widely deployed Web server in the world. You can choose from an incredible array of plug-ins to run dozens of programming languages or serve other kinds of dynamic content. Apache scales well, with outstanding caching plug-ins and good support for load balancers and sprayers (machines that efficiently spread requests across multiple Web servers). If you're looking for a safe solution, look no further than the Apache Web server.

Other options are lighttpd, Mongrel, and any Web server that supports CGI.

Action Web Service

Action Web Services have procedures for implementing SOAP and XML-RPC Web Services Protocols. Automatic generation of the WSDL (Web Service Definition Language) files or the contract for the Web Service is another feature of Action Web Service. To expose a method as an API, we have to use the ActionWebService::API::Base derivative. We can then specify the API definition class wherever we want to use the API. The implementation of the method is done separately to the API specification. Action Web Service camelcases the method names according to the Rails inflector rules for the API visible to the public. This means that the names in the WSDL file will be camelcased. For example, a name such as loan_amount gets converted to LoanAmount in the WSDL file. However, this can be disabled by using the inflect_names false command in the API definition. Input output parameters are stated as api_method :<method name>, :expects => [<data type>], :returns => [<data type>].

The Api can then be used in the controller class using the statement web_service_api <apiName>. The following are the advantages of using Action Web Services:

- Drop-in support for accessing other Action Web Service APIs into our controller code.
- Custom APIs generated using these Action Web Services and exposed as endpoints can be used by other platforms for their functionality.

The following are the disadvantages:

- Too much support for XML-RPC types of messages.
- It does not implement all supported features of W3 specifications, only those required for interoperability with other platforms such as .NET.

Choosing the XML Message Format in Rails for Web Services

Although SOAP stands for Simple Object Access Protocol, creating messages that can comply with all the interoperability rules (Basic Profile Compliance) is difficult, and the messages generated are complex and huge. Thus, another standard called REST based Web Services are becoming mainstream with industry leaders like Yahoo and Amazon exposing their Web Services as REST Web Services.

The creation of REST Web Services is supported by Rails 1.1 and above; this makes the creation of Web Services in Rails a child's play. REST stands for Representational State Transfer; this basically means that each unique URL is a representation of some object. We can get the contents of that object using an HTTP GET; to delete it, we might use a POST, PUT, or DELETE to modify the object.

Some Advantages of REST over SOAP

A REST call is actually accessing a Remote Resource instead of a normal Remote Procedure Call. It employs the usage of existing standards such as HTTP, XML, or TCP/IP, rather than creating new standards as in the case of SOAP. REST covers the most common scenarios and problems rather than trying to address every possible scenario.

Creating a Simple Web Service

The first step is to create a new Rails application and then create the raw structural files within which we will code the service.

```
rails FirstWebService
ruby script/generate web_service HelloWorld hello
```

Now, browse to the API folder and edit the helloworld_api.rb to define the interface. This step is optional but it's recommended to implement.

```
class HelloWorldApi < ActionWebService::API::Base
    api_method :hello,
    :expects => [:string],
    :returns => [:string]
end
```

Next, write the controller. Edit the helloworld_controller.rb. Here, we want to write the actual Web service code and specify the name of the API for the Web Service.

```
class HelloWorldController < ApplicationController
  wsdl_service_name 'Hello'

  web_service_api HelloWorldApi

  def hello (str)
     return "Hello "+str+", Howz life??"
  end
end
```

Running the Web Service

When finished with the code, use the following command to start the WEBrick server. The default startup port is 3000.

```
ruby script/server -p <port>
```

The endpoint for the Web Service is given below:

```
http://<machine-name>:3000/HelloWorld/invoke
```

The following is the WSDL endpoint:

```
http://<machine-name>:3000/HelloWorld/hello.wsdl
```

Testing Web Services

A big advantage to using Rails for Web Services development is having access to its testing features:

- Functional Testing: Test the APIs by creating a functional test for the controller dispatching the API. Then call the #invoke in the test case for performing the invocation.
- Scaffolding: Adding the following directive to the controller class automatically generates an HTML interface to manually test the Web Services (this feature is similar to what IDEs like Visual Studio provide): `web_service_scaffold :invoke`

Conclusion

Dynamic typed languages like Ruby extensively use REST instead of SOAP-based Web Services. The core architecture has many similarities with the J2EE architecture. However, the approach to development of Web applications is different in the two frameworks. Rails prefers explicit code instead of configuration files, and the dynamic nature of the Ruby language generates much of the plumbing code at runtime. Finally, the MVC structure of development of Web applications is the biggest advantage in Rails.

References

- *http://manuals.rubyonrails.com/read/chapter/67*
- *http://www.xml.com/pub/a/2006/04/19/rest-on-rails.html*
- *http://www-128.ibm.com/developerworks/java/library/j-cb08016/*
- *http://searchwebservices.techtarget.com/tip/1,289483,sid26_gci1180700,00.html*

About the Author

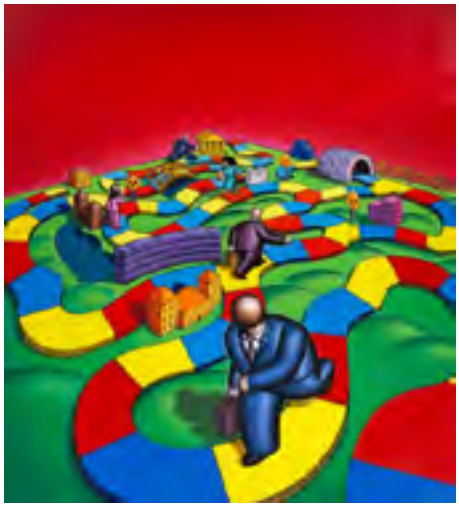
Abhishek Malay Chatterjee works as part of the Web Services COE (Center of Excellence) for SETLabs, the R and D arm of Infosys Technologies Ltd., a global IT consulting firm. He has substantial experience publishing papers and presenting at Web services and SOA conferences.

Abhishek_chatterjee@infosys.com

Service Orientation, the Enterprise Architecture Way

Examining SOA for what it is and isn't

WRITTEN BY **GEORGE S. PARAS**



Service-Oriented Architecture (SOA) is gaining momentum as a new IT implementation paradigm. Organizations are eager to capitalize on its benefits. However, with many of these organizations focusing too narrowly on project-specific implementations, though, some are at risk of never achieving the full value that the SOA concept could bring them. Their result will be a collection of technology-driven remnants instead of a holistic, business-driven system of integrated services.

When SOA is viewed as a "technology" it is often pursued without a clear understanding of how the resulting services fit as part of the larger enterprise architecture. The solution is to view service orientation as one of many means to conceptualize and deliver holistic enterprise architecture, not the other way around. By doing so, the outcome will reflect the true promise of SOA as a powerful business design paradigm.

SOA Fictions

Narrowly-focused, project-specific SOA approaches are a double-edged sword. While they can produce rapid, short-term value, they often also produce unintended consequences in the form of new silos. It is easy to understand how this can happen. Driven by different requirements, often from different parts of the business, implemented by different teams, at different times, during different phases of tactical decision making, each with there own interpretation of strategy, and during different stages of technological evolution - it is a leap of faith to believe that the end result will be an integrated, aligned, adaptive enterprise. Yet that is precisely what many enterprises hope.

While it is self-evident to many that it is a long-shot to achieve an adequately aligned and integrated enterprise, there continues to be hope among the SOA-faithful that technology and brute-force approaches will save the day. Don't misinterpret this statement as a criticism of SOA. SOA has the best opportunity of any recent paradigm change to profoundly influence the way businesses operate and the way IT participates. Indeed, great strides have already been made as the industry educates itself, adjusts and generally tunes in to the approach. However, like many innovative ideas of the past - such as early distributed computing frameworks, data-driven and metadata approaches, and pioneering object-oriented approaches – there's a risk that hasty and uncoordinated adoption could cause momentum to cool as expectations prove overly grand. To prevent this, it is important to examine SOA for what it is, and what it isn't, and define exactly what it means to the business. As examples, look at these common marketplace perceptions and the fictions underlying them.

Fiction: SOA and Web services are the same thing

While Web services represent the currently accepted (and perhaps best) way to implement a service oriented architecture, it's not the only way. More traditional technologies such as EAI (Enterprise Application Infrastructure) approaches can support the general concepts underlying service orientation. In fact, many 30-40 year old process-control systems and embedded real-time systems were implemented as services. There are even examples of services implemented in COBOL/CICS environments. Service-orientation is a logical concept that can be applied at many levels, not just via a technology such as Web services implementation methods.

Fiction: Just Start Building Services, the Approach Will Take Care of Itself

There is a common belief that there is so much power in the underlying standards, and their loosely coupled formats and protocols, that simply building collections of services that adhere to these standards will lead inevitably to a fully-functional representation of the enterprise. It is a rather large leap to expect that any two services, created at random in an enterprise, would have agreement on concepts as broad as strategic context or as narrow and specific as the meaning and proper use of a data element called "customer."

Deconstructing SOA

The acronym "SOA" continues to trigger an immediate association with Web

services, in large part due to the successful marketing by Web services proponents. To truly understand SOA, though, requires breaking this association and examining SOA in a different light. To begin, the term must be deconstructed so that the real meaning behind the concept can be discovered.

First, abandon the term "architecture." Most IT professionals are so comfortable with the tools, techniques, infrastructures, products, etc. that make up an architecture, it is only natural for them to view SOA as an "architecture," with its own enabling technologies, interfaces, standards, vendors, and implementation details. That approach misses not only the larger picture, but also its power.

Services and Service Orientation

Consider, instead, treating "service orientation" as the operative concept. Ignore the "architecture," for now, and examine the system characteristics that the term SOA describes - a "service-oriented" one. To appreciate service orientation, it is important to first have a clear understanding of what is meant by "service," in non-technological terms.

To define it simply, "service" is a task performed by a provider, upon objects, for a client known as a service requestor. Each service has a well-defined interaction with the outside world in the form of input received, output delivered, and objects modified. In software, this interaction can be viewed as its interface. Services, though, are not just software constructs. Humans and even mechanical systems behave in similar ways. Additionally, each service should be expected to satisfy specific performance criteria, often called service levels.

Services, therefore, represent well-defined building blocks (i.e., objects) that can be useful at various levels of abstraction. These range from the top-level of the business, down through human and automated systems such as business or IT capabilities, processes, applications, and their enabling infrastructural and operational underpinnings. Service orientation, therefore, is a defining characteristic of any system designed to be viewed as a service, up to a system as large as an entire enterprise. The enterprise can be viewed as a network of these services.

Service Orientation as a Natural Model for Business

The "business" references in the definition above are significant. While IT professionals tend to focus on infrastructure or other primitive services, the concept of services is surprisingly easy for business personnel to embrace. "Services" have always been a way for businesses to deconstruct what they do, even if they didn't use that specific term (though many did and still do).

One of my favorite examples comes from the first film version of the "Miracle on 34th Street." Early in the film, an overhead shot shows a large room in the department store headquarters with rows of desks. There is constant motion as employees move in and out of the room, dropping off and picking up stacks of paper from the desks. Even without any specific knowledge of what they are doing, it is easy to see that the room is a service. The service being performed is completely contained, has defined inputs and outputs, performs a service function, and (one would imagine) has an expected service level in terms of volume, processing rate and quality. One can easily imagine many other pre-IT examples including such services as "sell a product," "collect payment," "schedule a delivery," or "generate a quote."

