SYS-CON

NOW INCORPORATING XML JOURNAL

WebServices JOURNAL

.NET J2EE XML

February 2005 Volume 5 Issue 2

pg.36
IDENTITY-BASED
SERVICE-ORIENTED ARCHITECTURE
Regulating business needs with Web services

Best Practices on Integrating Data Models for SOA

Integrating underlying data models is an essential precursor to SOA

pg.10

Service-Oriented Architecture Best Practices

Meeting your goals takes a mixed effort

pg.14

ESB Myth Busters

Clarity of definition for a growing phenomenon

pg.22

Distributed Parallel Computing with Web Services

A pivotal role on the back end

pg.28

RETAILERS PLEASE DISPLAY
UNTIL MARCH 31, 2005

$6.99US $7.99CAN

0 71486 03420 9 02>

web services EDGE conference & expo

FEBRUARY 15-17, 2005
BOSTON, MA

INSIDE
2005 CONFERENCE GUIDE
PAGE 41

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

FEBRUARY
BUZZ

SOAPtest®

An Ounce of Prevention

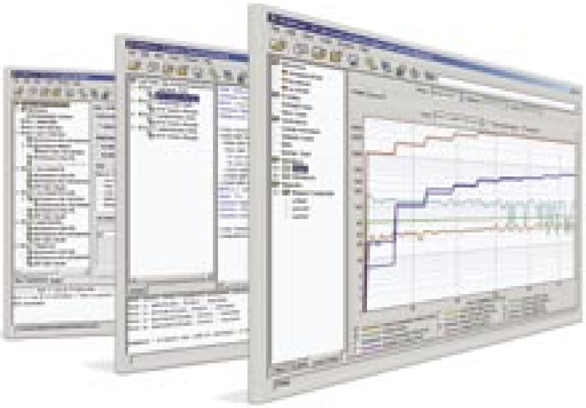
To maximize the benefits of coding standard analysis, it's critical that you recognize coding standards as a means of preventing errors — not detecting them. Many developers are disappointed if a coding standard violation doesn't point them to an obvious bug. When they explore a violation and find an error-prone construct rather than an error, they think that the coding standards aren't useful, eventually stop investigating violations, and later stop performing coding standard analysis altogether. This speaks to a fundamental problem with the software industry: Most development and testing teams are concerned with *removing* errors, but not with *preventing* them.

Error prevention involves correlating each error to the exact point in the development process that allowed it, then fixing that part of the process. This prevents the need to debug applications after the fact, and produces an exponential increase in product quality. Error prevention is very different than error detection, which is the process of finding and fixing errors *after* an application is built. When development focuses only on error detection, the flawed process that generated those errors is left uncorrected and the errors continue to occur. Furthermore, this misplaced focus has a negative impact on software quality and team productivity.

The coding standards promoted by industry experts were produced in response to other developers' mistakes and development process flaws. The experts performed the most difficult task: figuring out why errors were occurring and how to prevent them. To prevent errors, you just follow the coding standards that the experts designed and tested. Essentially, you get to benefit from the lessons learned from other developers' mistakes without having to suffer the consequences of personally committing those mistakes.

*– **Adam Kolawa**, Ph.D.*
Chairman/CEO of Parasoft

It's automated. It's fast. And it's the most versatile Web Services testing tool.



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

The simple fact is, no other Web service testing product can do what SOAPtest can do. From verifying functionality and performance to ensuring security and interoperability, SOAPtest automates all critical testing processes across the entire lifecycle of your Web service.

But don't take our word for it.... Go to www.parasoft.com/SOAPtest_WSJ and try it for free—no commitment, no obligation. If you like it (and we suspect you will) just let us know. We'll be more than happy to help you and your development team get up and running.

For Downloads go to www.parasoft.com/SOAPtest_WSJ
Email: SOAPtest_WSJ@parasoft.com Call: 888-305-0041 x1209

PARASOFT®
We make software work.℠

Features

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

Benefits

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

Platforms

Windows 2000/XP
Linux
Solaris

Protocol Support

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

Contact Info:

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

www.parasoft.com

eclipse CON™ 2005

2nd Annual Gathering of the Eclipse Community

Plug-in to Eclipse

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

Tim O'Reilly

Urs Hoelzle

John Wiegand

Erich Gamma

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

Urs Hoelzle,
VP of Engineering, Google

John Wiegand and Erich Gamma,
Eclipse Platform PMC Leaders

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

February 28-March 3, 2005
Burlingame, CA

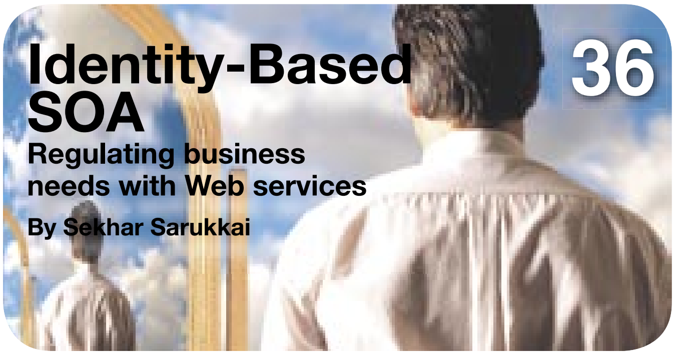
Register at www.eclipsecon.org

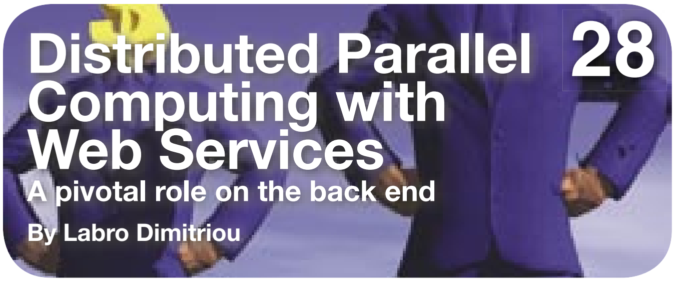
WebServices Journal
.NET J2EE XML

FEBRUARY 2005 Volume 5 Issue 2

Inside WSJ

XML JOURNAL





From the Editor

Industry Commentary

WSJ:Best Practices

WSJ: Web Services @ Work

WSJ: Product Review

Enterprise Service Bus

FEBRUARY

BUZZ

Jtest®

An Ounce of Prevention

To maximize the benefits of coding standard analysis, it's critical that you recognize coding standards as a means of preventing errors—not detecting them. Many developers are disappointed if a coding standard violation doesn't point them to an obvious bug. When they explore a violation and find an error-prone construct rather than an error, they think that the coding standards aren't useful, eventually stop investigating violations, and later stop performing coding standard analysis altogether. This speaks to a fundamental problem with the software industry: Most development and testing teams are concerned with *removing* errors, but not with *preventing* them.

Error prevention involves correlating each error to the exact point in the development process that allowed it, then fixing that part of the process. This prevents the need to debug applications after the fact, and produces an exponential increase in product quality. Error prevention is very different than error detection, which is the process of finding and fixing errors *after* an application is built. When development focuses only on error detection, the flawed process that generated those errors is left uncorrected and the errors continue to occur. Furthermore, this misplaced focus has a negative impact on software quality and team productivity.

The coding standards promoted by industry experts were produced in response to other developers' mistakes and development process flaws. The experts performed the most difficult task: figuring out why errors were occurring and how to prevent them. To prevent errors, you just follow the coding standards that the experts designed and tested. Essentially, you get to benefit from the lessons learned from other developers' mistakes without having to suffer the consequences of personally committing those mistakes.

*– **Adam Kolawa,** Ph.D.*
Chairman/CEO of Parasoft

Deliver better Java code in less time, with fewer resources.

Parasoft® Jtest® makes Java unit testing and coding standards analysis fast and painless.

With just a click, you can verify that your Java code is robust, high quality and secure. Jtest automatically verifies compliance to hundreds of coding rules while automatically generating and executing JUnit test cases — creating harnesses, stubs and inputs.

See how Jtest can enhance your Java development efforts...
Visit our Parasoft Jtest Resource Center at www.parasoft.com/jtest_WSJ for the latest information — including white papers, webinars, presentations, and more.

For Downloads go to www.parasoft.com/jtest_WSJ

Email: jtest_WSJ@parasoft.com Call: 888-305-0041 x2174

PARASOFT®
We make software work.™

Features

- Full Eclipse integration
- Quick Fix automatically corrects errors
- Automatically generates JUnit test cases
- Customizable testing and reporting

Benefits

- Makes error prevention feasible and painless, which brings tremendous quality improvements, cost savings, and productivity increases
- Makes unit testing and coding standard compliance feasible and painless
- Encourages the team to collaborate on error prevention

Platforms

- Windows 2000/XP
- Linux
- Solaris

Contact Info

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

WebServices JOURNAL
.NET J2EE XML

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

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

SOA What?

WRITTEN BY
SEAN RHODY

One of the fun parts of being a software architect is trying to figure out how to build whatever it is that you are supposed to build. It's even more fun when you look at the architecture for an entire enterprise, and have to make choices that integrate every complexity and account for every nuance of the portfolio, even if only long enough to get something in place before ripping something else out.

The advantage of buying a COTS product from a software vendor is that you get expertise at programming, and in a particular line of business, without having to hire, retain, and pay a staff of programmers. This ability to buy functionality was a major innovation in the entire software development process, and a boon to departments that no longer had to wait out a long, waterfall-based life cycle before they got applications to assist them in doing their jobs more effectively.

The downside, as the IT department found to its horror, was that the ISVs weren't interested in building to every platform under the sun. Instead, they'd choose a system or a platform, like the mainframe, AS/400, VAX, or even client/server and create software. While the software was good at the business task (or at least good enough) that the business users were happy, it was a nightmare for IT. Now they had one of everything to support, and instead of knowledgeable programming staff who knew the platform as well as the application, they had to quickly train whoever was handy. It's not a wonder that IT satisfaction plummeted over the years. Very rarely did the true cost of packaged software, in terms of support, and impact to other parts of the business, ever get addressed.

Fortunately we have Web services now. Many of the problems caused by silos of applications can be mitigated by applying the technologies developed for Web services. Platform differences can be overcome. Communication mechanisms can be established. Locations of services can be determined.

And yet, it's still possible to make programmatic spaghetti with Web services and to design services that don't scale, aren't secure, and can't be managed. That's because Web services provides technology, but not architecture.

And that's why service-oriented architecture (SOA) is so important. An SOA helps overcome the challenges of application integration using Web services, as well as other concepts, constructs, and tools that aren't necessarily part of the core Web services stack.

SOA is not really a product, or a technology. Although you can buy an SOA in the same way you can buy a development methodology, in most cases you aren't buying code but rather thoughts. Like the instructions to a complex model airplane, SOA will guide the construction and ensure that there are no pieces left over at the end.

Applying SOA in an existing environment can be a challenge. Services are a different mindset than applications – in fact, applications are built on top of services that may be reused in an SOA. The concepts of user interface and integration are different, and with existing legacy software, it may take years of careful, planned refactoring before the software that was is the service that should be.

This month's focus is on SOA and the enterprise service bus (ESB). The ESB is a similar concept, but slightly simpler – it provides a messaging backbone for enterprise communication. And for many organizations, adopting an ESB before an SOA is a wise move – sort of a crawl before you walk approach. Regardless of whether you do one, the other or both, Web services technology still underlies the concepts. WSDL, XML, SOAP, and other message bindings are core to both concepts. And the concepts themselves are key to avoiding building yet another stovepipe.

About the Author

Sean Rhody is the editor-in-chief of ***Web Services Journal***. He is a respected industry expert and a consultant with a leading consulting services company.

■ ■ ■ sean@sys-con.com

WebServices Journal
.NET J2EE XML

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

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

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

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

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

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

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

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

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

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

SYS-CON MEDIA

The Missing Link Between Business and Web Services?

WRITTEN BY
JOHN O'SHEA

The enterprise service bus (ESB) is, arguably, emerging as the preeminent platform for building, deploying, and managing Web services. However, once you have created and deployed your Web services, what's the next step? For many developers, it is the orchestration of those services into composite applications and business processes. In the past, this area has proved troublesome. The majority of existing business process tools are vendor specific and proprietary, using adapters and bridges to integrate with open platforms like Web services. However, the emergence of Business Process Execution Language (BPEL), coupled with the growth of Web services and the ESB, is providing a real, native, service-oriented architecture (SOA)–based alternative.