The point in telling this story is to raise the question, "What's different about business today?" The simple answer is: nothing. Businesses for the most part operate the same as they always have and services are a natural abstraction of the business.

In many cases, though, the concept of services and service functions don't map to the current way that businesses think of themselves. As a result of years of cross-training by IT staff - through the consumption of too many technical magazines and listening to too many consultants - business people now describe themselves in the context and language of applications, servers, databases, data, and business processes. Attend almost any business requirements session to see this in action. Business people don't list requirements or the services they need, even after pointed questioning from the IT professionals. Instead, they describe the end environment they want implemented, often listing databases, vendors, and platforms by name. This disconnect - between the business and IT organizations, and the systems and their purposes - can be profound.

One of the most significant opportunities afforded by the concept of service orientation is the chance to change the language that IT and the business use to communicate with each other. It is IT's chance to un-teach unnatural technological abstractions in favor of something that drives business and IT onto the same page. The strongest argument for a move to service orientation is to improve alignment between what the business needs, expects, and requests, and what IT can deliver.

Why Service Orientation?

Why, then, is so much market emphasis placed on reuse as a justification for the value of SOA instead of business value? When SOA is equated to Web services, and the subject of these Web services are fine-grained infrastructure services, reuse is an argument that can hold water for cost-justification purposes. With services as business constructs, as discussed above, that argument is not as strong.

The truth is that large, coarse-grained services that are meaningful to the business are not ideal candidates for reuse, at least in the traditional sense of replicated cross-functional instances. The cost-benefit equations often used for software reuse justification break down when there is only one user of the service. Instead, the justification for service orientation must be made at a higher level. Sure, nicely encapsulated services are easier to maintain and safer to test. The real value of service orientation, though, is in the ability to naturally map a service to the business capabilities it supports, and deliver to the business the ability to disassemble and reassemble collections of services to align with changing business strategies. Furthermore, the notion of "software as a service" opens up entirely new avenues to acquire, or to offer to others, a collection of business capabilities. Service orientation, then, is a way to construct a business to maximize the adaptive, agile enterprise. This, in turn, requires an enterprise-wide approach.

Services Orientation, the Enterprise Architecture Way

In the world of IT projects - with their requirements, schedules, annual budgets,

monthly progress reports and weekly operating measures - where is the center of gravity for this enterprise-wide approach? Enter "Enterprise Architecture", or EA. The principle objective of the EA function is to work with executive leadership to gain an appreciation for the big-picture enterprise perspective, to analyze the implications of strategy on the various aspects of the business and IT, and to guide each domain toward a target representing an integrated, balanced, coordinated and aligned enterprise, with all the piece-parts optimized to the objectives of the enterprise.

Enterprise Architecture is defined as a strategic management discipline that creates a single, business-driven, future-oriented perspective designed to deliver on the business vision and strategy of the enterprise. That perspective is represented by an integrated view of the desired business processes, information and services, as enabled by the technology. The primary design goal for enterprise architecture must be to enable efficient change in business processes and capabilities through the services that enable them.

In order to fully realize the benefits of services orientation - including meeting the requirements for enterprise strategic alignment, full delivery of the desired business capabilities, and the agility required to change and adapt strategy in response to a changing environment - the enterprise perspective needs to be a part of both strategic and tactical planning processes. If this is the case, there is a lower risk of uncoordinated service creation that can result in additional silos of process, information, technology, and applications.

A detailed description of how to organize and operate an effective EA program is beyond the scope of this article. However, one important element that must be present in every successful EA program is its collection of "roadmap" style documents describing the enterprise journey from its current state to one that fully delivers on the enterprise vision and strategy, known as the future state. These deliverables detail, in the form of information repositories, graphical models, principles, standards, lifecycles and roadmaps - each of the elements of the enterprise and their associated relationships. They also provide a framework for the hierarchy of abstractions that range from the highest level strategic alignment down to the lowest level implementation specifics (see Figure 1). EA work products must be created with a balance of large-scale perspective and lower-level detail appropriate to the enterprise's strategic objectives and its short-term tactical, operational, and pragmatic investment realities.

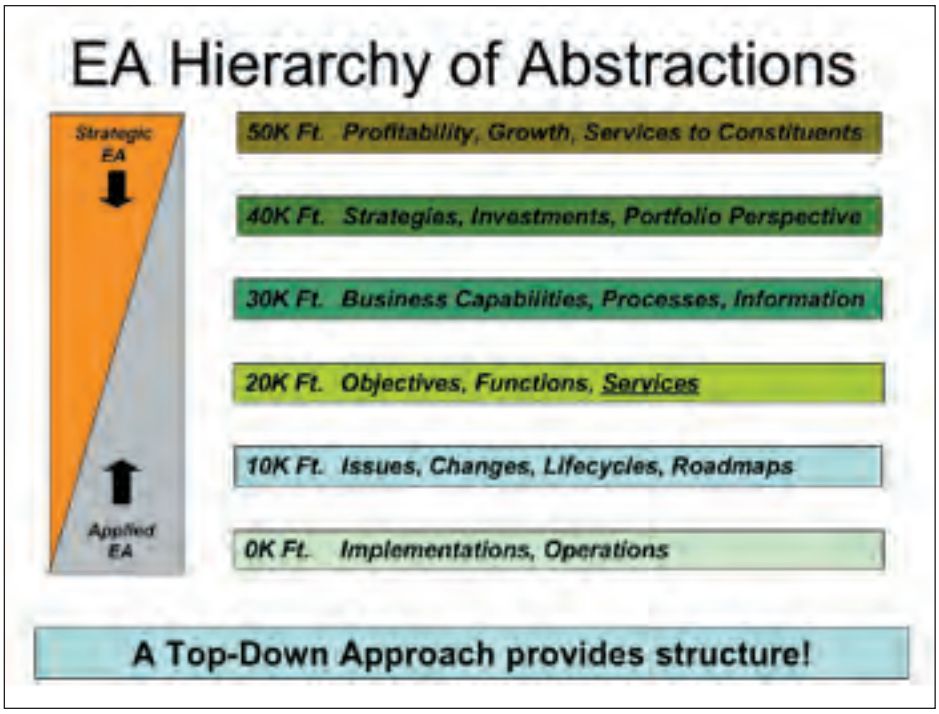


Figure 1: EA Hierarchy of Abstractions

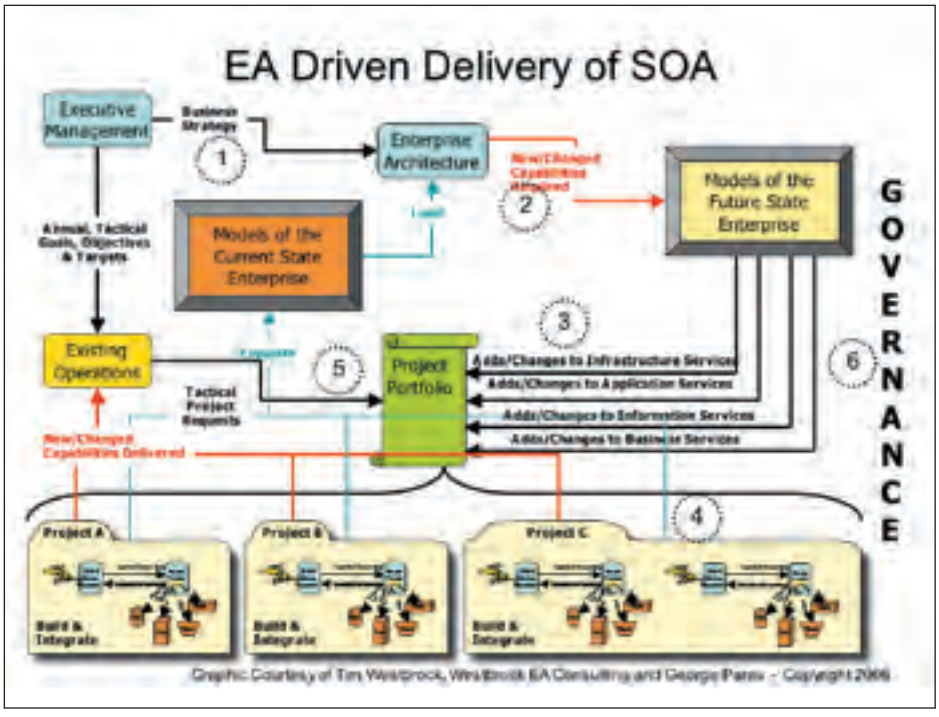


Figure 2: EA Driven Delivery of SOA

Balancing Enterprise Context and Detail with Services

In modern enterprise architectures, the ideal balance between strategic context at the enterprise level and the lower-level of detail that is required to assemble infrastructure, manage information, and build applications, pivots on the logi-

> **....these organizations focusing too narrowly on project-specific implementations....are at risk of never achieving the full value that the SOA concept could bring them**

cal concept of business services. Business services make up the linkage point between the constructs of business vision, strategy, and their requisite business capabilities with the implementation of those constructs as realized in applications, information models, processes, and applications.

EA Drives Services Orientation

Service orientation is one of many ways to conceptualize and deliver holistic enterprise architecture. It is not, in itself, the enterprise architecture, though it is hard to dispute that the concept of service orientation seems a natural fit for business, SOA is no more the enterprise architecture of corporations today than EAI was a few years ago. As IT trends evolve in the future, it is likely that SOA will not be the chosen approach. One of the only things we can really count on is that things will change. EA is much more than its resultant design and directional choices. It is the process that leads to design and directional decisions. The EA process embraces change by its very nature. EA teams of the future will continue to combine analysis of environmental factors, business vision and strategy, information technology trends and directions, investment priorities, and best-practices of the day in order to create the enterprise architecture directions relevant at that time.

Today, SOA and service orientation in general seem a logical choice for many enterprises. It is also worth noting that it may not be the choice for everyone. Whatever the degree/scope/size/rate of the SOA strategy, the way those services are implemented, how they grow and evolve, and even how they will eventually be replaced, are simply business as usual for the process of enterprise architecture.

Figure 2 diagrams the many aspects of the EA process that make it the ideal center of gravity for driving an SOA strategy.

Note the many feedback loops built into the process. A well-constructed EA process is dynamic and flexible to ongoing change. It is driven by executive management either through global business strategy changes or for the immediate needs of shorter-term tactical goals, objectives and targets. The numbered steps in Figure 2 illustrate the process:

1) Executive management - through declarations of large scale business vision and the portfolio of strategic intent - communicate the global business strategy to the enterprise.
2) The EA function interprets and analyzes the implications of that strategy, creating an integrated top-level model for the future state architecture that can deliver the required business capabilities in the way needed to optimize the operating parameters desired by management. These models reflect the intersection of business services, business functions, processes, information, technology, applications, and ultimately the required services. The resulting catalog of services and the portfolio of required enabling capabilities provide a powerful snapshot to guide current and future design decisions.
3) Comparing the currently installed elements of the enterprise to the future state model creates a clear picture of gaps in infrastructure, application, information and business services, suggesting possible projects meant to close those gaps and fulfill the vision.
4) The future-state EA provides directives in the form of policies, principles, technologies, techniques, and other models to guide proper development and synchronization of services. Some of these models include common information and data models, metadata definitions, common-use cases for business processes, etc., so that services created independently will work together by design.
5) As new tactical project suggestions arise from operational or other tactical requirements, they can be reconciled against the aggregate project portfolio created from the future-state EA. This provides the opportunity to respond in a way that will not jeopardize the enterprise strategic objectives.
6) The EA process is governed by an authoritative collection of approval, compliance, variance, escalation, and portfolio processes ensuring that the right services are implemented in the proper way, in the correct order, and with all dependencies properly accounted for.