This potential was brought home to me when I was working with a high-tech manufacturer who faced the perennial problem of managing old, proprietary software that linked their ERP systems with their partners (primarily contract manufacturers). The existing system was complex, required a lot of maintenance, and was difficult to change. The company identified Web services and SOA as an alternative solution that would enable them to rip out the legacy software while, at the same time, reduce the cost and complexity of linking with their partners.

By adopting a Web services approach, they immediately got cross-platform interoperability, standards-based development, location independence, integration with existing systems, and promotion of future service reuse. They used an ESB to build and deploy the Web services, which provided the necessary policy, transformation, and routing capabilities to enable their services to be exposed. The documents exchanged between the company and its partners were defined in XML; partner interfaces were quickly defined in WSDL.

However, Web services alone only partially solved the integration problem. They still needed a mechanism to define and coordinate service interactions to deliver composite applications and automate real-world business processes. These processes varied widely, from automating the selection of partners for delivering a product, to transferring documents with the selected third-party contract manufacturer, and ultimately to delegating the construction and delivery of products. The processes also controlled the exchange of shipping details, component warranty details, and, of course, invoicing.

Given their choice of Web services, the most obvious option for building the business processes was BPEL. Once these processes were described using BPEL, they were deployed as a service on the ESB. The combination of BPEL and the ESB offered a single solution that enabled them to expose their enterprise infrastructure as Web services, to apply policy to the Web services, and to compose the services into powerful and visible business processes.

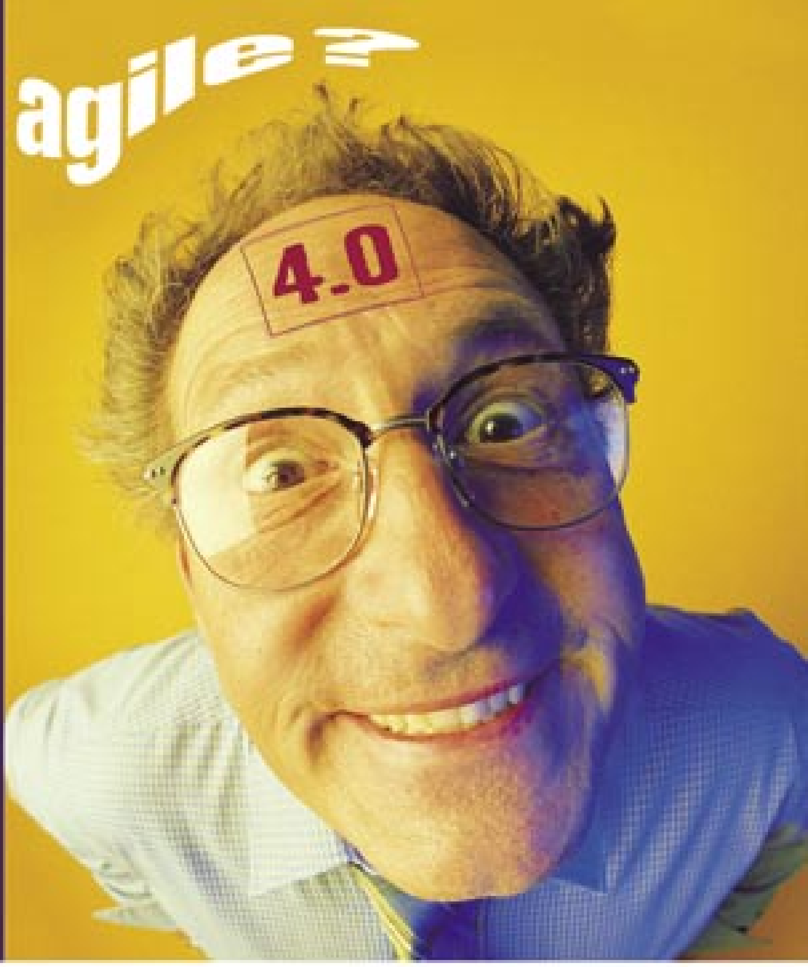
The customer was pleasantly surprised that BPEL met their requirements. The BPEL server they used was designed to run many processes. Each process typically ran for months after being inactive for considerable periods of time. The BPEL server ensured that critical process state was safely stored in a transactional datastore, should recovery be required in the event of transient service unavailability. The BPEL server also provided their administrators with a Web-based management console that enabled them to query and view the activity history of process instances and remotely control the execution of running or faulty processes.

BPEL enabled them to use off-the-shelf tools to define the process flow in a visual manner, as a series of interactions with SOA service interfaces hosted on the ESB. The BPEL process itself is hosted on the ESB and externally exposed as an SOA service. A stand-alone client driver application triggers the start of each BPEL process instance when an order is created or modified. From then on, all interaction between services takes the form of SOA-style invocations.

As the customer implemented the project, they learned a number of important lessons about BPEL:

▸ Continued on page 15

Is your SOA really agile?



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

Developers • Testers • Operations • Support

Mindreef SOAPscope 4.0

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

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

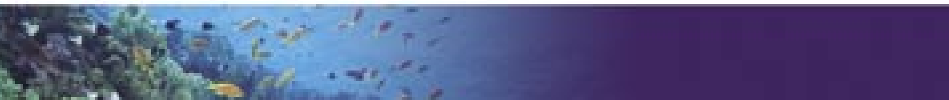
Share - Exported workspaces support sharing among team members preserving complete problem data including messages, WSDLs, and notes. Sharing workspaces is a great way to support web service users.

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

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

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

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



Mindreef, Inc.
www.Mindreef.com

Best Practices in Integrating Data Models for SOA

Integrating underlying data models is an essential precursor to SOA

■ This article describes how an essential precursor to any SOA implementation is a data modeling exercise that integrates all underlying data models, focusing more on the business requirements than on system- and application-specific requirements.

WRITTEN BY
JIM GABRIEL

Integrating data models in a complex enterprise can be difficult because the IT landscape often reveals massive duplication and redundancy. Refining this situation without semantic loss is a tough nut to crack. This article discusses the problem domain and makes some recommendations.

Gartner, Inc., advises organizations wishing to fully exploit service-oriented business applications to focus on integrating the processes and underlying data models, rather than on integrating individual application components. Failure to integrate these aspects will place the organization at a competitive disadvantage. Gartner believes that such metadata management is "essential to reducing the escalating complexity of management and maintenance of integrated software platforms."

This is very sound advice. When exposing an application through Web services, it is very tempting to write services that speak directly and specifically to the application, as this is the simplest, most cost-effective path in development (at least in the short term). Unfortunately, this gives us "tight coupling" – that is, we are exposing the interface to the application component and little more. Web services that access an application in this way provide the equivalent of RPC-like, point-to-point integration. "Tightly coupled" is not one of the many definitions applicable to SOA that we should by now have learned by rote. Rather, "loosely coupled" and "dynamic" spring to mind as far more applicable.

The benchmark for testing whether services are sufficiently loosely coupled and dynamic is the level to which services are application specific. If developers write services that can be application agnostic, much of the battle has been won. In other words, you should be able to unplug the underlying application component and plug in an equivalent from another manufacturer, and the service should not need updating.

The technical infrastructure that enables application agnosticism depends on a layered approach to SOA. That is, application-agnostic services are only possible if the SOA implements a data model that properly represents and integrates the underlying data models in the enterprise at a layer of abstraction higher than the interface layer. This is very important in the interests of long-term maintainability and evolution. The integrated data model is the source of all data definitions and interface definitions required by services, but it is also the basis for resolving model-to-model mappings in the interface layer. The relationship between services and underlying applications is illustrated in Figure 1.

The relationship between services and application components passes through a number of other layers. First, the payloads of message-centric services are described by schemas. These schemas are assembled from the integrated data model. The relationship between the payload schemas and the underlying data models is managed through transformations. The transformations are built against the integrated data model, which must therefore have knowledge both of the underlying data models and of the payload schemas that have been assembled from the model.

The interface layer is the transformation layer where data is transformed from its format, as described by the integrated data model, into the format required by the application model, and vice versa. For this reason, the integrated data model is sometimes referred to as the "interface model" or "transformation model."

This metadata topography offers up a number of interesting issues:

- ***Multiple layers of description***. Application landscapes can be validly described by the sum total of pieces in each layer. For example, the sum total of the schemas in the schema layer arguably provides an equivalent description of the application landscape to the sum total of the underlying data models.
- ***References to any given object can occur in multiple layers and contexts.*** From a programming perspective, these layers of metadata and application objects suggest a high level of duplication and potential redundancy. Maintenance and evolution considerations are of paramount importance in planning the implementation of your SOA.
- ***The integrated data model must include knowledge of the underlying data models that it integrates.*** An integrated data model is partly a collection of existing data models, and partly a new schematic representation of the data and processes in an enterprise.

In assembling the integrated data model, the following questions need to be answered:

- Is an integrated data model a real, tangible model, or is it a logical concept only?
- What modeling or schema language should the integrated data model use?
- How do you create the integrated data model?
- How do you resolve duplication and redundancy?
- Is an integrated data model a passive reflection or an active master?
- Where do you store the integrated data model?
- How do you manage the life cycle of the integrated data model?

Real or Logical

The integrated data model is real. You cannot build the various layers described in Figure 1 unless you collate in one place the metadata that describes all the data describing the SOA. For example, transformations in the interface layer are schema-to-schema mappings that can only be defined if the source and target schemas actually exist. Where you store it and how you manage it are very important questions that you will need to answer before implementing the SOA. Your decisions at this stage will have long-term consequences on the maintainability of your implementation.

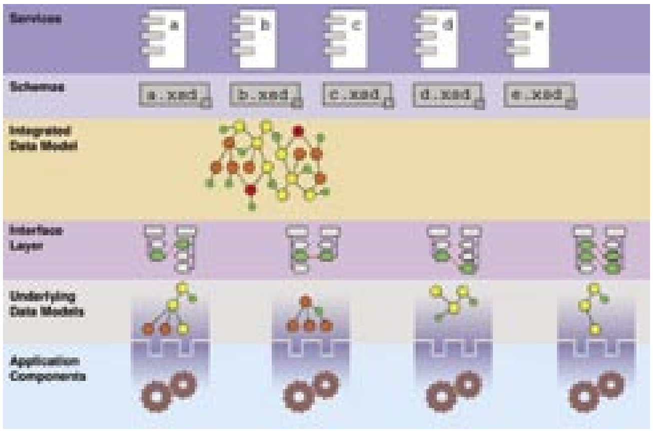


FIGURE 1 Layers of models and interfaces between application components and services

Flavor

The integrated data model is an XML data model that is best described in XML Schema; after all, service payloads are constrained by XML Schema. Note that the expression of the data model – that is, how you choose to deploy it as something tangible and usable (in this case a family of xsd files) – is not necessarily the same as the development image of the model. For example, some organizations capture everything in UML and export the resulting model as XML Schemas. However, it is good advice to stay as close to the final implementation as possible.

Creating the Integrated Model

To achieve an integrated data model for an organization, a single, homogeneous model must be created that fully represents all the relevant underlying data models, including any schemas used for trading partners or industry standards. This is a serious data modeling exercise that typically requires the input of highly experienced analysts and architects. The end result is a set of custom standards for the enterprise.

The starting point is often a metadata-gathering phase, where all the existing models are imported into a central place (usually a repository). Exposing metadata is not always straightforward, as some applications do not publish interfaces or schemas and others require some interpretation or embellishment along the way. For example, database schemas in an Oracle database can be exported as DTDs or XML Schema files using the XSU utility (XML-SQL Utility), but ensuring that relationships and constraints are properly expressed in the resulting schemas requires a manual pass over the metadata.

Be aware that importing metadata from existing systems creates an application-specific view of the landscape, warts and all. There is little point in creating application-specific integration models, as this does not abstract us high enough above the underlying application specifics. This exercise must therefore be accompanied by a broader analysis that models the actual data requirements of the enterprise, preferably with a very forward-looking appraisal of the expectations facing the business. You should look at this as a bonus. As Thomas Erl says in his article "Best Practices for Transition Planning" in the November 2004 issue (***WSJ*** Vol. 4, issue 11), in "planning a migration to a standardized adoption of SOA you…have an opportunity to erase some of the neglect from the past." This is a speculative analysis action that certainly applies to the data-modeling phase.

The data modeling phase also provides an excellent opportunity not only to erase some of the neglect from the past, but also to introduce other essential best practices into your SOA, such as a service-oriented security model.

At this metadata gathering stage, some organizations use metadata management tools, equipped with the drivers and connectors, to facilitate the import of metadata from various systems in the application landscape. However, before you rush out and buy a metadata repository, be sure you understand exactly what your long-term requirements of such a system might be, paying particularly good attention to questions of maintenance and evolution.

Duplication and Redundancy

When data models are integrated, some translation and rationalization is inevitable because duplication and redundancies must be resolved. It is essential that no object properties are lost or diminished during the rationalization phase, as this would result in a reduction of the metadata and therefore of the potential functionality. Rather, the resulting integrated data model should be more descriptive and functional than a simple sum of the component models.

When rationalizing data models, we often have to interpret and manage the fact that two objects in separate application models are essentially the same object. How we deal with this depends on the requirements. If two objects exist because different teams of developers have made up their own different names for the object – and this is a very common problem in XML-driven systems – you can either straighten out the problem in the underlying models before integrating them or create a new "alias" object in your integrated model that can map to each variant.

Another common problem is the use of the same name in underlying models for what are essentially different objects. The solution here is similar to the previous problem: either solve it before integrating, or integrate by creating new objects at a higher level. Note that when working with XML Schemas, namespaces can determine how you proceed. Clearly, if two teams have used the same name in the same namespace for different purposes, and the act of integrating the data models exposes that problem for the first time, there is no way you can allow both objects to continue coexisting with different meanings.

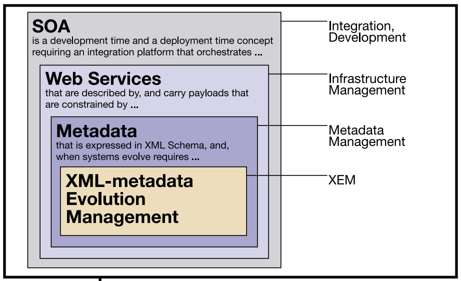

SOA
is a development time and a deployment time concept requiring an integration platform that orchestrates ...
Web Services
that are described by, and carry payloads that are constrained by ...
Metadata
that is expressed in XML Schema, and, when systems evolve requires ...
XML-metadata Evolution Management
Integration, Development
Infrastructure Management
Metadata Management
XEM

FIGURE 2 **Metadata evolution management**

When two objects that are essentially the same object need to coexist for transformation purposes, the integrated model must describe both objects rather than attempt to resolve them into one, otherwise it is not possible to create the transformation. For example, in a trading partner scenario where a legacy system from a supplier processes a credit card payment according to constraints described in a DTD (and your system speaks an XML Schema equivalent), the correct way to integrate your data models would be to load both the DTD and the XML Schema equivalent, and then create the mappings between the constituent objects.

Passive or Active?

Active metadata is business driven; passive metadata is technology driven. Once we have an integrated data model, the metadata takes on an active role. Prior to this stage, the role of metadata was passive because it served no further purpose other than to describe and constrain data. It reflected existing systems and application components. Active metadata drives new development effort from within the metadata. In other words, when we need to modify the payload of a service to satisfy a changing business requirement, our starting point must be the externalized schema that describes the payload. This is a very serious consideration for tooling and evolution management. From this point onward, any changes to the way the business functions will force developers to go to the metadata first – that is, the integrated data model – and make their changes there before changing code.

Storing the Model

Bearing in mind the active nature of the metadata in your SOA, it is essential that you store the model in an environment that supports the concept of change. Repositories that provide container functionality and business analysis support are essential parts of the IT landscape before you implement your SOA. Once you start implementing the SOA, the integrated data model must be managed in a model-driven environment with full developer support. This allows changes to be made centrally and deployed out to the environment in the form of version-controlled schemas, transformations, and so on.

Managing the Life Cycle of the Model

The term that is applicable to managing the life cycle of a data model is "metadata evolution management." This is at the heart of any SOA, because SOA is a development-time and deployment-time concept that requires an integration platform that orchestrates Web services. Web services are described by, and carry payloads that are constrained by, metadata that is expressed in XML Schema. When systems evolve, the metadata must also evolve, thus making XML-metadata evolution management an essential part of the infrastructure (see Figure 2).

This was the subject of my article "Metadata Evolution Management in your SOA" in the last issue (***WSJ*** Vol. 5, issue 1), so I would refer you to that discussion for more detail and some recommendations for managing the evolution of metadata in the SOA. Suffice it to say that managing the life cycle of your Web services development, particularly from the perspective of the evolution of metadata, is not simply a schema-versioning problem. Versioning schemas is about technical constructs and development processes, not about the management of metadata evolution. Metadata evolution management is the real problem facing the long-term life-cycle management of Web services development projects.

Summary

The data modeling exercise is such an important precursor to any SOA implementation that I would describe it as more than a best practice: it is essential. This data modeling exercise must integrate all underlying data models, focusing on the business requirements and abstracting above system- and application-specific requirements where possible. Integrating data models will reveal duplication and redundancy, which must be powerfully resolved by suitably experienced architects and analysts. The end result will be a metadata model that will assume an active role in your IT landscape. Preserving the integrity of the metadata model as systems and business requirements evolve will constitute the next major challenge for the IT department, for which a strategy and technical solution must be prepared.

References

- Gartner, Inc. (2003). "Predicts 2004: Application Integration and Middleware." ⓔ

About the Author

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

■ ■ ■ jgabriel@digitalml.com

Service-Oriented Architecture Best Practices

Meeting your goals takes a mixed effort

The concept of a service-oriented architecture is a powerful tool for simplifying enterprise integration. Following the three principles of modularity, encapsulation and loose coupling will achieve some amount of improvement for an individual service. It is insufficient to have a loose set of principles to guide enterprise architecture designs. It is also important to have "best practices" on how to build, run and manage enterprise systems. This document will cover the best practices that have been culled from Epiphany, a number of Epiphany's top customers and leading analyst firms.

WRITTEN BY
BILL ROTH

Define the language of your enterprise. First and foremost, there must be a common language that your, enterprise must speak. This amounts to being able to define important entities as XML Schemas and transformations as XML style sheets. Enforcement is crucial, as well ensuring interoperability across an enterprise wide system. This is the single most important action an enterprise can take to smooth its transition to an SOA.

In one sense, this principle should not be necessary if industry standards groups could agree on basic XML document formats for simple things like names and addresses. Since this is unlikely to happen anytime soon, it is important that your organization set its own language policies now, and deal with transformation issues via XSLT later.

Establish a naming convention for your enterprise. In a SOA environment, having a document that formulates how to name Web services, service interfaces, legacy system endpoints, or any public component will help the enterprise architects, administrators, and developers provide consistent services.

Define service interfaces first, implement later. In a SOA, the interface for a Web service is more important than its actual implementation. When designing composite applications (i.e. applications composed from a cluster of web services), *recognize that heterogeneity is the norm*. One analyst study put the average number of hardware/software platforms in the Fortune 500 at 6. This underscores setting a baseline "language" for your organization, as heterogeneity is an unavoidable fact in any large enterprise.

In addition to having best practices around people, practices, and technology, it is also important to have "design patterns" that can be reused in order to make the design and construction of an SOA faster and easier. Just as there are design patterns for object-oriented programming, there is a group of experts who are looking at describing common enterprise workflows as a set of patterns. Led by BEA and the Middleware Company, initial work has been done to catalog a number of enterprise workflows captured as patterns for an SOA. While their efforts are in the early stage, the work appears promising.

Establish a service categorization. Services come in a number of flavors. An organization should decide on its services taxonomy. This definition will help to define development roles, as well as help to suggest a level or organizational structure. At a high level, there are only a few types of services. The Middleware Company group is attempting to pull together a set of best practices for building out a service-oriented architecture. The expert group has defined a taxonomy of services, which is a useful starting place:

- ***Component services:*** Simple atomic services potentially acting on single enterprise resource (e.g., database, code, etc).
- ***Data services:*** Service providing data querying, combination and transformation for multiple data sources.
- ***Business services:*** Atomic services composed of combinations of component services and rules.
- ***Workflow services:*** Long lived business processes coordinating other services with external interactions.

Separate rules. It is important to categorize business logic/rules, further, into "process" type business rules versus "UI" type business rules. Process-type business rules are good candidates to encompass within business services and UI-level business rules should be separated out.

Don't skimp on training. Start with a common skill set. Most organizations are new to the technologies behind SOA (like WS-I, WS-BPEL, etc.), so it is important for an IT staff to acquire the skills to understand and implement SOA on an enterprise-wide scale.

Architect a common management layer. A service-oriented architecture provides both additional opportunity and complexity for managing an operational system. Many of the infrastructure vendors like IBM and BEA provide the ability to monitor and manage Web services. It is important to implement a uniform management infrastructure for managing hardware, operating systems, applications, and Web services for maximum visibility.

Analyst firms have begun to discuss implementation issues around SOA as well. In a recent report, Forrester offers several suggestions for SOA best practices. The Forrester report offers several points for enterprises to consider:

Align services with business processes. This is important for a number of reasons. First, the service must be understood by the business users for it to be useful and successful. Second, the service should match the business process it is managing to mitigate the need to change management. Third, change management will be easier. As business processes change, determining which services need to change will be made easier.

Design decisions need to be made regarding integrating business rules within services. Consumer application-specific business rules may not have to be integrated within the producer services if there is no value in having these rules apply to other consumer applications. This also facilitates easy maintenance of services because there is less coordination with multiple consumer applications

Start with services, do Web services later. Moving to an SOA does not necessarily require Web services. Many companies have been successful building out an SOA message-oriented middleware (MOM). This includes JMS-based software as well as IBM WebSphere MQ (the former MQSeries). Applications rooted in MOM-based systems generally met the requirements of SOA: encapsulation, loose-coupling and modularity, but with the notion of messages on a queue rather than SOAP/XML messages.

Wrap packaged applications in service interfaces. This was mentioned above, but is important enough to restate. Enterprises should be wary of vendors claiming to provide a platform. Leveraging a single vendor's platform will unnecessarily tie an organization to one vendor. This is a risky strategy. It is better for enterprise information assets to speak a neutral, common language that will ensure interoperability.

Have a version resolution architecture. Web services present the ability to reuse software like never before. A Web service can be a public resource used by hundreds of applications. However, if the Web service changes the applications will have to change. This will create a maintenance nightmare. It is important to have an architecture where interfaces have a version associated with them, and a way of resolving a particular interface to a particular version. As Web service producers are changed, the consumers can be migrated slowly to the new service. This way disruption is minimized.

Conclusion

Successful enterprises know that achieving an organization's goals is a mix of best practices around people, policies, and technology. This article has attempted to illustrate that it is not sufficient to adopt SOA technologies. It is important to adopt best practices around how a group is organized and how it behaves. It is also important to adopt a set of policies that ensure smooth, visible operations and minimal effort in the face of change. This article has attempted show some of the current thinking around best practices from Epiphany, Epiphany's customers, and leading analyst firms. The nascent body of SOA best practices is still developing.

Resources

- Some efforts in this area, like xNAL, are by no means universally accepted. See http://xml.coverpages.org/xnal.html for more details.
- Gamma, E., et al. (1995). Design Patterns: Elements of Reusable Object-Oriented Softwar. Addison Wesley.
- SOA Blueprints: www.middlewareresearch.com/soa-blueprints/

About the Author

Bill Roth is VP of product marketing at BEA. He wrote this article while chief technology evangelist at Epiphany. With over 15 years in this industry, he has played numerous roles including product marketing and product management for the initial launch of J2EE at Sun.

■ ■ ■ br@sys-con.com

▸ *Continued from page 8*

- Design BPEL processes from the top down, not the bottom up. When designing and implementing BPEL processes, it is critical to ensure that the BPEL process directly models the underlying business process in the most abstract form possible, without losing any critical process data.
- Fault-tolerant BPEL servers cannot compensate for BPEL scripts that do not implement sufficient fault-handling logic. Be prepared to handle faults, especially when invoking on or receiving from a remote Web service.

As organizations build out services from their existing infrastructure using the ESB, there is no question that the next step is to orchestrate those services to support new and existing business processes. BPEL enables the creation of meaningful service-based business processes that promise to become more important as the adoption of Web services and SOA continues.

About the Author

John O'Shea is a senior consultant with Cape Clear Software. He advises companies on best practices in the use of Web services, SOA, and ESB to integrate applications and build out new services. John previously worked for software firms such as Headway Software, IBM, SI Corporation, and IONA Technologies.

■ ■ ■ john.oshea@capeclear.com

Could Enterprises Be Pushing Back on ESBs?