The Potential of SOA – Delivered through EA

Imagine an enterprise constructed from business services aligned exactly to the required business capabilities. There are neither too many nor too few of these services; they mutually support one another, are flexible and adaptive; and they connect via well-defined, clean interfaces that are contextually consistent. These services hide unnecessary details, are simple to maintain and can optionally be sourced from multiple places. Silos don't exist, and the organization no longer wastes time, resources, or money on duplicate, overlapping, and ill conceived SOA projects. This is the reality of an EA-driven services oriented approach. EA provides the structure for top-down services orientation. Not there yet? An EA-driven approach models how the desired structure can be reached from the current-state architecture in the most direct, least painful way possible. ■

About the Author

George Paras has helped establish current thinking on EA discipline best practices and methods through his thought leadership, research, analysis, and evangelism of EA concepts as Chairman and featured speaker for the Enterprise Architectures Conferences (EAC) and as Editor-in-Chief of Architecture and Governance Magazine. He also currently operates a private EA coaching practice.

Web Services and SOA

Practical Interoperability Approaches, WS-Security and WS-Addressing Explained

WRITTEN BY **CLEMENS UTSCHIG, HEIDI BUELOW, AND JESUS RODRIGUEZ**

People sometimes ask what a service-oriented architecture enables today that could not have been done with the older, proprietary integration stacks of the past 5 to 15 years, such as those from Tibco, IBM, or Vitria. One such ability is the greater degree of interoperability between heterogeneous technology stacks that is made possible by the standards SOA is built on, such as Web services and BPEL. Although interoperability is only one facet of the SOA value proposition, it is one that has become increasingly more important, due in large part to the evolving IT environment, merger and acquisition activity, and increased partner connectivity.

Building business solutions for SOA requires the ability to secure data exchanged over a network, and control access to services in an environment where long-running business processes and asynchronous services are increasingly common. To meet these key requirements, two WS-* standards have moved to the forefront: WS-Security for authentication and encryption of service data, and WS-Addressing for correlation of messages exchanged with asynchronous services.

As these standards have begun to take hold, many commercial technologies have been introduced that add support for them. Likewise, many developers are implementing them in custom applications or with open source frameworks. Furthermore, standards that are logically layers above core Web services and security are referencing them. For example, the WS-BPEL specification is a Web service orchestration language with rich support for both synchronous and asynchronous services. BPEL, as it is commonly known, is highly complementary with WS-Security and WS-Addressing.

This article focuses on interoperability with asynchronous messaging and on the security challenges of using BPEL processes to orchestrate Web services deployed onto various technology plat-

forms. The specific example used is BPEL processes deployed on Oracle BPEL Process Manager, invoking services implemented with Microsoft .NET Windows Communication Foundation (WCF).

WS-BPEL and WS-Addressing Interoperability Challenges

For those readers who may not be versed in asynchronous service requirements, we will first provide some background on why a standard such as WS-Addressing is needed. The core Web services standards, including WSDL, SOAP, and XML schema are sufficient for synchronous service operations in which a client of a service sends a request and either gets no response at all (a "one-way" operation) or gets a result back as the output of the operation itself. In either case, the operation completes the interaction between the service client and the service itself.

However, for logical operations that may take a long time to complete, the concept of an asynchronous operation whereby the client initiates a service operation but does not wait for an immediate response makes sense. At some later time, the service will call the client back with the result of the operation – or with an error or exception message. In this case, the client must pass at least two pieces of information to the service: a location where the service can call the client back with the result, and an identifier of some sort that will allow the client to uniquely identify the operation with which the callback is associated. Early in the development of Web services standards, individual projects would include custom mechanisms for interacting with asynchronous services; however, this meant that developers had to explicitly code this support, and interoperability among toolkits was nonexistent.

WS-Addressing provides a standard for describing the mechanisms by which the information needed to interact reliably with asynchronous Web services can be exchanged. In the long term, this promises seamless interoperability, even for asynchronous services, between clients and services implemented on different technology stacks.

The main purpose of WS-Addressing is to incorporate message-addressing information into SOAP messages (for example, where the provider should send a response). SOAP is an envelope-encoding specification that represents Web service messages in a transport neutral format. However, SOAP itself does not provide any features that identify endpoints. The usual endpoints, such as message destination, fault destination, and message intermediary are delegated up to the transport layer. Combining WS-Addressing with SOAP creates a complete messaging specification. WS-Addressing specifies that address information be stored in SOAP headers in an independent manner, instead of embedding that information into the payload of the message itself. WS-Addressing is complemented by two other specifications, WS-Addressing SOAP Binding, and WS-Addressing WSDL Binding which specify how to represent the WS-Addressing properties into SOAP and WSDL respectively.

At a high level, WS-Addressing defines an EndpointReference construct to represent a Web service endpoint. It also defines a set of headers, ReplyTo, FaultTo, RelatesTo, and MessageId which are used to dynamically define an asynchronous message flow between endpoints.

BPEL relies on WS-Addressing to enhance endpoint representation and asynchronous Web services invocations. However, because WS-Addressing has evolved through several versions, interoperability can be a challenge. Today up to four different WS-Addressing versions are commonly used—three versions of the specification are named by their release date: the March 2003 version, the March 2004 version, and the August 2004 version, developed before the specification moved to W3C. The 1.0 version, recently completed in May 2006, was developed after the specification went under the umbrella of W3C. After moving to W3C, the specification split into multiple parts: a core specification, and two specifications that describe bindings for SOAP and WSDL.

Explicit vs. Implicit Addressing Mechanisms

Ideally, all server platforms would support all possible versions of WS-Addressing, but we are forced to live (and code) in the real world. At this time, many servers support one or more active WS-Addressing versions, but it is still all too possible that a service and client will be built on platforms that support incompatible WS-Addressing versions. However, interoperability is possible with a minimal amount of developer effort.

When the same WS-Addressing version is supported by both the process (client) and service layers, it is called "implicit" addressing because the developer need only state at the metadata level which version of WS-Addressing should be used to correlate asynchronous messages. In this case, WS-Addressing manipulation is completely transparent to the BPEL process itself, and the SOAP layer simply adds the requested SOAP headers as needed.

However, in order to interoperate with WS-Addressing versions not implicitly supported, a server should provide an

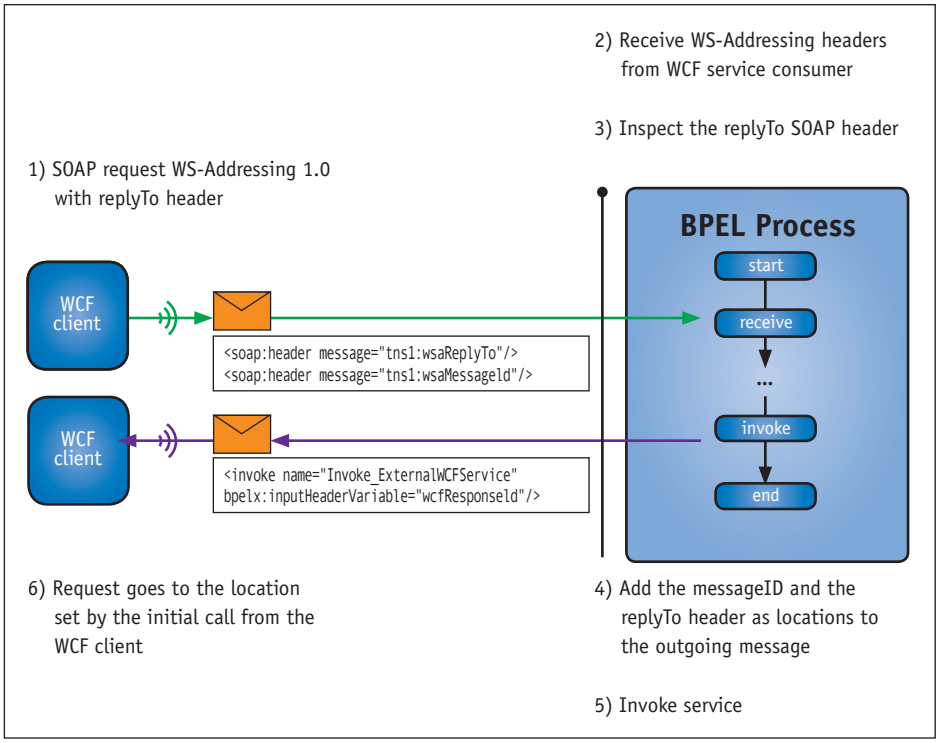


Figure 1: WS-Addressing interoperability

Although interoperability is only one facet of the SOA value proposition, it is one that has become increasingly more important, due in large part to the evolving IT environment, merger and acquisition activity, and increased partner connectivity

explicit mechanism by which developers can build and attach WS-Addressing to SOAP messages easily. The following section describes an explicit addressing mechanism used to achieve asynchronous service interoperability between Microsoft WCF using WS-Addressing 1.0 and Oracle BPEL Process Manager using WS-Addressing March 2003; however, the same principles should hold true for interoperability between any two BPEL and Web service toolkits.

WS-Addressing Interoperability Example: WCF and WS-Addressing

Microsoft's Windows Communication Foundation (WCF) represents the next generation of distributed programming and service-oriented technologies built on top of the Microsoft .NET platform for the upcoming Windows Vista release. WCF unifies the existing set of distributed programming technologies such as ASP .NET Web services, .NET Remoting, COM+, and so on, under a common, simple, and service-oriented programming model. WCF implements a vast set of WS-* protocols, including WS-Addressing 1.0.

To demonstrate explicit interoperability with WCF, we use Oracle BPEL Process Manager. It has had rich support for WS-Addressing for several years and includes WS-Addressing of March 2003, March 2004, and August 2004. This example uses BPEL with WS-Addressing March 2003 and WCF with WS-Addressing 1.0 to demonstrate explicit addressing support. Consider the WS-Addressing interoperability scenario illustrated in Figure 1.

The following explains the occurrences in Figure 1:

- A BPEL process exposes WS-Addressing headers on the process WSDL to expose a long-running process as an asynchronous service.
- A WCF client invokes the BPEL process, and passes the ReplyTo the WS-Addressing v1.0 (http://www.w3.org/TR/2005/CR-ws-addr-core-20050817/) header representing the URL of a WCF service that is expecting the operation response message. The client also sends a MessageID WS-Addressing v1.0 header to uniquely identify the request (step 1).
- The BPEL process receives the message, performs various operations, and uses the ReplyTo address to define a dynamic endpoint using the WS-Addressing 03/2003 (http://msdn.microsoft.com/webservices/webservices/default.aspx?pull=/library/en-us/dnglobspec/html/ws-addressing0303.asp). (steps 2–4).
- The BPEL process sends a reply message to the WCF service specified on the ReplyTo address, and passes the RelatesTo WS-Addressing v1.0 header to enable the WCF client to correlate the original request with the response (step 5).
- The WCF service receives the response message and is able to correlate it back to the request (step 6).

In this example, WCF uses WS-Addressing v1.0; however, the BPEL service uses the March 2003 version of WS-Addressing. To make this work, explicit strategies for interoperability need to be applied, as described below.

As part of the process, the WSDL, which represents the interface of the BPEL process, imports the WS-Addressing v1.0 XSD and declares the ReplyTo and MessageID headers as part of the binding section. It also declares messages of type ReplyTo, MessageID, and RelatesTo as variable types in the BPEL process, as shown in Listing 1. Note: By using this technique, we're explicitly declaring that the BPEL process expects the WS-Addressing ReplyTo and MessageID headers as part of the incoming message.

Based on the messages types in Listing 1, the BPEL process also defines variables of message type ReplyTo, MessageID, and RelatesTo:

```
<variable name="wcfServiceAddr" messageType="ns1:wsaReplyTo"/>
<variable name="wcfRequestId" messageType="ns1:wsaMessageId"/>
<variable name="wcfResponseId" messageType="ns1:wsaRelatesTo"/>
```

With this in place, we can assign the SOAP header information to them later on and vice versa. The next step is to populate these variables from incoming SOAP message:

```
<receive name="receiveInput" partnerLink="client"
                 portType="client:WCFAddr" operation="initiate"
                 variable="inputVariable" createInstance="yes"
                 bpelx:headerVariable="wcfServiceAddr wcfRequestId"
/>
```

By using bpelx:headerVariable (an extension of the WS-BPEL standard), the process code has access to the MessageID sent from the client as well as to its callback location.

Let's define a variable of type EndpointReference, which will provide the dynamic endpoint reference, needed for initiating the partnerLink later:

```
<variable name="wcfEndpoint" element="ns3:EndpointReference"/>
```

Note that the ns3 prefix is associated with the WS-Addressing 03/2003 namespace (xmlns:ns3=http://schemas.xmlsoap.org/ws/2003/03/addressing).

The next step is to populate the wcfEndpoint variable (defined in the previous step) using the ReplyTo header from wcfServiceAddr (Note the <copy> sections, marked yellow).

By using standard BPEL activities, these values are assigned by using a series of copy rules in an <assign> construct, as shown in Listing 2.

Assign the wcfEndpoint variable to the wcfService partnerLink, which represents an outgoing reference to a Web service. With this in place, the partnerLink knows which location to call:

```
<assign name="PartnerlinkWSAAssign">
```

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

```
  <copy>
    <from variable="wcfEndpoint"/>
    <to partnerLink="wcfService"/>
  </copy>
</assign>
```

In order to allow the client to correlate the request and response messages, we have to copy the value of the wcfRequestId (the unique MessageID) to wcfResponseId (RelatesTo):

```
<copy>
  <from variable="wcfRequestId" part="parameters" query="/ns2:Mes-
sageID"/>
  <to variable="wcfResponseId" part="parameters" query="/ns2:Relat-
esTo"/>
</copy>
```

The last step on the BPEL server-side is to use an invoke activity, which will call the WCF service (defined through the wcfService partnerLink), and to pass the RelatesTo header, available within the wcfResponseId variable. Make sure to use bpelx:inputHeaderVariable for this.

```
        <invoke name="Invoke_ExternalWCFService"
partnerLink="wcfService"
                portType="ns1:IOperationCallback" operation="SendResult"
                inputVariable="wcfRequest"
                bpelx:inputHeaderVariable="wcfResponseId"/>
```

After the server side, create a WCF client, which invokes the BPEL process through SOAP. Then create a WCF BindingElement that allows the use of WS-Addressing v1.0, and wrap the call to the BPEL process within an OperationContextScope to populate the WS-Addressing headers, as shown in Listing 3.

Testing the code in Listing 3 produces a SOAP message that follows. Note the <a:Address> field containing the service address:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
            xmlns:a="http://www.w3.org/2005/08/addressing">
       <s:Header>
              <a:Action s:mustUnderstand="1">http://tempuri.org/IOpera-
tionCallback/SendResult</a:Action>
              <a:ReplyTo>
                     <a:Address>WCF Service Address...</a:Address>
              </a:ReplyTo>
              <a:To s:mustUnderstand="1">Oracle BPEL Process Ad-
dress...</a:To>
              <a:MessageID>urn:uuid:847b546e-16e5-4ea9-8267-
b6fe559f0c1f</a:MessageID>
       </s:Header>
       <s:Body>Body</s:Body>
</s:Envelope>
```

After receiving the request, the BPEL process sends the response message to the WCF service specified within the ReplyTo header. The request and response messages are correlated using the RelatesTo header. On the client side, the WCF service operation includes the following code in order to receive the WS-Addressing v1.0 headers:

```
UniqueId relatesTo = OperationContext.Current.IncomingMessageHeaders.
RelatesTo;
```

Using these steps, messaging solutions can interoperate even with different versions of WS-Addressing.

BPEL and WS-Security Interoperability Challenges

In the second part of this article, we focus on WS-Security interoperability, because it is another key specification in any WS-* implementation. The main goal behind WS-Security is to enhance SOAP messaging by providing message integrity, confidentiality, and authentication. WS-Security also serves as the base standard to other WS-* protocols, such as WS-SecureConversation, WS-Trust, and WS-Federation. The key function of WS-Security is to provide a general purpose mechanism to associate security tokens with SOAP messages. In this scenario, security-related information is carried in the message itself, instead of in external artifacts. This is known as message-based security and can be used in conjunction with a transport layer security such as SSL.

Message Security vs. Transport Security

SSL has proven its effectiveness in securing resources through the internet. When dealing with Web services, however, SSL is a very coarse-grained tool and has several specific limitations:

- SSL is strictly tied to TCP-based transports such as HTTP. For Web services that use non-TCP transports such as MSMQ or SMTP SSL, it is not a viable option.
- SSL was designed for point-to-point communications. This makes it difficult to implement a routing scenario between Web services. In a routing scenario, the final receiver must be able to validate the original credentials. Extensibility using SSL becomes difficult to provide when those credentials are not stored in the message.
- SSL protects the messages on the wire (between the endpoints) but does not provide protection for the message at the endpoints. This allows access to the whole message upon gaining access to one of the endpoints.

In regards to business process integration, the WS-BPEL standard recommends the use of WS-Security to ensure authorization, authentication, confidentiality, and integrity. The implicit vs. explicit mechanisms explored earlier are also applicable to WS-Security interoperability.

WS-Security Interoperability Example

WCF provides an up-to-date implementation of the WS-* security stack including WS-Security, WS-SecureConversation, WS-Trust, and WS-Federation, allowing developers to implement transport and message security scenarios. Specifically for WS-Security, WCF supports a number of token types, including UserName and Password.

The WS-Security UserNameToken profile provides a standards-based way to send user credentials, which enables applications and platforms to unify their approach. A message-based security approach moves credentials outside the actual operation and into SOAP headers, which makes it possible to alter authentication without touching the operation.

The example used in this article consists of a WCF service that is being invoked from a BPEL process. Before the first call is made, the

BPEL process must authenticate to the WCF service using WS-Security and UserName and Password.

The code in Listing 4 shows the target WCF service.

The bindings and behaviors shown in Listing 5 enforce the use of UserName and Password for the authentication.

Consider the sample code below, which shows a custom class that provides the UserName and Password validation:

```
public class MyUserNameValidator :UserNamePasswordValidator
{
    public override void Validate(string userName, string password)
    {
        if (null == userName || null == password)
        {
            throw new ArgumentNullException();
        }
        if (!(userName == “user1” && password == “password1”))
        {
            throw new SecurityTokenException (“Unknown Username or
Incorrect Password”);
        }
    }
}
```

Next we implement the BPEL process that invokes the WCF service. The first step is to create a BPEL partnerLink that represents the WCF service. Here is the partnerLink declaration.

```
<plnk:partnerLinkType name=”ServiceSoap_PL”>
        <plnk:role name=”ServiceSoap_Role”>
                    <plnk:portType name=”tns:MathService_ASPNetSoap”/>
        </plnk:role>
</plnk:partnerLinkType>
```

In the next step, we provide the required WS-Security headers as part of the <invoke> activity. Using bpelx:inputHeaderVariable (an extension to WS-BPEL standard) we can pass SOAP headers as part of the outgoing message. This is the primary way to send the SOAP-specific headers. The bpelx:inputHeaderVariable extension is specific to Oracle BPEL Process Manager, which also provides deployment descriptor properties (wsseHeaders, wsseUsername, wssePassword) to handle WS-Security headers implicitly.

Listing 6 highlights the code fragments dealing with WS-Security headers. Note especially the inputHeaderVariable that is passed along the invoke activity, as well as wsse:Security.

Running the BPEL process produces the SOAP message shown in Listing 7 when the WCF service is invoked. Pay close attention to the marked section.

This technique allows the creation of BPEL processes that can truly interoperate with WCF services by using WS-Security. The flexibility introduced by manipulating header variables in BPEL allows developers to create interoperability explicitly, when implicit means are not sufficient.

Conclusion

In this article we demonstrated how interoperability between Microsoft .NET WCF and WS-BPEL can be achieved by using explicit means.

SOA applications that require long-running business processes with security and messaging can be built quickly and easily with Web services as the underlying specifications. Web services promise heterogeneous interoperability, and while we anticipate implicit interoperability approaches to be sufficient in the near future, explicit capabilities are still needed in the short term. Such interoperability is real today, as shown by the examples in this article that use Microsoft WCF and Oracle BPEL Process Manager.

About the Authors

Jesus Rodriguez is the Chief Software Architect at Two Connect Inc., a Microsoft Gold partner based in Miami, Florida. He is also a Microsoft BizTalk Server MVP. You can contact Jesus at jrodriguez@twoconnect.com or through his blog at http://weblogs.asp.net/gsusx

Clemens Utschig works within the Oracle SOA Product Management team responsible for security and cross-product integration. Clemens can be reached at clemens.utschig@oracle.com, or through his blog at http://clemensblog.blogspot.com/

Heidi Buelow is a product manager for the Oracle SOA Suite. Heidi can be reached at heidi.buelow@oracle.com

Listing 1

```
<definitions xmlns:tns1=”http://xmlns.oracle.com/WCFAddr”
             xmlns=”http://schemas.xmlsoap.org/wsdl/”
             xmlns:ns1=”http://tempuri.org/”
             xmlns:plnk=”http://schemas.xmlsoap.org/ws/2003/05/part-
ner-link/”
             xmlns:ns2=”http://www.w3.org/2005/08/addressing”
             xmlns:client=”http://xmlns.oracle.com/WCFAddr”
             xmlns:wsa10=”http://www.w3.org/2005/08/addressing”
             xmlns:soap=”http://schemas.xmlsoap.org/wsdl/soap/”
             targetNamespace=”http://xmlns.oracle.com/WCFAddr”
name=”WCFAddr”>
   <types>
      <xsd:schema xmlns:xsd=”http://www.w3.org/2001/XMLSchema”>
         <xsd:import namespace=”http://www.w3.org/2005/08/addressing”
                     schemaLocation=”http://www.w3.org/2006/03/ad-
dressing/ws-addr.xsd”/>
      </xsd:schema>
      <!-- other types -->
   </types>