Debate continues over messaging format viability

I just got back from the Gartner Application Integration show. This is perhaps the seventh of these conferences I attended, including the first one back in 1998. The good news is that the conference was packed and application integration seems to be making a strong resurgence with a little help from something called Web services. The bad news is that many of the attendees appeared confused, and the world of application integration and Web services/SOA seems more complex and difficult to grasp for the rank-and-file IT person than it was in the past. There don't seem to be as many clear answers these days, which is too bad because the problems seem to be getting bigger and integration is becoming a top-three IT priority according to the analysts.

WRITTEN BY
DAVID S. LINTHICUM

The real surprise for me was the backlash against the concept of ESB (enterprise services bus) and its proper application within the enterprise as well as in the world of SOA. It seems to be falling off the current buzzword list as enterprises press forward with their SOA planning and implementation work, choosing more Web services–compliant messaging infrastructures instead.

In fact, I was most surprised to hear an analyst admit during a presentation that there was a flaw in the ESB concept, as end user organizations may over-integrate proprietary ESBs. This comment points to the fact that many ESBs will layer on top of existing messaging systems already in place, and may also be redundant with newer, more Web services–compliant approaches. In other words, you could end up with your traditional messaging system alongside your new ESB messaging system, as well as a new SOA infrastructure, all with redundant messaging capabilities.

The analysts could have a point there when considering the long-term strategic directions of SOAs. Indeed, many are calling ESBs "transitory" technology, something that will bridge enterprises to Web services, but perhaps not be the end solution.

Of course the ESB guys would argue that their ESB solution is indeed Web services/standards-based, albeit the underlying technology is based in Java or proprietary messaging systems. If you deal with the technology through standard interfaces, you should not be concerned with what happens behind the scenes. Moreover, they have all vowed to keep up with emerging Web services standards.

What's driving the debate even further is the appearance of new SOA infrastructure players with ESB-like capabilities, such as Blue Titan and their Network Director product with embedded queue technology in its Web services routers. Like the ESBs, they provide guaranteed message delivery, but they are also compliant with the current version of WS-ReliableMessaging.

The Essence of ESBs

I've heard the term ESB from both the analysts and vendors (both claiming ownership, by the way) to describe a technology stack that places service-oriented interfaces on top of messaging systems, such as JMS. Instead of invoking the Java interface to push and pull messages from a queue, you leverage a Web services interface. Going further, many of the ESB vendors have added "traditional EAI" features to their stacks, including transformation, routing, flow control, and process integration. Indeed, many have described ESB as kind of an EAI light.

Thus, those who employ an ESB would use it primarily for information-oriented integration, leveraging the ESB to move information between applications, which is traditionally the role of a messaging system. They also add transformation, routing, and other information brokering capabilities. However, most ESBs lack service- or behavior-based integration capabilities and the ability to effectively create and manage composite applications through this infrastructure. They do support simple orchestration, but again, typically around information movement.

The lack of true service-based integration is really the kicker for me when considering an ESB for use in an SOA. I believe the real value of creating an SOA and moving to

Web services is the ability to reuse services inside and outside of an enterprise and create new applications by assembling services. This is how SOAs make you money. Also, SOAs provide us with the ability to create orchestration layers, abstracting services, and information flows for the purposes of creating business processes.

Thus, the purists are putting forth the argument that most SOAs should leverage pure Web services standards, otherwise integration and interoperability will be compromised. They further state that enterprises are going to double or triple up on messaging layers, implementing a JMS-based messaging system as well as a pure Web services–compliant messaging system at the same time. Moreover, ESBs typically exist with legacy messaging systems that are already in place. Furthermore, now that WS-ReliableMessaging is almost ready for primetime, as well as WS-Reliability (Sun's version, now ratified by OASIS), we're nearing the time when we can build Web services–compliant messaging systems without the need for an ESB. Thus SOA architects have other options today.

So, What's the Deal?

Clearly, building an SOA is one of the most confusing things to do these days. Not because the technology is so difficult to leverage; it's not, but there are many decisions to make, such as wading through competing technology as well as overlapping and numerous standards.

ESBs, at their essence, are reinvented messaging systems, but do have value for those who need messaging systems today and want tighter integration with their new service-based infrastructure. They do raise questions, however, as to their long-term value in light of newer, more Web services–compliant technology.

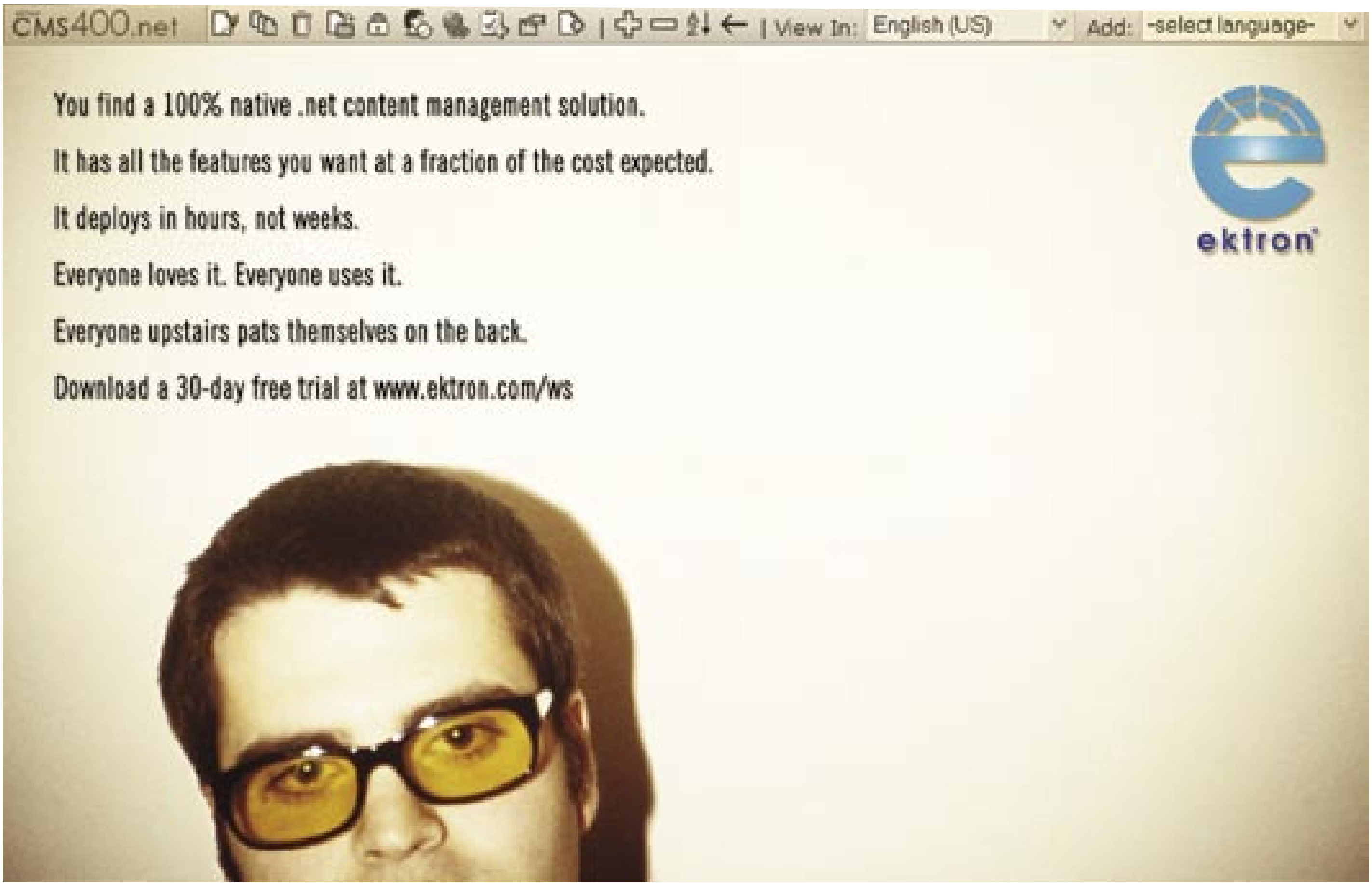
ESB vendors, however, are not going to stand around and watch their technology become irrelevant. Count on them to morph their product to meet changing expectations, including addressing emerging and existing Web services standards. Indeed, almost all ESB vendors have aligned with Web services standards, including WS-ReliableMessaging or WS-Reliability.

The key issue is that of fit. If you already have an existing enterprise middleware layer and are moving to service-oriented or Web services–compliant integration, leveraging an ESB is a step you may be able to skip. However, ESBs do seem to be a good fit for those enterprises performing simple information movement between stovepipes that don't need the heavy duty nature of more traditional integration technology. Once again, you have to map the appropriate technology to the problem, as well as consider longer-term strategic direction.

About the Author

David Linthicum is the CTO at Grand Central Communications (www.grandcentral.com), and a leading expert in the application integration and open standards areas. He has held key technology management roles with a number of organizations, including CTO of both Mercator and SAGA software. David has authored or coauthored 10 books, including the groundbreaking and best-selling *Enterprise Application Integration*, released in 1998. His latest book is *Next Generation Application Integration, From Simple Information to Web Services*.

■ ■ ■ linthicum@att.net


CMS400.net
View In: English (US)
Add: -select language-
You find a 100% native .net content management solution.
It has all the features you want at a fraction of the cost expected.
It deploys in hours, not weeks.
Everyone loves it. Everyone uses it.
Everyone upstairs pats themselves on the back.
Download a 30-day free trial at www.ektron.com/ws
ektron

Oracle Application Server 10*g*

A robust option for the enterprise

A number of solutions exist for creating Java-based Web services from a variety of different providers. Options range from individual processing engines that plug into existing application servers to large enterprise-class platforms in which Web services is one of many components. Each option provides its own set of challenges and benefits while addressing different types of requirements. For the enterprise, Oracle's Application Server 10*g* provides a robust option.

WRITTEN BY

BRIAN BARBASH

The 10*g* version of the application server from Oracle provides a highly flexible and scalable Java-based Web services capability. The tools provided by the system allow for almost any Java object or PL/SQL function to be exposed as a Web service.

Web Services Support

The Oracle Application Server Web services architecture in 10*g* provides the infrastructure to expose a variety of objects as Web services, including:

- Stateless Java classes
- Stateful Java classes
- Stateless session EJBs
- Stateless PL/SQL functions or procedures
- Java Message Service (JMS) objects

Figure 1, from the application server's documentation library, identifies the supporting components for each of the above sources. The boxes on the right side of the figure represent individual servlets that handle the documents sent to and from each type of source component. For Web services that are stateful, HTTP session objects manage state between service calls.

Both RPC and document-style Web services are supported by this architecture. The key difference between the two styles within the application server is the set of parameters and data types allowed in the implementing class. In the RPC style, Oracle supports instances of basic Java primitives and object types as parameters, along with Element, Document, and DocumentFragment objects from the DOM (XML API). Document-style Web service implementation classes only support instances of the Element object from the DOM (XML API).

Tools for Exposing Services

If you're using Java classes to implement Web services, the development should follow the same sound principles as when developing any other classes that will be hosted in an application server, since there are no physical limitations beyond those described in the previous section. Once the implementation classes are built, the developer must create the configuration file to describe how the service is to be assembled by the application server. This is a basic XML document not unlike a J2EE or Web application deployment descriptor. It contains information about the service itself, including name, description, and path information, as well as the individual services and their implementation classes (or other objects, depending on the implementation type). For the purposes of this article, a simple Document-style Web service was created as a standard Java class to accept an incoming document and store it in a repository.

With the configuration file complete, Oracle Application Server provides a tool called the WebServicesAssembler that reads the file information and generates the appropriate class files and deployment descriptors, and optionally the WSDL file and proxy classes for the service. This is a command-line utility that can easily be incorporated into any Ant script as part of a standardized build process. Inspection of the generated EAR file shows a Web application module with a specialized servlet for handling document-style Web services.