   <!-- definition of messages -->
   <message name=”wsaRelatesTo”>
      <part name=”parameters” element=”wsa10:RelatesTo”/>
   </message>
   <message name=”wsaReplyTo”>
      <part name=”parameters” element=”wsa10:ReplyTo”/>
   </message>
   <message name=”wsaMessageId”>
      <part name=”parameters” element=”wsa10:MessageID”/>
```

```
   </message>

   <!-- Port types -->

   <!-- Binding -->
   <binding name="WCFAddrBinding" type="tns1:WCFAddr">
      <soap:binding style="document"
                    transport="http://schemas.xmlsoap.org/soap/
http"/>
      <operation name="initiate">
         <soap:operation soapAction="initiate" style="document"/>
         <input>
            <soap:header message="tns1:wsaReplyTo" part="parameters"
use="literal"/>
            <soap:header message="tns1:wsaMessageId"
part="parameters" use="literal"/>
            <soap:body use="literal"/>
         </input>
         <output>
            <soap:body use="literal"/>
         </output>
      </operation>
   </binding>
</definitions>
```

Listing 2

```
<assign name="endpointAssign">
    <copy>
        <from>
            <EndpointReference xmlns="http://schemas.xmlsoap.org/
ws/2003/03/addressing"
                               xmlns:ns1="http://tempuri.org/">
                <Address/>
                <ServiceName/>
            </EndpointReference>
        </from>
        <to variable="wcfEndpoint"/>
    </copy>
    <copy>
        <from variable="wcfServiceAddr" part="parameters"
              query="/ns2:ReplyTo/ns2:Address"/>
        <to variable="wcfEndpoint" query="/ns3:EndpointReference/ns3:
Address"/>
    </copy>
    <copy>
        <from expression="'ns1:wcfService'"/>
        <to variable="wcfEndpoint"
            query="/ns3:EndpointReference/ns3:ServiceName"/>
    </copy>
</assign>
```

Listing 3

```
HttpTransportBindingElement addrBindingElement = new HttpTransport-
BindingElement();
addrBindingElement.ManualAddressing = true;

TextMessageEncodingBindingElement encElement = new TextMessageEncod-
ingBindingElement();
encElement.AddressingVersion = AddressingVersion.WSAddressing10;
encElement.MessageVersion = MessageVersion.Soap11WSAddressing10;
CustomBinding opbinding = new CustomBinding(encElement, addrBinding-
Element);
EndpointAddress opendpoint = new EndpointAddress(WCF Service Ad-
dress...);
ChannelFactory<IOperationCallback> factory = new ChannelFactory<IOper
ationCallback>(opbinding, opendpoint);
IOperationCallback op = factory.CreateChannel();

using (new OperationContextScope((IContextChannel)op))
{
    MessageVersion CurrentVersion = OperationContext.Current.Outgo-
ingMessageHeaders.MessageVersion;
    OperationContext.Current.OutgoingMessageHeaders.ReplyTo = new
EndpointAddress(opendpoint);
    OperationContext.Current.OutgoingMessageHeaders.To = new
Uri(Oracle BPEL Process Address...);
    OperationContext.Current.OutgoingMessageHeaders.MessageId = new
UniqueId();
    //Execute operation...
 }
```

Listing 4

```
public interface Icalculator
{
    [OperationContract]
    int Add(int param1, int param2);
}

[ServiceBehavior(IncludeExceptionDetailInFaults=true)]
public class CalculatorService : ICalculator
{
    public int Add(int param1, int param2)
    {
        Int result = n1 + n2;
        return result;
    }
}
```

Listing 5

```
   <bindings>
      <wsHttpBinding>
         <!-- Username binding -->
         <binding name="Binding1">
            <security mode="Message">
               <message clientCredentialType="UserName"/>
            </security>
         </binding>
      </wsHttpBinding>
   </bindings>

   <behaviors>
      <serviceBehaviors>
         <behavior name="CalculatorServiceBehavior"
                   includeExceptionDetailInFaults="True">
            <serviceCredentials>
               <!--
                    The serviceCredentials behavior allows one to
specify a custom
```

```
                    validator for username/password combinations.
                -->
               <userNameAuthentication userNamePasswordValidationMod
e="Custom"
  customUserNamePasswordValidatorType="OracleWCF.Samples.CalculatorSe
rvice+MyUserNameValidator, service"/>
            </serviceCredentials>
         </behavior>
      </serviceBehaviors>
   </behaviors>
```

Listing 6

```
<process name="InteropPrj" targetNamespace="http://xmlns.oracle.
com/InteropPrj"
         xmlns="http://schemas.xmlsoap.org/ws/2003/03/business-pro-
cess/"
         xmlns:xp20="http://www.oracle.com/XSL/Transform/java/oracle.
tip.pc.services.functions.Xpath20"
         xmlns:bpws="http://schemas.xmlsoap.org/ws/2003/03/business-
process/"
         xmlns:ns4="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-secext-1.0.xsd"
         xmlns:ns1="http://tempuri.org/"
         xmlns:ldap="http://schemas.oracle.com/xpath/extension/ldap"
         xmlns:xsd="http://www.w3.org/2001/XMLSchema"
         xmlns:client="http://xmlns.oracle.com/InteropPrj"
         xmlns:bpelx="http://schemas.oracle.com/bpel/extension"
         xmlns:ora="http://schemas.oracle.com/xpath/extension"
         xmlns:orcl="http://www.oracle.com/XSL/Transform/java/oracle.
tip.pc.services.functions.ExtFunc">
        <partnerLinks>
                <partnerLink name="client" partnerLinkType="client:
InteropPrj"
                             myRole="InteropPrjProvider"/>
                <partnerLink name="SecureSrv"
partnerRole="ServiceSoap_Role"
                             partnerLinkType="ns1:ServiceSoap_PL"/>
        </partnerLinks>
        <variables>
                <!-- Reference to the message passed as input during
initiation -->
                <variable name="inputVariable"
                          messageType="client:InteropPrjRequestMes-
sage"/>
                <!--
                      Reference to the message that will be returned
to the requester
                      -->
                <variable name="outputVariable"
                          messageType="client:InteropPrjResponseMes-
sage"/>
                <variable name="msgIn" messageType="ns1:AddSoapIn"/>
                <variable name="msgout" messageType="ns1:
AddSoapOut"/>
                <variable name="wsseheaders" element="ns4:Security"/>
        </variables>
        <sequence name="main">
                <!-- Receive steps -->
                <assign name="Assign_1">
                        ...Copy Rules
                        <copy>
                                <from>
                                        <wsse:Security
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd">
                                                <wsse:UsernameToken
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd">
                                                        <wsse:
Username>User1</wsse:Username>
                                                        <wsse:Pass-
word Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
username-token-profile-1.0#PasswordText">Password1</wsse:Password>
                                                </wsse:UsernameToken>
                                        </wsse:Security>
                                </from>
                                <to variable="wsseheaders"/>
                        </copy>
                </assign>
                <invoke name="Invoke_1" partnerLink="SecureSrv"
                        portType="ns1:MathService_ASPNetSoap"
operation="Add"
                        inputVariable="msgIn" outputVariable="msgout"
                        bpelx:inputHeaderVariable="wsseheaders"/>
                <!-- Final steps -->
        </sequence>
</process>
```

Listing 7

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
               xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
               xmlns:xsd="http://www.w3.org/2001/XMLSchema">
        <soap:Header>
                <wsse:Security soap:mustUnderstand="1"
                               xmlns:wsse="http://docs.oasis-open.
org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
                               xmlns:wsu="http://docs.oasis-open.
org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
                        <wsse:UsernameToken xmlns:wsu="http://docs.
oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd"
                                            wsu:Id="SecurityToken-
941458a9-6eff-4bc7-9d56-c6833ff249d4">
                                <wsse:Username>ok</wsse:Username>
                                <wsse:Password Type="http://docs.
oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-
1.0#PasswordText">ko</wsse:Password>
                                <wsse:Nonce>yHOCbdMAoejJz8l1PDi3Fw==
</wsse:Nonce>
                                <wsu:Created>2006-02-18T15:52:32Z</
wsu:Created>
                        </wsse:UsernameToken>
                </wsse:Security>
        </soap:Header>
        <soap:Body>
                <Add xmlns="http://tempuri.org/">
                        <param1>12</param1>
                        <param2>23</param2>
                </Add>
        </soap:Body>
</soap:Envelope>
```

WS-BPEL 2.0: Not Backward Compatible?

Orchestration is a necessity

WRITTEN BY **DAVID S. LINTHICUM**

Let's face it, WS-BPEL 1.1 was not a great standard, and left so much out that many end users and vendors found it useless. In response, the vendors put a ton of proprietary extensions in their BPEL 1.1–based products, thus diluting its value to the point of "Why bother?" This was a dirty little secret in the world of SOA. Considering that BPEL 2.0 is on the horizon, I think it's time we began to talk about what's really there, how you can fix it, and what you need to do to get from point A to point B.

What's most frustrating about the issues here is that orchestration is indeed a core feature of SOA...the configuration component that makes orchestration that part of the architecture providing agility. Orchestration, at least the notion, is a necessity if you are building an SOA. It's the layer that creates business solutions from the vast array of services and information flows found in new and existing systems. Orchestration is a god-like control mechanism that's able to put our SOA to work, as well as provide a point of control. Orchestration layers allow you to change the way your business functions, as needed, to define or redefine any business process on-the-fly. This provides the business with the flexibility and agility needed, and is the core value of SOA.

Ranting

While I've been ranting and raving about this for some time, Dave Chappell does a much better job of explaining the limitations (from his blog http://www.davidchappell.com/blog/2005_11_01_weblog).

"Promoting BPEL's portability helps significantly in the first of these goals, since customers like the idea of not being locked in to a single vendor. But actually making customers successful has typically required extending BPEL in proprietary ways, which works against the language's promised portability. While BPEL purists might argue that all of these extensions should be provided via programming language-neutral web services interfaces, this isn't what's actually happening in the products."

With huge investments in BPEL by the larger SOA players out there, the dirty little secret is that BPEL 1.1 never really worked as advertised, and the amount of custom and proprietary extensions required to make the technology useful meant that the money, time, and effort was pretty much wasted if "standards and portability" were the goals.

The hype behind the standard, and the larger vendors supporting it, has really pushed the notion of BPEL in the last few years. However, those who have had to build solutions using the base 1.1 technology have run into many roadblocks, and thus the vendors have responded with their own unique extensions and toolsets. Are we employing a standard that's living and breathing, allowing us to protect our investment in our work, or just an instance of a vendor's tool? This is now a key question to answer.

BPEL 2.0

Enter WS-BPEL 2.0, and another opportunity for vendors to get BPEL right. You can find the draft specification on the OASIS site (http://docs.oasis-open.org/wsbpel/2.0/wsbpel-specification-draft.pdf), which I read, and discovered that this spec was much improved, but many issues still remain. For instance, there are considerable differences in WS-BPEL 2.0 compared to its previous 1.1 version. The major differences include syntax changes to the language, the inclusion of new features including parallel for-each, and modifications to the semantics of existing constructions, such as compensation handling. There are a few more, and I urge you to read both the 1.1 and 2.0 specs before diving into BPEL, or assessing how deep you're in already.

What if I already implemented orchestrations using BPEL 1.1, what do I do now? In short, you purchased Beta and the world is moving to VHS. I thought that this article by BEA's Alexandre Alves (http://webservices.sys-con.com/read/155617_1.htm) did a good job of summing things up.

"For all but the simplest business processes, migration from BPEL 1.1 to BPEL 2.0 is not an easy task. Some of the syntactic changes can be automated as shown [in the Alexandre Alves article], however the semantic differences, especially when dealing with links, messaging, compensation handling, and data manipulation, will demand a comprehensive and time-consuming process."

What bothers me most about this issue is not the fact that the standard, at least in some respects, is letting SOA down, but that orchestration is such a powerful need within SOAs, and there are few other alternatives that offer a better approach, standardized or not. Thus, end users who are tasked with building solutions using emerging SOA standards such as BPEL should begin to lean on their vendors now, as well as get more involved with their respective standards organizations.

I suspect the BPEL sales guys will get some angry calls around Christmastime from their users, perhaps more so considering the use of many proprietary extensions to get around the limitations of BPEL 1.1. Not a good way to start a standard. I guess that's why my old boss told me never to buy products until release 3.0. ■

About the Author

David S. Linthicum is the president and CEO of BRIDGEWERX, and the author of several books on application integration and service-oriented architecture, and the host of the SOA Expert Podcast.

david@bridgewerx.com

REGISTER TODAY AND SAVE!

Rich Internet Applications: AJAX, Flash, Web 2.0 and Beyond...

www.AjaxWorldExpo.com

AJAXWORLD™ EAST

CONFERENCE & EXPO

NEW YORK CITY

THE ROOSEVELT HOTEL LOCATED AT MADISON & 45th

SYS-CON Events is proud to announce the AjaxWorld East Conference 2007!

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 unique, timely conference – especially the web designers and developers building those experiences, and those who manage them.

BEING HELD MARCH 19 - 21, 2007!

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 in learning about a wide range of RIA topics that can help them achieve business value.

Governance: The Last Mile of SOA

So what exactly is SOA governance, why is it important, and what needs should it address?

WRITTEN BY **GARY SO**

The phrase "SOA governance" is more likely to generate a sense of apprehension among IT professionals than any warm feelings. After all, most companies are still in the early stages of SOA adoption and so the practice of governance - and likely the concept itself - will be new territory. Yet, if companies are to realize any meaningful and lasting impact from SOA, then governance is a fact of life with which organizations are going to have to become comfortable. Governance - more than any other factor over the long term - will make the difference between SOA success and failure, and proficiency in governing the SOA environment will distinguish IT leaders from laggards.

SOA Governance Defined

While the term may be novel, the act of governance and the underlying rationale will be very familiar to IT. In most organizations, virtually every IT resource and process will have some level of governance associated with it in the form of policies, rules, and controls that define how a particular asset is managed and utilized, or parameters around how a certain IT function is performed. For example, IT management will have established policies for how projects are initiated, funded, prioritized, and staffed. An enterprise application package such as SAP or Siebel will have rules and operational procedures concerning issues such as how change requests are processed and approved, which employees are allowed access to the system and their level of access, and how new versions of the software are rolled out. Collectively, the act of establishing and enacting these rules falls under the broad umbrella of IT governance, in order to institutionalize discipline and maturity in IT processes so as to gain greater control and economies.

SOA governance, then, is the subset of IT governance related to establishing policies, controls, and enforcement mechanisms - within the context of the activities and constructs associated

with SOA - similar to those that exist for managing and controlling other aspects of IT. Initially, the concept of SOA governance was applied narrowly to the development and use of Web services, for example, validating the conformance of a Web service with specific standards or managing Web services in the SOA runtime environment. Today, however, it is generally understood that SOA governance has a broader meaning that spans SOA architecture as well the governance of services across the entire implementation life-cycle.

Before exploring the scope of SOA governance in further detail, it is worth considering why governance is so fundamental to SOA's success.

The Case for SOA Governance

SOA provides organizations with a powerful opportunity to transform the way in which they deploy information technology. This includes benefits for both the IT organization and the line of business. For IT, these benefits include:

- Increased standardization of systems and operational practices
- An improved ability to consolidate existing computing assets to eliminate redundant functionality, while preserving needed features and services
- The development of new systems offering higher levels of visibility, control, and compliance
- The ability to evolve the systems infrastructure incrementally

For the business, SOA holds the promise of increasing the flexibility of IT resources, improving the visibility and adaptability of business processes, and increasing the organization's agility and responsiveness to market conditions.

By definition, however, enterprise-level SOA means a dramatic increase in the number of moving parts in the environment. The trade-off is that this increase in the number of moving parts is accompanied by a corresponding increase in complexity.

Each service is an asset that has to be properly designed to be useful within a larger portfolio of business services; each must be versioned, secured, managed, and monitored to ensure that it performs with the expected quality of service. IT is accustomed to dealing with these issues on an aggregate level - at the scope of an application package, for instance - but SOA creates a challenge that is an order of magnitude greater by requiring these issues to be addressed at the level of individual services. Further compounding the challenge is the fact that services use and interact with other services, increasing the complexity of change management, testing, and deployment. How do you recreate a network of interdependent services in an isolated environment to perform regression testing, for example?

Thus, success with SOA is highly correlated with an organization's ability to manage complexity and to develop the necessary maturity and infrastructure - in the form of governance control and enforcement mechanisms - to maintain order over the SOA environment. Without effective SOA governance, organizations will experience some predictable challenges, including:

- A fragile and brittle SOA implementation
- Services that cannot easily be reused because they are unknown to developers or because they were not designed with reuse in mind
- Lack of confidence in services as enterprise assets which results in a "build it myself" mentality (further compounding the lack of reuse with redundancy and unnecessary duplication of functionality)
- Security breaches that cannot easily be traced
- Unpredictable performance

Failure to overcome these challenges will cause a drag on an organization's continued deployment of SOA and, in all likelihood, result in SOA being discarded as a failure. Fortunately, the majority of organizations exploring SOA recognizes the need to address these issues, and companies who are serious about SOA are generally serious about governance, too.

So, having established the importance of governance in making SOA successful, on what fronts is SOA governance required?

Architecture Governance

The first requirement of SOA governance is "architecture governance." Architecture governance is necessary to ensure that SOA as architecture evolves by design and not by accident. To the extent that it mirrors governance requirements in other areas of IT architecture, SOA architecture governance practices can be adapted from existing Enterprise Architecture processes. These include:

- Establishing corporate technology standards
- Defining the high-level SOA architecture and topology , as well as the infrastructure capabilities that the SOA should incorporate
- Determining the SOA platform strategy and making decisions about particular vendor products and technologies
- Specifying the management, operations, and quality of service—security, reliability, and availability—characteristics of the SOA
- Establishing criteria for SOA project design reviews

In addition, a key aspect of SOA architecture governance is defining a roadmap that will guide the smooth and orderly evolution of the architecture over time. The majority of corporate SOA strategies will involve overlaying and transforming the existing systems architecture in stages, rather than a wholesale replacement of the current infrastructure. Governance is needed to ensure that decisions made along the way align in a consistent direction and maintain the coherency of the SOA architecture.

SOA architecture governance is a discipline that architects and IT organizations will acquire with relative ease if they have experience in parallel areas of distributed systems architecture, for example, establishing the architecture of an enterprise-wide integration infrastructure. In contrast, when it comes to the second frontier of SOA governance, organizations are likely to encounter certain requirements for the first time.

Service Life-Cycle Governance

A fundamental underpinning of SOA is that it involves the creation of discrete, well-defined services that exist not only as building blocks of larger systems and applications, but as independent entities. For the first time within the IT environment, SOA exposes standalone application functionality at a fine-grained level of granularity, thus necessitating a new form of governance—service-level governance.

Service-level governance applies at the level of individual services and covers a wide gamut of requirements and situations. A useful approach to categorize the scope of activities associated with service-level governance is to consider the life-cycle of a service - from its design, to its use in a runtime environment, to ongoing management and change of the service - as well as the constituencies who have a vested interest in governance of services.

Design-Time Governance

Design-time governance is primarily an IT development function that involves the application of rules for governing the definition and creation of Web services. Policies might include ensuring that services are technically correct and valid, and that they conform to relevant organizational and industry standards. Examples of this type of validation might include checking that a service is compliant with the Web Services Interoperability (WS-I) profiles - usage guidelines that ensure Web services implemented on different platforms are interoperable - by automatically verifying service schemas, validating namespaces, and other such controls.

If an organization has an SOA governance infrastructure in place - in the form of software that facilitates the implementation of SOA governance practices - these checks can be invoked automatically when developers check services into a registry. In addition, approval and notification workflows can be triggered by a governance-enabled registry to ensure that services pass through predefined review and approval steps so that they meet architectural and organizational standards for business function encapsulation, reusability, reliability, and so on. By ensuring that these reviews are performed by appropriate members of the organization, it becomes possible to manage the quality and coherency of the service portfolio effectively.

Design-time governance will be of most concern to business analysts, architects, and developers building services. Key issues to consider include:

- Determining the fitness of a service as an enterprise class asset (where fitness is a function of the business functionality that is encapsulated, the likelihood of reuse, and the importance of the service within the overall portfolio of services)
- Identifying which services to build against the backlog of business requirements
- Ensuring the strategic design of business services and ensuring that their interfaces and implementation conform to established design patterns and other corporate standards and best practices
- Establishing the governance standards to which different categories of services will be held, understanding that different levels of governance will be appropriate for different classes of services (internal use vs. services exposed to business partners, for example)

Other capabilities of design-time governance include fine-grained access control over assets in the registry, so that only authorized users are able to publish, search, and view services. In addition, the ability to label services and classify providers and consumers makes it possible to have some services visible to certain classes of service consumers and not others, a feature that is particularly important for partitioning access in a shared services model.

Runtime Governance

Runtime governance is primarily of interest to IT operations. Governance at runtime revolves around the definition and enforcement of policies for controlling the deployment, utilization, and operation of deployed services. These runtime policies typically relate to non-functional requirements such as trust enablement, quality of service management, and compliance validation. Examples of runtime governance include:

- Checking a service against a set of rules before it is deployed into production, for example, to ensure that only a certain message transport or specific schemas are used by services
- Securing services so that they are accessible only to authorized consumers possessing the appropriate permissions, and that data is encrypted if required
- Validating that services operate in compliance with prescribed corporate standards, in effect, to confirm that a service is not just designed to be compliant, but that its implementation is actually compliant

A more specific case of runtime governance involves service-level monitoring and reporting. In order for the runtime SOA infrastructure to assess whether a given service is performing at the required level for a given consumer - in terms of response time, throughput, and availability - it is necessary to have an explicitly defined service level agreement (SLA) between the service consumer and provider. SLAs can be expressed in terms of service contracts between consumer-provider pairs, and they establish the reference points for compliance monitoring and reporting by the SOA runtime environment. By tracking the actual performance of a service and comparing it to the requirements specified in the SLA, the system can identify non-compliant services and prompt remedial action (for example, automatically instantiating another instance of the service to improve load-balancing or alerting operations staff).

Runtime governance relies on an SOA infrastructure that is able to exercise policy enforcement in a way that is transparent to, and independent of, the service providers and consumers. This is generally achieved through an agent or intermediary that resides between provider and consumer, and a registry that addresses both the needs of service discovery as well as policy enforcement. The intermediary interacts with the registry to find services and their runtime policies, and enforces the policies during the execution of the service (See Figure 1).