JMS Web services are slightly different because two JMS destinations and possibly an MDB are required, depending on implementation. In this scenario, a SOAP call

ORACLE

Company Info

Oracle Corporation
500 Oracle Parkway
Redwood Shores, CA 94065
www.oracle.com
Phone: 1.800.ORACLE1
http://www.oracle.com/technology/products/ias/index.html

At the time of this writing, it was indicated that an upgrade to the Oracle Application Server was forthcoming with support for J2EE 1.4 Web Services, SAAJ 1.2, Apache WSIF, SOAP 1.1/1.2 and WSDL 1.1. Adds support for the WS-Security standard and provide a reliable messaging framework that supports the WS-Reliability standard. This release includes a new management console for managing Web services, a full design time in Oracle JDeveloper and for non-Oracle-IDE environments, a command line tool, and a set of Ant tasks for creating Web services.The new version is now available

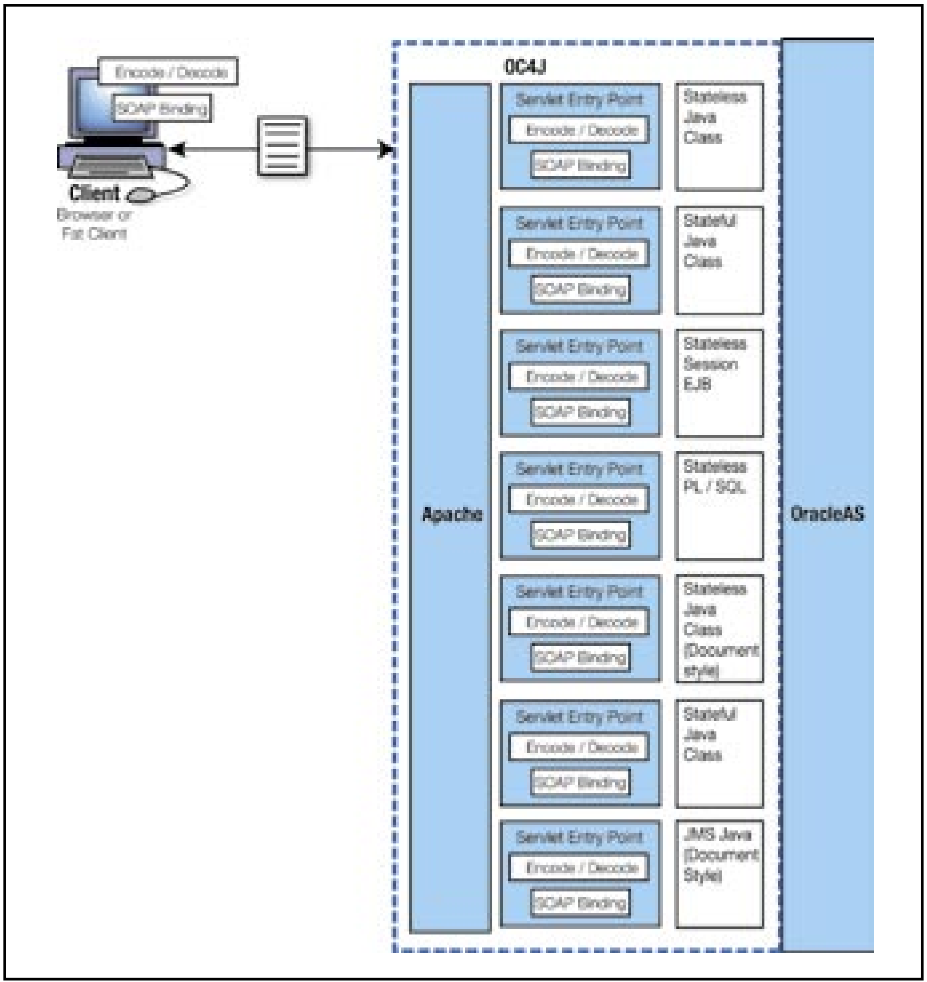


FIGURE 1 **Oracle Application Server Web services architecture**

goes through the following progression:

1. The Web service client sends a SOAP message to an HTTP servlet
2. The Web service listener servlet drops the message to a JMS destination
3. An MDB or alternate JMS client picks up the incoming message and processes the contents
4. The result of the operation is placed on a second JMS destination
5. The Web service listener servlet returns the service result to the Web service client via a SOAP message

The configuration file for JMS Web services also differs in that the connection factories and queues for sending and receiving the SOAP document from the Web services listener servlet are identified instead of the implementation classes.

Similar changes are needed when developing Web services that are implemented by PL/SQL objects. In this scenario, the configuration file identifies the username and password of the host database, its JDBC URL, the JNDI name of the connection pool, and the package and procedure to execute.

In addition to the Web services architecture described above, Oracle Application Server 10*g* provides an alternative method for Web services implementations called OracleAS SOAP. This toolkit is based on the Apache SOAP implementation. It should be noted, however, that the Oracle Application Server documentation recommends that the Oracle Application Server Web services architecture be used to create and deploy new Web services applications.

Oracle UDDI Registry

Cataloging and publishing of available Web services is supported by the UDDI registry provided in Oracle Application Server 10*g*. This repository includes support for the SOAP API as defined in version 2 of the UDDI specification. Web services may also be published to the UDDI registry at deployment time from the Oracle Application Server Administrative Console.

Securing Web Services in Oracle

Oracle Application Server 10*g* allows administrators to secure Web services using the following techniques:

- Web services over SSL
- HTTP Basic Authentication
- Authorization via Oracle's implementation of JAAS

When interacting with secure Web services hosted on an Oracle Application Server instance, the client application must have the relevant system properties set. If JAAS is implemented for authorization, the username and password provided by the calling client may be used to retrieve a Principal object from the User Manager to be passed along to any relevant processes.

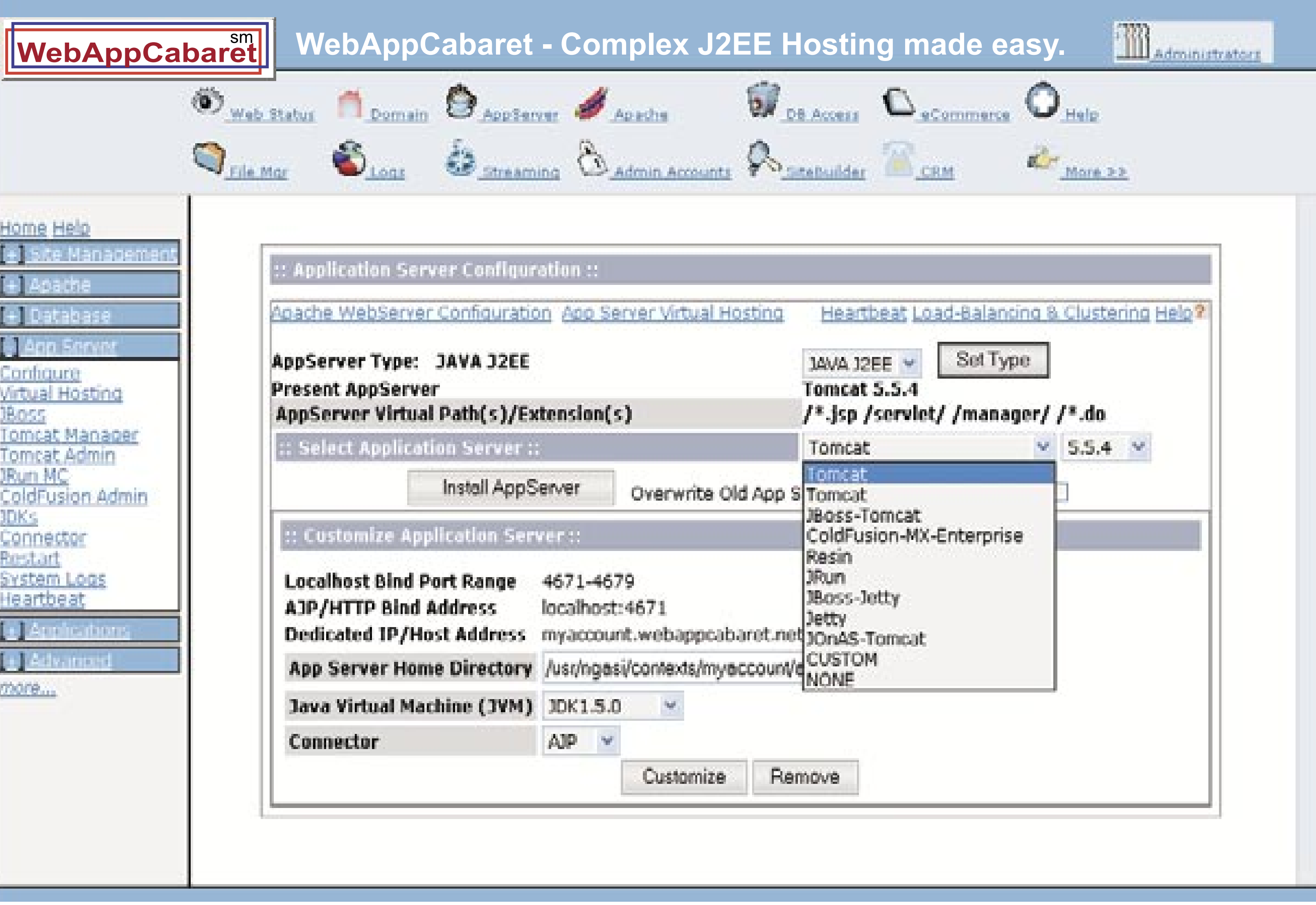
Conclusion

Oracle Application Server 10*g* provides a powerful, flexible, and scalable Web services architecture for Java-based systems. The tools provided isolate the developers from the required code to manage the details of processing SOAP requests and allow them to focus on creating business logic. The UDDI server provides the cataloging and directory services to publish available services. Overall, the 10*g* version of Oracle's Application Server provides a complete and robust solution for delivering Web services in Java.

About the Author

Brian R. Barbash is the product review editor for *Web Services Journal.* He is a senior consultant and technical architect for the Envision Consulting Group, a management consulting company focusing on contracting, pricing, and account management in the pharmaceutical industry.

■ ■ ■ bbarbash@sys-con.com



:: Application Server Configuration ::

AppServer Type: JAVA J2EE	JAVA J2EE
Present AppServer	Tomcat 5.5.4
AppServer Virtual Path(s)/Extension(s)	/*.jsp /servlet/ /manager/ /*.do
:: Select Application Server ::	Tomcat 5.5.4

Overwrite Old App S

:: Customize Application Server ::

Localhost Bind Port Range	4671-4679
AJP/HTTP Bind Address	localhost:4671
Dedicated IP/Host Address	myaccount.webappcabaret.net
App Server Home Directory	/usr/ngasi/contexts/myaccount/c
Java Virtual Machine (JVM)	JDK1.5.0
Connector	AJP

Imagine a hosting company dedicated to meet the requirements for complex web sites and applications such as those developed with Java J2EE.
At WebAppCabaret our standards based process and tools make deploying Java J2EE applications as easy as a point-and-click.
We call it **Point-and-Deploy Hosting**.
Our advanced NGASI Web Hosting management Control was designed for the hosting and management of complex web sites and applications thus cutting down on maintenance time.

Backed by an experienced staff as well as a **Tier 1 Data center** and network. Our network is certified with frequent security audits by reputable security firms.

All hosting plans come with **advanced tools** to manage your application server and Apache web server. Not to mention the other features, such as virus-protected email, bug tracking, and many more portal components.

Complete power and control at the tip of your fingers. We take care of the system and hosting infrastructure so you can concentrate on development and deployment of your application. That is **real ROI**.

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

NGASI POWERED

WebAppCabaret sm

http://www.webappcabaret.com/jdj.jsp 1.866.256.7973

Point-and-Deploy J2EE Hosting

Prices,plans, and terms subject to change without notice. Please log on to our website for the latest price and terms. Copyright © 1999-2005 WebAppShowcase • All rights reserved • Various trademarks held by their respective owners.

ESB MYTH BUSTERS

Clarity of definition for a growing phenomenon

Since releasing my latest book, *Enterprise Service Bus* (O'Reilly Media, 2004), I have been doing a fair amount of visiting corporations, conducting seminars, and generally discussing with enterprise architects the subject of enterprise service-oriented architecture (SOA) and how an enterprise service bus (ESB) backbone can be leveraged to provide a framework for an enterprise SOA. Along the way, I have been asked many questions about the nature of an ESB. I have also fended off some misconceptions that have been growing in the general IT population regarding what an ESB is and when, where, and how it can be used. I have gathered together the most popular questions and misconceptions, and offer some clarity in the form of a "top ten" list.

WRITTEN BY
DAVE CHAPPELL

Myth #1. ESB is just a new name for EAI.

While many IT architecture groups are focusing on building SOAs, they still inevitably beg the question of "how is ESB different from EAI?" An ESB is an infrastructure for building an enterprise SOA, and is capable of being used in a more general way than a conventional EAI broker.

According to Forrester Research, an ESB helps enterprises obtain the value of SOA by increasing connectivity, adding flexibility that speeds change, and providing greater control over use of the important resources that it binds.

An ESB can be used to handle integration projects that have traditionally been relegated to EAI tools. However, an ESB can also be used for establishing B2B relationships across companies.

An ESB provides EAI capabilities, but is based on a fundamentally different architecture that is providing the basis of an industry transition from traditional integration to coordinated service interaction. EAI brokers are historically implemented as a monolithic stack, using centralized hub-and-spoke architecture.

An ESB provides the same base functionality as an EAI broker – connectivity, application adapters, routing of messages based on rules, and data transformation engine – yet, in an ESB, these capabilities are themselves SOA based in that they are spread out across the bus in a highly distributed fashion and hosted in separately deployable service containers. This allows the selective deployment of integration broker functionality exactly where you need it, with no additional over-bloating where it's not needed. The distributed nature of the ESB container model allows the independent scalability of integration components, which are plugged into your SOA as event-driven services on an as needed basis.

In order for an integration broker to be truly capable of supporting an SOA, and to be considered a true ESB, it would need to have its base functions broken up into their constituent parts, which would then be capable of being separately deployed across the bus while working together in harmony as necessary.

Let's use an example of an XSLT-based transformation engine that accepts an incoming XML document and applies an XSLT style sheet to it in order to produce an outgoing document in another XML format. I can tell you that there is nothing that can chew up computing resources more than the parsing and manipulation of XML. If this particular XSLT transformation sits between two popular applications that communicate regularly with each other, then that individual transformation can become a performance and scalability bottleneck. If you are using a monolithic hub-and-spoke integration broker approach, in order to remove the bottleneck and scale up the deployment you would need to either install that integration broker on one big powerful machine, or install the integration broker across multiple machines – just to support that one transform scenario! All the while, the other integration broker capabilities, such as the execution of routing rules, are competing for the same computing resources as the transformation operation.

In contrast to the monolithic hub-and-spoke architecture of an integration broker, the foundational core of an ESB provides a distributed services architecture. This architecture is built for integration and has the ability for integration broker functionality, such as message routing, data transformation, and application adapters to be selectively deployed on an as-needed basis. These are separate integration services that are a natural part of an SOA processing pipeline across the bus.

An individual XSLT transformation can be deployed as a service in its own ESB service container, and multiple instances of that container can be load-balanced across many machines. If the ESB container implementation is cross-platform, then you can be flexible as to what kinds of machines you spread the transform service across – Linux boxes, Solaris boxes, Windows boxes, and so on. And for those of you who don't find solace in the architectural purity of this discussion, consider this: the ESB vendors who are leading the charge in defining and delivering ESB products are also putting forth a license model where there is no additional cost for deploying as many of these lightweight ESB service containers as necessary to get the job done.

The integration services provided by the ESB can be combined with other services into SOA-based processing pipelines that can span business boundaries. The distributed services in an ESB can be combined with itinerary-based routing (see Myth #7) to allow self-directed, message-oriented service interactions, which allow different parts of the ESB to operate independently of one another, without relying on a centralized routing engine.

Myth #2: Microsoft is building an ESB with their "Indigo" project.

Microsoft's Indigo combines Microsoft's Messaging Queuing, its Component Object Model COM+, .NET, and Web services. What they are building is a message bus with Web services extensions. This is very different from an enterprise bus. A messaging bus exposes the details of lower-level messaging techniques and requires the writing of code to define the relationships between applications and services. The mantra of the ESB is configuration rather than coding, which removes the necessity of hardwiring relationships between interconnected applications. An ESB facilitates and promotes the use of loose coupling between applications, which are exposed through the bus as event-driven services. The good news here is that applications built on Indigo will at least be message based and therefore easier to integrate through an ESB.

That said, there are elements of the BizTalk Server that, correctly combined with Indigo, could start to look like an ESB. However, there is one missing element in the formula, in that BizTalk is still a hub-and-spoke integration broker and is subject to all of the caveats mentioned in the ESB versus EAI discussion. You can't split out the XML transformation engine from the rest of the BizTalk Server and expect to run it as a load-balanced service across multiple machines without incurring the cost and overhead of the entire Biztalk Server (see the previous discussion on EAI vs. ESB).

Myth #3: The adoption of WS-* specifications, such as WS-Reliability and WS-Reliable Messaging, obviate the need for an ESB.

An ESB should be designed to accommodate these evolving specifications as they become mature and achieve commercial viability. Evolving WS-* specifications will help make application endpoints more interoperable through an ESB.

As part of the evolving standards process of Web services specifications, there exists much uncertainty due to the many overlapping efforts underway. As these specifications mature and achieve widespread adoption, they will still require an infrastructure to support them. An ESB can provide a consistent model for building, orchestrating, and managing SOAs, while insulating the IT organization from changes in underlying interoperability standards.

A WS-Reliability implementation requires that there be an industry-proven reliable message persistence and store-and-forward processor to support it. A foundational component of an ESB is an enterprise messaging layer that provides quality of service of message delivery through messaging conventions such as message persistence, store-and-forward delivery, message acknowledgements, and interfaces with external XA-compliant transaction managers. The ESB implementation may also provide transparent routing of messages across sophisticated network topologies, and continuous availability of the messaging infrastructure through a fault-tolerant messaging server architecture. The science of making all of this work together to ensure reliability under high-stress enterprise environments requires many person-years of effort to get right.

> “An ESB is a highly distributable backbone upon which to build enterprise service-oriented architectures”

That being said, ESBs that are implemented today using a proprietary messaging layer should also adopt one or more of the WS-Rel* as additional protocols or "on-ramps" for getting on and off the bus. However, it is not a one-size-fits-all solution and many combinations of messaging and protocol support are necessary.

Myth #4: Pattern or Product: The term "Enterprise Service Bus" (ESB) is not really a product category; it is simply an abstract concept that can be applied toward a coupling of an existing application server and integration middleware.

An ESB is a highly distributable backbone upon which to build enterprise service-oriented architectures (SOA). Enterprises build service-oriented architectures, and an ESB is the backbone upon which to build it. As a result of the tumultuous disruption to the integration market caused by the advent of the ESB, some established integration vendors have laid down a smokescreen by saying that an ESB is simply an abstract pattern that can be overlaid across a composition of existing middleware and application server infrastructure that they already have. In fact, an ESB is definitely a coherent piece of infrastructure that you have been able to purchase from a number of vendors for at least a couple of years now. There are already dozens of ESB deployments in place across a variety of vertical industry segments including manufacturing, financial services, telco, and retail.

The definition of an ESB includes these basics:

- A distributed services architecture, which includes a lightweight container model for hosting integration components as remote services
- An enterprise messaging backbone that provides reliable delivery of messages between applications and services
- XML Data transformation
- Service orchestration and intelligent routing of messages based on their content
- A flexible security framework
- A management infrastructure that lets you configure, deploy, monitor, and manage your remote services

The distributed services architecture of the ESB allows the referencing of services via abstract endpoints, which are globally accessible across a federated namespace. The distributed services architecture is layered upon an interconnected system of lightweight service containers that allow remote services to be configured, deployed, managed, and monitored. These service containers are held together through a standards-based messaging backbone that enables scalability, continuous availability, low-latency throughput, and consistent security and quality of service (QoS) across the enterprise.

Myth #5: ESBs compete with the J2EE app server products.

An ESB is highly complementary to a J2EE app server. J2EE app servers can integrate well with other app servers, and with non-J2EE environments, by plugging into the ESB using standard interfaces such as JMS, MDB, JCA, or Web services.

Most adopters of ESB technology are also heavy users of application server technology. These customers use the combination of their application server and ESB as best-of-breed components in their integrated environment – the app server for hosting business logic and serving up Web pages in a portal server environment, and the ESB for integrating the app server with a variety of back-end applications and data sources across their extended enterprise.

Myth #6: Portals can be connected to back-end systems by simply using a Web service call.

While Web service calls can theoretically be used to connect portals with back-end target systems, this approach won't scale past a few back-end systems. An ESB allows the portal server to have a single interface to the bus, with the bus providing the mediation between the diverse connectivity options, protocols, security, and data formats across all of the possible back-end systems that a portal server may call upon to fulfill its requests.

Using an ESB as the layer between the portal server and the various back-end applications that the portal server needs to interact with, ESB adopters are building for the future by providing a more scalable and flexible SOA that is capable of handling the ever-expanding scope of integration as the portal project becomes more successful and business requirements change.

Myth #7: ESBs will be obsolete once BPEL is widely available.

An ESB may support multiple ways of coordinating the interaction between event-driven service invocations using formal business process definitions. BPEL (Business Process Execution Language) is one way of doing it, and there are others as well. An ESB also has itinerary-based routing, which provides a message with a list of routing instructions. These routing instructions, which represent a business process definition, are carried with the message as it travels through the Bus across service invocations. The remote ESB service containers determine where to send the message next.

Itinerary-based routing significantly contributes to the highly distributed nature of the ESB, as there is no centralized rules engine to refer back to for each step in the process. A centralized rules engine for the routing of messages, such as those offered by the typical hub-and-spoke EAI broker approach, can be a bottleneck, and also a single point of failure. The use of message itineraries, messages and process definitions is self sufficient and can therefore allow different parts of the ESB to operate independently of one another.

Message itineraries are most useful for discreet process definitions that are stateless and usually contain a finite set of steps that don't take extended periods of time to complete. Gartner refers to this type of process definition as "microflows." Simple branching within itineraries may occur based on the use of content-based routing services.

When more sophisticated process definitions are required, a process orchestration engine may be layered onto the ESB as an additional service. The process orchestration may support stateful processes, which can span long durations of time. It may also support parallel execution paths, with branching, and merging of message flow execution paths based on join conditions or transition conditions being met. Such a process engine may support BPEL, or some other process definition grammar such as ebXML BPSS. Sophisticated process orchestration can be combined with stateless itinerary-based routing to create an SOA that solves complex integration problems.

Myth #8: The ESB technology category, like so many others, seems to have come out of nowhere and is now barreling its way up the hype curve and rapidly approaching the "trough of disillusionment."

The ESB concepts were created as a result of working with IT thought leaders across a variety of industries, including manufacturing, e-marketplaces, telco, financial services, and retail. The ESB as a concept was born out of a necessity, based on their desire for a new approach over distrib-

WSJ Advertiser Index

Advertiser	URL	Phone	Page
EclipseCon 2005	www.eclipsecon.org	503-252-5458	3
Ektron	www.ektron.com/ws	866-4-EKTRON	17
EV1Servers	www.ev1servers.net	800-504-SURF	6
Gilbane Conference	www.lighthouseseminars.com	781-821-6734	33
Global Knowledge	www.globalknowledge.com/lw	919-461-8600	27
HostMySite.com	www.hostmysite.com/mxdj	877-248-4678	Cover IV
Itemfield	www.itemfield.com/itsolutions	800-838-1055	39
Mindreef	www.mindreef.com/tryout	603-465-2204	9 & 11
OpenLink Software	http://virtuoso.openlinksw.com	800-495-6322	19
Parasoft	www.parasoft.com/soaptest_WSJ	888-305-0041	Cover II
Parasoft	www.parasoft.com/jtest_WSJ	888-305-0041	5
Promind Systems	www.xslmaker.com	415-359 9117	35
Secrets of the XML Masters	www.sys-con.com/freecd	888-303-5282	57
Web Services Edge	www.sys-con.com/edge	201-802-3066	41-50
WebAppCabaret	www.webappcabaret.com/jdj.jsp	866-256-7973	21
WebRenderer	www.webrenderer.com	61 3 6226 6274	Cover III

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

by their agents or agencies who are acting on behalf of the advertiser. This index is provided as an additional service to our readers. The publisher does not assume any liability for errors or omissions.

uted computing models and EAI technologies they had already been moderately successful with. These IT thought leaders all came with a common request: "We really like your distributed messaging infrastructure, and we would like to build upon it a standards-based event-driven SOA for integration. We would like it to include things like Web services, XML data transformation, content-based routing, and a service invocation model based on distributed process coordination." Because of this, the concepts presented in the ESB architecture are sound; they are grounded in reality. Also because of this, ESB technology has been adopted as it has been built. There are 100s of ESB deployments already in use today in areas such as supply chain and logistics automation, straight through processing in financial services, real-time provisioning in Telco, and remote storefront integration in retail.

> "Evolving WS-* specifications will help make application endpoints more interoperable through an ESB"

Myth #10: An ESB container can be implemented using an EJB container.