Intermediaries serve as the policy enforcement points for SOA. Without SOA, the ability to control and manage applications is restricted both by the scope and the capabilities of the underlying platform. Thus, when different applications are integrated, it is generally infeasible to apply a common policy context to the integrated result. A typical challenge is enforcing access security when two applications with different user communities are integrated.

Figure 1

For example, application A automatically draws data from application B, but application A users are not authorized to use application B. How do you now control the data that users have gained access to within application A? With intermediation, it becomes possible for a distributed network of services to share a common policy-managed context. This is a powerful capability which emerges as a direct result of SOA.

Change Time Governance

Change is inevitable and at some point, services deployed in the runtime environment will have to be changed to adapt to new business requirements. Since the majority of services will be designed once and then modified several times over their lifespan, change time governance - the act of managing services through the cycle of change - is arguably more important in the long term than design time governance. Change time governance requirements and considerations include:

- Understanding inter-service relationships and dependencies
- Performing impact analysis to determine the implications of changing a particular service within the runtime environment
- Managing the rollout of services into the existing runtime environment
- Managing service custody transfers through the design, coding, testing, and deployment stages
- Managing changes to existing policies and service-level agreements

An important aspect of change time governance is involvement from the line of business. While easy to overlook when looking at governance from an IT-centric perspective, this need arises from the fact that services exist to support business functions as well as the inter-organizational relationships and dependencies that are implicit in SOA, particularly when services are exposed and invoked across organizational and corporate boundaries. Since changes are generally initiated and driven by business requirements, business users need to be intrinsic participants in the governance lifecycle.

Consider, for example, a service that enables vendor managed inventory. A change in the service, say, to reduce inventory data latency from a week to one day, will involve not only technical changes to the service (and possibly source applications and databases), but more importantly, changes in the business relationship between the company and its suppliers. A comprehensive governance strategy is needed to ensure coordination between the technical and business-level changes. As with design time and runtime, this change time governance can be facilitated by a governance-enabled SOA infrastructure that allows change time policies to be defined and enforced through service contracts and workflows.

Getting Started

For the many companies who are just getting started with SOA, at what point should consideration of governance come into play and where should it be focused?

One priority is to make an SOA governance strategy a subset of any larger SOA strategy, and to ensure that governance capability related milestones are synchronized with SOA adoption milestones, so that you do not end up trying to retro-fit governance after the fact. Ideally, the right time for governance is before you put any services into place so that any SOA pilot proves out not only the approach itself, but also the related governance practices along the way.

As part of the SOA governance strategy, there should be a roadmap that defines which specific governance capabilities the organization wants to put into place and when they will be implemented. Typically, companies will first want to pay attention to the SOA architecture and to design time governance policies in order to get the SOA journey off on the right foot. In fact, since architecture and governance are what separate a collection of Web services from being a true SOA, organizations that went down the path of simply developing and exposing Web services without the appropriate controls or a broader architecture will want to consider an investment in governance.

Finally, while governance is not a solution that comes in a box, having the right technology framework makes it easier - and in some situations is the only feasible way - to enforce policies and controls. As explained in this article, this framework should include mechanisms for defining and enforcing policies and service contracts through the service life cycle of workflows, intermediaries, and other automated means. By establishing the right balance between organizational practices and supporting technologies, companies will be able to turn the concept of SOA governance into a practical reality. ■

About the Author

Gary So is Vice President, Office of the Chief Technology Officer, at webMethods, Inc, where he is responsible for advancing the company's status as a recognized industry thought leader. Gary has a Masters degree in Computer Engineering from the University of Toronto.

Nimaya Accepted into IBM's SOA Specialty Practice

(*Fairfax, VA*) – Nimaya, a leader in Integrated Customer Management (ICM), has announced its acceptance to the IBM SOA Specialty after successfully completing IBM's rigorous SOA (service oriented architecture) technical and business requirements. With this achievement Nimaya's CustomerGrid integrates with the IBM SOA Foundation, to create a single, integrated platform, combining the industry's top application server and integration capabilities with Nimaya's innovative technology.

The IBM SOA Specialty provides technical enablement and a skills-building roadmap for partners who want to access the growing SOA market.

For more information, visit www.nimaya.com.

International Turnkey Systems Selects IONA for SOA Enablement

(*Dublin & Waltham, MA*) – IONA® Technologies, a leader in high-performance Service-Oriented Architecture (SOA) infrastructure solutions for mission-critical IT environments, has announced that International Turnkey Systems (ITS), has selected ArtixTM, IONA's extensible Enterprise Service Bus (ESB), to deliver Service Oriented Architecture (SOA) enablement to its family of products. ITS is a provider of integrated information technology solutions to a wide spectrum of industries and government entities across the Middle East, Africa, and Asia.

For more information, please visit www.its.ws.

Sphere Communications Unveils Unified Communications on Next Generation IP Telephones Powered by Sphericall Web Services Technology

(*San Diego, CA*) – Sphere Communications announced that it will unveil its unified communications solution on next generation IP phones providing communications call controls, unified communications services, video, and internet/intranet information to new micro-browser based IP telephones from Aastra, Polycom, and integrated IP video phones from Tandberg.

Building upon its software-based unified communications services engine, featuring a suite of Web services capabilities, Sphere is expanding its engine to include an integral Web server that can host a range of services for browser-based devices such as IP telephones. Providing an open system that can be easily programmed using standard scripting, XML files and a set of Call Control & Media Services functions the system can be easily customized to adapt to the needs of enterprises and their business applications. Built on open industry standards, the Sphere communications services engine makes it easy for enterprises, ISVs, and developers to optimize their unified communications to their business needs.

For more information please visit www.spherecom.com.

Stampede Technologies Announces Availability of Industry's First End-to-End Acceleration Solution for SOA and AJAX Applications

(*Dayton, OH*) – Stampede® Technologies, Inc., a provider of enterprise acceleration solutions, has announced the general availability of the Stampede Web 2.0 Performance SeriesTM, a solution specifically designed to accelerate the growing array of new enterprise applications implemented using Service Oriented Architecture (SOA), Extensible Markup Language (XML) and Asynchronous Java and XML (AJAX) technologies. This solution platform is the first to combine traditional Application Delivery Controller (ADC) and WAN Optimization Controller (WOC) acceleration techniques with XML Content Based Routing, Schema Validation and Threat Management Services.

The Stampede Web 2.0 Performance Series was named a Finalist in the Best of Interop Performance Category at Interop 2006 Las Vegas, and extends Stampede's unique symmetrical (two-sided) approach to enterprise acceleration that was initially delivered in the Stampede Application Acceleration Series solutionTM. The Stampede solutions are implemented with a hardware appliance at the data center, and client software technology for bi-directional acceleration. This unique approach, with the recently launched Acceleration On-DemandTM technology, sets the standard for what is now being called the Software WAN Optimization Controller, or "Soft WOC".

For more information, please visit www.stampede.com.

SOA Software Selected as "Next Big Thing" at Enterprise 2006

(*Los Angeles*) – SOA Software, the provider of comprehensive SOA and Web services management, security, mediation and run-time governance solutions, has announced it was selected

as one of five companies for the "Next Big Thing" session at the prestigious Enterprise 2006 conference sponsored by the Sand Hill Group.

Held from October 8 through October 10 at Pebble Beach's Inn at Spanish Bay, this conference brings together CEOs and leaders from diverse segments to candidly share ideas, express opinions and debate strategic issues of common interest. The conference includes the top 200 leaders in the enterprise and technology markets, and attendance is coveted. This year, selected companies will deliver a presentation to Elite CIOs.

For more information, see http://www.sandhill.com/conferences/enter2006.php.

IBM Global Services Agrees to Resell LogicLibrary's SOA Governance Solution

(*Pittsburgh, PA*) – LogicLibrary®, a leader in design-time SOA governance and software reuse, has announced it has completed a world-wide reseller agreement with IBM Global Services for their Enterprise Application Modernization and Management practice. IBM Global Services will be able to resell and provide services for LogicLibrary's flagship product, LogidexTM. Additionally, LogicLibrary will provide training and support to IBM's sales and consulting teams as they implement service-oriented architecture (SOA) projects that include Logidex software.

For more information, visit http://www.logiclibrary.com.

JackBe and Cynergy Systems Announce Strategic Partnership Agreement

(*Chevy Chase, MD*) – JackBe, the Rich Enterprise Application (REA) software company, has announced that it has completed a strategic partnership agreement with Cynergy Systems. The partnership is based upon the complementary strengths of the two companies in delivering enterprise-class Ajax and SOA solutions to global 2000 customers through their product architectures, solutions and methodologies. Cynergy has a long history of delivering best-in-class Rich Internet Applications (RIA) solutions to its customers. JackBe's PrestoTM REA Platform enables the delivery of highly interactive applications based on Ajax and SOA that offer enterprise-grade reliability, scalability, and service governance.

For more information, visit www.jackbe.com or www.cynergysystems.com

The SOA Forum Surpasses 1,000 Members to Become Largest SOA Organization for Executives

(*Cambridge, MA*) – The SOA Forum, an exclusive roundtable of senior IT executives and enterprise architects from Fortune 500 companies and Government agencies, has announced that is has reached a significant milestone with over 1,000 members to date.

The SOA Forum is a roundtable of executives who are mandated with the challenging mission of driving a Service Oriented Architecture (SOA) within their organizations. The SOA Forum is celebrating over three years of successful and interactive collaboration between its members. Today the forum includes over 1,000 members representing more than 350 organizations.

The next SOA Forum roundtable will feature Darren Wesemann, Chief Technology Officer for SunGard's Financial Systems businesses and is chief architect of the Common Services Architecture (CSA), who will present "Achieving a Common Services Architecture" on Thursday, October 5 at 2:00 PM EDT. The discussion will cover SunGard's 5-year SOA effort (dubbed the Common Services Architecture, or CSA), which is both a technical architecture as well as a collaborative process including a governance model. Learn how the CSA has transformed the way SunGard works and the SOA lessons they have learn along the way.

For more information, visit http://www.weblayers.com/theSOAforum/.

SOA Software Announces European Expansion

(*London and Los Angeles*) – SOA Software, the provider of comprehensive SOA and Web services management, security, mediation and run-time governance solutions, has announced that it has opened European Headquarters in London, UK to meet the growing demand from Europe for the only complete SOA Infrastructure solution available in the market. As part of the launch, SOA Software plans to expand its European support, sales team and channel partners. Current global alliances include BEA Systems, IBM, Oracle, Microsoft and SAP.

SOA Software enables enterprises to realize the value of SOA and Web services throughout their legacy technologies, and Java or .NET applications, and customer and partner systems. Service-oriented architecture allows greater alignment between IT and business, however, in order to realize true SOA, it is essential to have an underlying infrastructure in place that manages, secures, governs, and mediates Web services environments.

For more information, please visit http://www.soa.com.

StrikeIron Launches Super Data Pack

(*Research Triangle Park, NC*) – StrikeIron Inc., provider of the Web Services Marketplace, has announced the launch of the StrikeIron Super Data Pack. Designed to ease the development of composite applications, Web applications, mashups, and other data-driven applications, the Super Data Pack provides risk-free access to dozens of different data sources all under a single registered account. As part of the package available now, StrikeIron is providing access to the first 10,000 hits per month across all of the various data sources at no cost or commitment from the user.

Leveraging external data sources help an organization reduce the costs associated with fragmented, redundant, and out-dated data, including the maintenance and hardware costs of storing and updating data internally. As an example, the Web services within the Super Data Pack can quickly be integrated into a service oriented architecture (SOA) in order to expand the breadth and depth of data available throughout the enterprise while decreasing the cost of maintaining the same data internally.