One of the key components of an ESB architecture is a highly distributed, lightweight service container. The service container allows the hosting of integration components as event-driven services, such as a content-based routing service that applies an XPath expression to an XML message to determine where to route it next. The service container can also host custom services or specialized adapters for hooking into packaged applications.

Unlike its distant cousins, the app server container and the integration broker, the ESB service container allows the selective deployment of integration services exactly when and where you need them, and nothing more. In contrast, you need to install an entire appserver stack everywhere that an individual piece of integration functionality is needed. This results in what is referred to as the "appservers everywhere" problem. There is an unnecessarily high cost in licensing, installation, and cost of ownership over time associated with this practice.

The mantra of the ESB is "configuration rather than coding." In an application server–centric approach to integration, you typically write code to describe the dependencies between services. The EJB model follows the client/server model of interaction, which usually results in tightly coupled interfaces between services in an SOA, which is built into code, and compiled into class files that need to be modified and redeployed every time a change needs to be made.

In an ESB, a service is configured with information regarding its input and output channels for sending and receiving message-based request/response patterns and one-way event notifications that are then coordinated by the surrounding invocation framework – not by the service itself.

An ESB service can be configured and deployed, by merely supplying it with the XSL stylesheets, XPath expression, scripts, and parameters which are read in from a configuration repository. Once deployed, the implementation is remarkably resilient to change.

Myth #9: ESBs are simply plumbing and do not provide sophisticated tooling, such as a graphical editor for designing business process flows.

There is a new breed of IDE, which Gartner Group refers to as an ISE (integrated services environment), that allows you to design, configure, test, and debug the integration services that you develop when building an SOA with an ESB. Using a graphical interface, an integration architect draws diagrams using UML notation to describe process definitions. You may also use the ISE to graphically create data transformations between different data formats, and create and debug XSLT stylesheets.

Summary

To sum this up, make sure that your understanding of ESB contains these things:

- An ESB provides the backbone for building an enterprise SOA-based integration environment.
- The evolving WS-* specifications will help make ESBs even more interoperable than they are today. Adopting an ESB today will allow you to build for the future and be capable of adapting to the WS-* specifications as they become commercially viable.
- ESB is ***not*** just an abstract pattern. It is a product category with a distinct definition and many vendor offerings.
- ESBs and application servers are highly complementary.
- ESBs can help portal server integration to back-end systems by providing the necessary diversity in connectivity and scalable infrastructure.
- ESBs provide many choices for coordinating service interactions.
- ESB technology is grounded in reality and is already being adopted by many industries.
- ESBs can provide the higher-level visual tools for integrating services in an ISE environment.
- ESBs provide a service container environment that is lightweight, configurable, and highly distributable.

About the Author

Dave Chappell is vice president and chief technology evangelist for Sonic Software (www.sonicsoftware.com), the inventors of the enterprise service bus. Dave is also author of the book *Enterprise Service Bus* (O'Reilly Media).

■ ■ ■ chappell@sonicsoftware.com

Are you Red Hat® ready?

Distributed Parallel Computing with Web Services

A pivotal role on the back end

Web services technology has become the ubiquitous connectivity fabric amongst diverse business domains and technical camps. At the same time, distributed parallel computing is becoming the de facto architecture for managing the performance of computationally intensive, long-running programs.

WRITTEN BY
LABRO DIMITRIOU

So, is it counterintuitive to consider Web services when pursuing performance improvement of compute-intense, long-running applications? It may seem that way but, most amazingly, Web services play a critical role not in one but in two areas of High Performance Computing (HPC) and distributed parallel computing:

- Communications/deployment
- Classifications/discovery services of resources

In other words, Web services play a role in the application adaptation and infrastructure layer, respectively. Sensibly enough, Web services deliver again on the promise of semantic and syntactic universal collaboration and wide acceptance in yet another coming-of-age technology.

This article looks at how the Web services scenario is unfolding in the distributed parallel computing space. First it discusses the developing infrastructure standards, followed by some definitions, and then delves into the real grid opportunity: to improve application response time while workload balancing. I'll conclude with a recent project implementation.

Grid Provides the Infrastructure for Parallel Distributed Computing

The Globus project and its proposed Open Grid Services Architecture (OGSA) specification describe how Web services can facilitate the creation, life-cycle management, and security requirements for reliable and industrial strength Grid Services Architecture (GSA). However it only seems to address the infrastructure and resources associated with the grid. So what about the application, you might ask. After all, the grid is only as valuable as the applications you run on it.

The most common approach to enabling an application for a grid environment is to decompose the application into smaller, independent subtasks or job "chunks," submit the jobs to the grid (schedule and deploy), and hope for the best. This paradigm provides a good solution for batch cycle applications and is well suited for coarse-grain parallelizable applications.

But what about the standards, providing best practices, support, and tools for distributing more complex applications, sometimes referred to as "fine-grain parallelizable."

Furthermore, many have argued that such services are orthogonal to what OGSA addresses. Yes, I agree that the specification is about the infrastructure, but only for now. Remember the debate about middleware, EAI, application servers, and the J2EE stack? I foresee that the same line of arguments regarding distributed parallel computing will soon be coming to a grid near you. In fact the two technologies – J2EE and grid – have very similar characteristics, including services, specifications, best practices patterns, and supporting technology middleware, runtime containers, and desirable quality of services. And in both cases Web services play a key role. Perhaps the overriding similarity is n-tiered distributed computing.

Parallel Distributed Computing and the Grid

First, let's talk about grid and distributed parallel computing, or are these the same?

Some say it's a matter of definition again. This is especially true for uncharted technology territories: there seem to be as many definitions as opinions. However, some commonly used terms and concepts have been emerging. This is not surprising as parallel computing has been around for quite a while, for example as parallel computing execution on symmetric multiprocessing (SMP) machines or massively parallel computers. Programming to leverage these configurations required executing multiple parallel threads and sharing common memory – fast memory, that is. Silicon Graphics and a number of mid-range to high-range Sun and AIX multiprocessing machines are a fine example of parallel computing machines – the next best thing to Cray supercomputers!

These SMP configurations provide horizontal scaling by adding CPUs (typically up to 64). You add more processors until you can't add any more. But what happens when your application isn't completing fast enough and you need a 65th CPU? The only choices that don't require reprogramming are to buy a system with more powerful CPUs or a system that can support more CPUs. If this doesn't help the outlook is either chunking and distributing or bleak.

So the real distinction between distributed parallel processing and parallel processing is the access to the data that would have been in the shared memory in the parallel configuration. This could require a minor change and some data selection and movement logic being added, or it could be a major consideration because the movement of the data among the processors executing the instances of the application could take more time that the additional CPUs reduced. Typically, distributed parallel computing involves dozens or hundreds of computers, the grid or compute farm, and concurrently running components of an application.

I saved for last the notion of parallel running of an application. In a nutshell parallel computing exploits concurrency of execution, so no arguments here: it is parallel concurrently computing as opposed to serial computing. There is only one thing missing from the alphabet soup, the main character: the application.

Applications: Making Fast Faster

High Performance Computing is about doing more with less or making programs that run fast run faster! Consider an HPC application currently running as fast as possible using parallel computing (multiple system resident CPUs). Moving this application to a multiple system configuration typically requires reengineering the program. This translates to diverting developer focus away from new projects, and may require recertifying the application and hiring engineers conversant in distributed computing and data considerations. But most of the programs were designed with sequential flows in mind and the programmers (those who are mere mortals) typically think in terms of sequential flows and are most effective in designing, writing, and debugging sequential programs, albeit some are fluent with multitasking techniques at times. So where is the magic?

So the options to move programs from sequential to SMP machines are:

- Use special parallel compilers that leverage the multiple CPUs which at best offer less than optimal improvements without major tuning
- Use an advanced language that supports threads and write multithreaded programs. It's certainly not a piece of cake to develop, not to mention what happens to the application when "the thread guru" moves on to bigger better things or goes on in pursuing alternative life styles.

Developers and organizations recognize that the ever increasing business volumes will drive their already loaded SMP-based applications beyond the affordable expansion point, and are looking at ways to avoid this problem. They are looking at ways to distribute the individual program components using less expensive systems as part of cluster and grid strategy.

So the question remains: how do you improve the response time of an application by distributing it to a compute farm? If the application has a high-volume, sub-second response, transactional characteristic we are in luck. WebLogic and BEA's robust clustering technology can effectively distribute a high transactional application to a cluster. But there is a large class of application legacy of some sort, such as finance quant apps, engineering numerical analysis solvers, life science genome applications, or computational statistics which require processing large amounts of data and are compute-intense. These still need a solution.

The Compute-Intensive Application

To discuss the distributed parallelization options, it is helpful to further classify applications into stages based on the best practices parallel distributed computing design patterns they fit in. I've called them Stage I and Stage II variants of distributed applications.

Stage I distributed application candidates can be described as "same instructions small different data" design patterns. Consider the search for extra terrestrial seti@home project or the search of large Mersenne prime numbers – both items arguably on the far right of compute-intensive. In both cases the same application needs to be executed again and again using a small amount of different data; hence the name "same instructions different data." The only requirement to distribute the calculation is to use a job scheduler that distributes the calculation segments across computers. Sounds like a classic mainframe batch scheduler could do the job. No wonder IBM calls grid computing the next big thing – they have been providing the underlying services for years. From an application programming point of view, the grid is a vast number or transactions and jobs distributed by a scheduler.

These applications have one entry point and one exit point, and can have their work computed in parallel without sequential or data dependencies on the results of each. An application with these characteristics is referred to as "embarrassingly parallel."

A variant of the embarrassingly parallel application is a job or application that for historical reasons is being run as a sequential string of steps. These can easily be decomposed into smaller jobs and fall under the Stage I distribution umbrella. So all you need to distribute Stage I applications is to identify them, chunk them into parallel executable steps, and get a scheduler that deploys the “chunks” or steps. Several vendors, including Platform and Sun, provide such scheduling services.

While there are many applications that fit into the Stage I model, these applications have input data and usually intermediate result interdependencies. These are classified as Stage II distribution applications. You will recognize these as having the sources of their resource consumption embedded in loops with complex data dependency requirements. In other words, you can’t just chunk the application and expect it to run faster with the same results

A few different algorithmic patterns immediately come to mind as Stage II distribution application candidates: serial nondependant, master-slave, binary non-recombining tree, simplex optimization algorithms, and so on.

Consider a simple binary search algorithm for finding the maximum from a sequence of numbers used in a sorting problem. Although it’s a simple, commonly used algorithm, it exhibits the structural characteristics of Stage II distribution applications. You can devise a simple Web service that receives two numbers and returns back the maximum. A simple control master program could generate slave/worker services executing the comparisons on a compute farm, and returning the final result back to the parent/root node of the tree. The commercial tools for building such solutions include low-level parallel middleware (e.g., MPI and PVM), and a few higher-level paradigms that provide virtual shared memory or shared processes paradigms such as Java Spaces (a technology transfer of Linda Spaces), GemFire from Gemstone, and GigaSpaces. These offer generic supporting middleware services, with which the programmer must be conversant in order to enable the distribution communications and data transfer.

A recent entry that approaches such challenges differently and focuses on the application rather than on the infrastructure is ACCELERANT from ASPEED Software. It offers the developer a high-level algorithmic and computational pattern-aware interface that can be inserted in existing or new applications. This approach masks the application developer from having to deal with the middleware and distributed expertise while providing the resultant application with the use of the required runtime services to optimally manage the application execution across all instantiations of the execution.

An even more challenging variant of Stage II applications is one characterized as “impossible” to parallelize. Examples of this variant are step-wise iterative algorithms, where each step requires the computation of the previous one. A simple example is the common summation technique for adding a sequence of numbers. You iterate through the sequence of numbers and at each loop you add the next number to a tally. It turns out that even these Stage II variant algorithms can be recast to be handled like the easier Stage II algorithms, e.g. a binary tree master-slave algorithm can be applied to the above summation problem. This greatly simplifies the programming effort but obviously requires reverification since the algorithm has been altered. Other advanced techniques such as genetic algorithms are also available but their discussion does not belong to Web Services Journal – not until there is a Web services solution for them!

Now let’s move on to how Web services facilitate grid enabling of a complex compute-intense application and harnessing the computing power of HPC center for fast pricing portfolio bond options. The Callable Bonds Portfolio pricing was selected because it is a representative class of particularly complex Monte Carlo simulations that yield greater accuracy.