For more information, visit www.strikeiron.com.

JackBe Unveils Rich Enterprise Application (REA) Platform to Accelerate Business Value from SOA Investments

(*Chevy Chase, MD*) – JackBe, the Rich Enterprise Application (REA) software company, has announced its Presto REA platform, the industry's solution for delivering enterprise Ajax applications based on SOA and Web services. Applications built on the Presto platform gain business-class reliability, scalability, and service governance. The results are highly interactive browser–based applications that help users to optimize both their regular and ad hoc activities.

Enterprises are making significant investments to create SOA infrastructures that produce SOA and Web services. The focus is now shifting to "putting a face on SOA" by enabling distributed business units to easily consume these services and thus accelerate SOA benefits and ROI. JackBe's Presto will enable enterprises to fully realize their SOA vision through Rich Enterprise Applications deployed at the forefront of business.

The Presto REA platform offers an enterprise-grade architecture based on a complete services governance foundation, unlike consumer-grade solutions or client-side mashup tools that leave governance to the browser. Complementing Presto's governance features are components that provide reliable Ajax messaging, dynamic combination or 'mashup' of disparate SOA services, development and run-time frameworks, and a browser-based studio that empowers users to address their own application.

For more information, visit www.jackbe.com.

BlueNote Networks Unveils SessionSuite SOA Edition

(*Tewksbury, MA*) – BlueNote Networks, the provider of Business Communications Platforms for delivering real-time interactive communications as services in a Service Oriented Architecture (SOA), has announced the availability of two new products in its SessionSuite(TM) product family: SessionSuite SOA Edition and SessionSuite Desktop. SessionSuite SOA Edition is a software platform that delivers voice and video communication services as distributed, re-usable Web Services, allowing enterprises to build and deploy innovative communications-enabled business applications. SessionSuite Desktop is a Windows-based client application that delivers integrated voice capabilities to a desktop or notebook personal computer, and can augment a companion desk phone or mobile phone.

IT organizations are embracing SOAs to deliver agile IT application infrastructures that allow them to easily adapt to changing business requirements. In comparison, enterprise telephony infrastructures have not advanced with the same level of innovation, and remain inflexible, stand-alone, proprietary and hardware-centric. SessionSuite SOA Edition enables organizations to drive similar business benefits from telephony services that they realize today from data applications, by leveraging the distributed software applications architecture of an SOA - where services are built once and shared across many applications. With this approach, enterprises benefit from the ability to quickly and easily leverage re-usable voice and video services for new business opportunities, with reduced development time and costs. Using flexible, standards-based Web Services that can be called upon from any remote software application over any network, BlueNote allows business analysts, application developers and independent software vendors (ISVs) to easily add SIP-controlled voice and video services to any business process, application or Web site.

For more information, visit www.bluenotenetworks.com.

DataDirect Technologies' Shadow RTE Product Enables Advanced SOA Implementations for Mainframes; Mainframe Customers Combine Functionality of Web Services and Real-Time Events to Re-Define Service-Oriented Architectures

(*Bedford, MA*) – DataDirect Technologies, the leader in data connectivity and mainframe integration and an operating company of Progress Software Corporation, has announced the proliferation of advanced service-oriented architecture (SOA) deployments by a number of Shadow(R) RTE customers and mainframe users.

Sometimes referred to as SOA 2.0, advanced SOA entails the merging of SOA and Web services interoperability with other architectural types, most notably event-driven architectures (EDA). Moving beyond Web services to include event notifications and enterprise-class computing environments, users experience a more agile and responsive IT infrastructure capable of achieving what Gartner, Inc. defines as the Real-Time Enterprise - an enterprise that allows businesses to more effectively compete by using up-to-date information to progressively remove delays to the management and execution of its critical business processes.

For more information, visit www.datadirect.com.

Camilion Solutions Announces ProductAuthority V5.1; Leverages SOA and Standards to Enable Easy Integration of Product Data and Rules into Back-end Systems for Agility and Speed to Market

(*Toronto*) – Camilion Solutions, the provider of product management, underwriting and online sales solutions for the insurance industry, has announced the release of ProductAuthority(TM) V5.1, the company's flagship enterprise product management solution.

ProductAuthority V5.1 now incorporates the foundational standards - WSDL, SOAP and UDDI - to make the solution fully Web Services compliant, and incorporates a key ACORD XML Life and Annuity Specification to improve efficiency and manage risk in the annuity sales process.

ProductAuthority V5.1 enables insurers using an open Service-Oriented Architecture (SOA) to more easily and cost effectively integrate product data and rules into their operational systems, including policy administration systems. Product data and rules are stored in the central repository of ProductAuthority's "pure" product configurator where they can be exposed as Web Services to all back-end systems that need to be 'product-aware'. This not only eliminates the hard-coding of product data and rules into multiple systems, but also results in systems with unprecedented agility, allowing insurers to implement or change products in days.

This new version also includes a new testing facility to allow insurers to thoroughly test products before they are launched. Insurers can now test all product rules; define, store and manage test suites; and share or re-use test suites across products. The result is substantial improvements in product quality and reduced development and implementation time.

With the incorporation of ACORD's XML Life and Annuity Specification Policy Product Transmittal No. 1201, ProductAuthority also now automatically generates a standard definition of a Product Profile for Annuities (PPfA), which includes the basic set of information and business rules necessary to submit a properly qualified and complete application for variable and fixed annuity products. These PPfAs are then exposed to other product-aware systems as Web Services. The annuities sales process is now more efficient and has less risk because product errors are significantly reduced and the product information shared between an insurer, its selling partners and the end customer is consistent.

For more information, visit www.camilion.com.

Mindreef Partners With HR-XML Consortium to Accelerate Web Services Industry Standards for Interoperability

(*Hollis, NH*) – Mindreef(R), Inc., which helps organizations drive quality, trust, and reuse into their service-oriented architecture (SOA) deployments, has announced that it is partnering with the HR-XML Consortium to accelerate Web services industry standards for interoperability. The HR-XML Consortium is a global, non-profit consortium dedicated to the development and promotion of a standard suite of XML specifications to enable e-business and the automation of human resources (HR) related data exchanges.

The current phase of Mindreef's work with the consortium involves the creation and availability of an "HR-XML Profile" as a downloadable element for use with Mindreef SOAP-

scope Server (TM), a team-based platform with tools for governance, testing, diagnostics and support that span the entire service-oriented lifecycle. Teams at Wachovia, Fidelity National Financial, Charles Schwab, IBM and more than 3,000 enterprises and organizations worldwide use Mindreef products to design, implement and deploy projects faster and achieve pervasive SOA quality. The resulting quality and trust leads to increased adoption and reuse of services throughout the SOA.

For more information, visit http://www.hr-xml.org or http://www.mindreef.com.

Volvo Cars Selects XPower and Progress Software to Drive Superior Customer Service

(*Bedford, MA*) – Progress Software Corporation, a provider of application infrastructure software to develop, deploy, integrate and manage business applications, today announced that Volvo Cars Belgium has deployed the XDMS Dealer Management System (DMS) from XPower Automotive Software, a Progress Application Partner (AP). This service-oriented architecture (SOA) enabled system, based on Progress Software's Progress(R) OpenEdge(R) business application platform and Progress Sonic ESB(R) (enterprise service bus), was deployed across all 65 dealers in Belgium in a record-breaking eight months. The Progress-based system enables dealers to offer an enhanced purchase, support and service experience for their customers and ensures that 98 percent of Volvo parts are delivered overnight.

The XPower XDMS is uniquely designed for the automotive industry and uses vehicle identification numbers (VINs) to track each car through its entire lifecycle. This includes the original vehicle order, maintenance history, warranty repairs, service appointments, technician time, parts and customer invoices. The system also handles back office accounting functions for the dealer, including monthly reporting for the vehicle manufacturer.

For more information, please visit www.xpower.be or www.progress.com/index.ssp.

Blue Cross and Blue Shield of Nebraska Selects LogicLibrary as Foundation for Major SOA Initiative; Health Insurance Provider Adopts Best-of-Breed SOA Governance Solution to Manage Development and Reuse of Software Assets

(*Pittsburgh, PA*) – LogicLibrary(R), a leader in design-time service-oriented architecture (SOA) governance and software reuse, has announced that Blue Cross and Blue Shield of Nebraska (BCBSNE) has selected Logidex(TM) to manage services within the organization's new SOA infrastructure. The SOA is an integral part of the organization's IT modernization project and is intended to produce high-quality business services for BCBSNE customers.

An independent licensee of the Blue Cross and Blue Shield Association, Blue Cross and Blue Shield of Nebraska insures or provides benefit administration to more than 560,000 people. In order to support the agility required in today's marketplace, BCBSNE is moving to an SOA. LogicLibrary's Logidex will serve as the SOA Repository that makes it possible for BCBSNE to access, secure, reuse and govern software development assets (SDAs) in its SOA and other application development initiatives.

For more information, visit http://www.logiclibrary.com.

SOA Software Products Drive More Than 10 Billion Web Service Transactions; Company's Products Manage and Secure over 425 Million Transactions Per Month in Production at the World's Largest Companies

(*Los Angeles*) – SOA Software, a provider of comprehensive SOA and Web services management, security, mediation and run-time governance solutions, announced that its products have processed more than 10 billion Web services transactions at large enterprise customers. With a fast growing run-rate of over 425 million transactions per month, SOA Software confirms its status as the most proven SOA infrastructure vendor.

There can no longer be any doubt that Web services and SOA are a fact of life in large enterprises. More and more large companies are using Web services extensively and are relying on an SOA strategy to reduce costs and better align IT with the business. SOA Software's customers, include leading companies such as Merrill Lynch, Verizon, Citigroup, Pfizer and JetBlue, are using its products to provide security and management capabilities for mission-critical applications.

For more information, please visit http://www.soa.com

MAKE YOUR SOA SECURITY MOVES WISELY...

XWALL
WEB SERVICES FIREWALL

XRAY
WEB SERVICES DIAGNOSTICS

VULCON
VULNERABILITY CONTAINMENT SERVICE

SENTRY
SOA SECURITY GATEWAY

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

FORUM SYSTEMS ENTERPRISE SOA SECURITY SOLUTIONS:

- TRUSTED SOA MIDDLEWARE
- WEB SERVICES SECURITY
- XML ACCELERATION

WWW.FORUMSYSTEMS.COM

FORUMSYSTEMS

THE LEADER IN WEB SERVICES & SOA SECURITY

Complex and evolving systems are hard to test...

Parasoft helps you code smarter and test faster.

Start improving quality and accelerating delivery with these products:

Awarded **"Best SOA Testing Tool"** by Sys-Con Media Readers **SOAtest™**	**InfoWorld's 2006 Technology of the Year pick** for automated Java unit testing. **Jtest™**	Automated unit testing and code analysis for C/C++ quality. **C++test™**	Memory errors? Corruptions? Leaks? Buffer overflows? Turn to… **Insure++™**	Easier Microsoft .NET testing by auto-generating test cases, harnesses & stubs **.TEST™**	Automate Web testing and analysis. **WebKing™**

PARASOFT®
We make software work.™

Go to www.parasoft.com/WSJmagazine • Or call (888) 305-0041, x3501