Now is as good a time as ever to shed light on the celebrated Monte Carlo techniques. First, it’s not just a jargon to confuse the unwary. It’s a real scientific tool. Monte Carlo techniques are counterintuitive in the sense that they use probabilities and random numbers to solve some very concrete real-life problems. Buffon’s Needle is one of the oldest problems in geometrical probability tackled with Monte Carlo. A needle is thrown on a lined sheet with a distance between the lines that is the same as the length of the needle. Doing the experiment many times computes the number π (pi) of the circle, with great accuracy I must say. You can design a Web service that executes ranges of millions of throws on a compute farm. The master control program (server-side Web service) aggregates the experiments and serves you back the value of pi!

The Use Case Requirements

So let’s talk about the use case designed and implemented at a buy-side financial services boutique. The business problem was to price a portfolio of callable bonds using Monte Carlo (MC) techniques. Think of a bond as a series of cash flows. You pay a price to buy it at issue day. Then every so often, say six months, it pays a coupon back. At maturity, the bond pays back its valued principal. Pricing a bond means to access its value at any given time. One popular way of pricing is to run complex computational statistical techniques called Monte Carlo simulations. Callable bonds have the added complexity that can be, well…called at any day before maturity. In order to anticipate the value of a hypothetical call, you must run additional “what if” scenarios. This happens to be one of the easier examples of a huge set of analysis, modeling, pricing, and risk assessments being used and in need of distribution in order to run within very stringent time constraints at an affordable price.

The company had existing C++ legacy code implementing the pricing model. Some of the bond portfolio calculations could take over an hour on the existing hardware. The new business requirement was to make the response time less than 30 seconds. Given the needs and the current implementation something had to give, and adding expensive cycles and reengineering the application was very risky – pardon the pun. The front GUI presented yet another challenge. The trading desk uses a new Java-based front-end trading system, but the sales desk primarily uses Excel spreadsheets for pricing for simplicity and easy of use.

Web Services for Robust Shared Business Services

After assessing the business objectives and the technical constraints, the desirable architecture was in place (see Figure 1). A strategic decision was made to outsource the IT infrastructure to a commercial-strength High Performance center. By lowering the cost of ownership and increasing availability, the client was able to harness on-demand computing using state-of-the-art equipment. With the appropriate SLA in place, incremental scalabil-

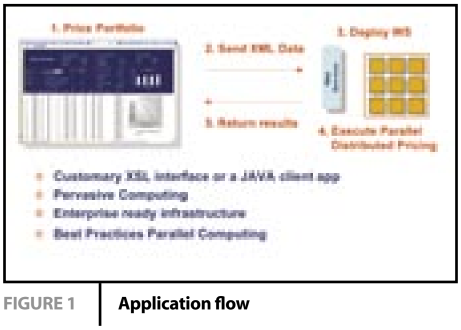

1. Price Portfolio
2. Send XML Data
3. Deploy WS
5. Return results
4. Execute Parallel Distributed Pricing
Customary XSL interface or a JAVA client app
Pervasive Computing
Enterprise ready infrastructure
Best Practices Parallel Computing

FIGURE 1 Application flow

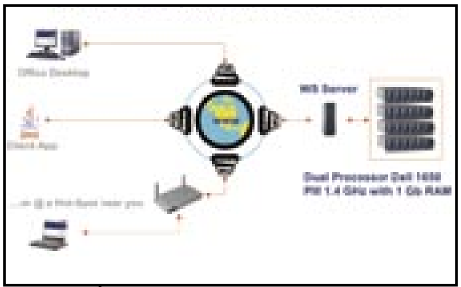

Office Desktop
Java Client App
WS Server
Dual Processor Dell 1650 PIII 1.4 GHz with 1 Gb RAM
...or @ a Hot-Spot near you

FIGURE 2 Technical infrastructure

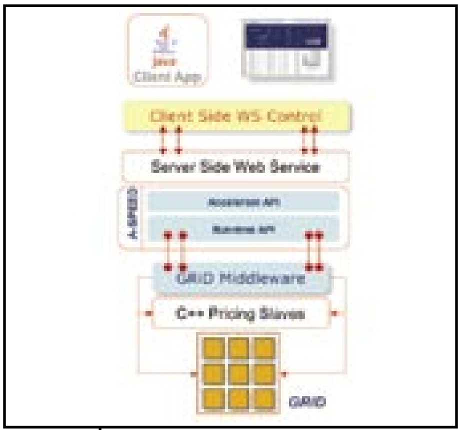
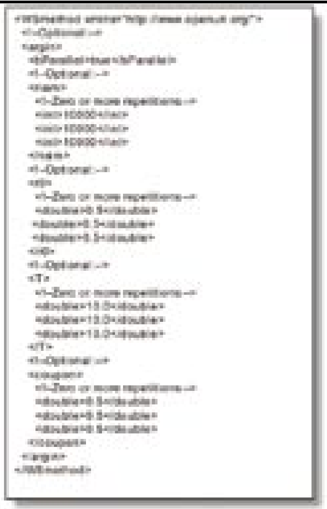


FIGURE 3 The software stack



```
<WSmethod xmlns="http://www.aspeed.org/">
  <!--Optional:-->
  <args>
   <bParallel>true</bParallel>
   <!--Optional:-->
   <nans>
    <!--Zero or more repetitions:-->
    <int>10000</int>
    <int>10000</int>
    <int>10000</int>
   </nans>
   <!--Optional:-->
   <sig>
    <!--Zero or more repetitions:-->
    <double>0.5</double>
   <double>0.5</double>
   <double>0.5</double>
   </sig>
   <!--Optional:-->
   <T>
    <!--Zero or more repetitions:-->
    <double>13.0</double>
    <double>13.0</double>
    <double>13.0</double>
   </T>
   <!--Optional:-->
   <coupon>
    <!--Zero or more repetitions:-->
    <double>0.5</double>
    <double>0.5</double>
    <double>0.5</double>
   </coupon>
  </args>
</WSmethod>
```

FIGURE 4 Sample input file

ity turned out to be predictable and affordable.

A Web service interface provided the single API to the pricing engine and facilitated the Shared Business Service for the two LOBs. Furthermore, it provided an elegant solution to the technology interoperability gap. An unforseen benefit was the ability for the salesman to execute remotely from his laptop the Web service at a third- party office or at the coffee shop nearby (see Figure 2).

The server side of the Web service is the master/control program starting the computational/slave units on grid configuration using ASPEED's ACCELLERANT on-demand application servers and HP center's middleware fabric. Each slave component encapsulates the computation aspect of the portfolio pricing, the C++ legacy code (see Figure 3).

Every time the Web service is called, a number of slave calculations are fired on the grid. ACCELLERANT's on-demand server provides quality of services such as dynamic load balancing, fail-over, and managed optimal response time.

Building the Client Web Service

BEA's 8.1 Platform technology provided the ideal environment for building the client Web service call.

1. The first step was to create a Web service control. This was achieved simply by pointing at the published Web service URL followed by ?WSDL.:

```
http://someWebServicesURL/Pricing.
asmx?wsdl
```

2. The WSDL file received was saved at a local project directory. Figure 4 shows the input definition, a segment of the WSDL file.
3. Browse to the project and directory where the saved WSDL is located.
4. Right-click the WSDL file and select Generate JCX from WSDL. The resulting JCX file is a Web service control, which can be used from the Java client trading system.

Figure 4 shows a simple <XML> input file that is sent to the Web service.

Conclusion

In this article, I demonstrated how Web Services play a critical role in two fundamental areas of distributed parallel computing: infrastructure middleware and parallelization. I then defined the three Stages of application distribution: Stage I, embarrassingly parallel and two variants of Stage II, complex interdependent; and "impossible" to parallelize. I concluded with a case study of parallelizing a Stage II application using BEA's 8.1 platform and ACCELERENT from ASPEED.

The computational grid and distributed parallel computing deliver substantial performance improvement today. While the standards are still evolving, practitioners design and implement missing-critical applications, doing more at a faster rate in diverse commercial areas, and enjoy-great competitive advantage. Financial services professionals can execute complex financial models and provide exotic products to their clients for higher profits while they meet more stringent regulatory risk requirements and improve the bottom line through more efficient capital allocation. Pharmaceutical companies speed up preclinical and early clinical trials by a factor of five or more and gain FDA approvals faster. Manufacturing uses fluid dynamics, executing on powerful compute farms and connecting designers via Web services, to deliver faster simulation and to shorten new product life cycle while delivering better, cheaper, stronger products.

Web services play a pivotal role not only in the infrastructure back-end space, but also closer to the "final mile." I predict in the next 18 to 24 months, as the product stack matures and bandwidth increases, Web services, dynamic business process choreography, and informal on-demand networks will be able to tap the idle power of powerful compute farms, or even commuters' sleepy laptops, and deliver content on pervasive devices like never before. But remember, the grid is only as profitable as the applications you run on it.

Until then, get the grids crunching.

About the Author

Labro Dimitriou is a BPMS subject matter expert and grid computing advisor. He has been in the field of distributed computing, applied mathematics, and operations research for over 20 years, and has developed commercial software for trading, engineering, and geoscience. Labro has spent the last five years designing BPM-based business solutions..

■ ■ ■ labro@bpmsinc.com

Demystifying the ESB

What's real and what's marketecture?

■ Like pretty much everything that gets caught up in the IT hype cycle, the enterprise service bus (ESB) has taken on mythic capabilities. And, like pretty much everything that becomes the focus of myth, there's a kernel of truth embedded within all those super powers.

WRITTEN BY
ROSS ALTMAN

To begin the process of separating fact from fiction, let's start with a workable definition of an ESB.

"An *enterprise service bus* is an integration platform that utilizes Web service standards to support a wide variety of communications patterns over multiple transport protocols and deliver value-added capabilities.

Now, let's pick that apart a bit so we can better understand just what an ESB can really do.

1. ***An ESB is an integration platform...***

While an ESB can be used to build a stand-alone application "from scratch," it's not ideal for that purpose. The main competition here is the various component-based application runtime platforms such as the J2EE application servers and the Microsoft .NET framework. These application runtime platforms are designed specifically to support development of new, modular applications where all those components run within the same environment. Also, and perhaps more tellingly, J2EE and .NET applications tend to be more tightly coupled – each component is written with specific knowledge of the data and process semantics of the other components in the application.

In contrast to that, ESBs are designed to support the development of composite applications – applications that involve some new code "on top of" lots of existing code from multiple previously built systems. Unlike the more tightly coupled architecture of the typical J2EE or .NET application, solutions built on an ESB are best implemented within a loosely coupled architecture.

2. ...that utilizes web services...

The "service" in enterprise service bus refers to the architectural concept in which a component offers to execute functions on behalf of any other program that calls the component's published API. There is a presumption in this pattern that the called program may change its internal processing as long as it honors the API – i.e., as long as the API doesn't change and as long as the visible results obtained through invocation of that API don't change.

Given that usage, it's clear that, at least conceptually, an ESB could use any protocol as its normative API. However, as a practical matter the term ESB has become inextricably intertwined with Web services. As a result, in the real world of product marketing and vendor-delivered functionality, an ESB uses Web services to deliver its component-to-component connectivity.

3. ...to support a variety of communications patterns...

Saying that an ESB supports Web services protocols doesn't limit the communications patterns that can be delivered by an ESB. While the predominant Web services–based application pattern is built around synchronous request/reply interaction between service providers and service consumers, an ESB can and must support a much broader variety of one-way and two-way, synchronous and asynchronous program invocation models.

4. ***...over multiple transport protocols...***

At first glance, it might look like we have mutually exclusive requirements here. Many would immediately assume that an ESB that "utilizes Web services" couldn't also support communications "over multiple transport protocols." However, a little digging into the most fundamental Web services standards shows that Web services can be implemented on top of a variety of transport protocols. The most common implementation by far is SOAP over HTTP. But, SOAP can also be implemented over SMTP or MQSeries or JMS or...well, you get the picture.

5. ...and deliver value-added capabilities.

We're pretty far along in the process of building a complete array of standards to support enterprise services; however, we're still not very close to the pot of gold at the end of that rainbow. In short, it will be many years before we have a complete set of mature and widely implemented Web services standards that enable robust, secure, performant, and highly available communications. In the meantime, developers will look to the ESB vendors for the value-added functionality that the available standards haven't yet addressed.

Now that we've defined an ESB, let's clarify its use. While an ESB can certainly be used to create a brand new, stand-alone, component-based application, an application server is probably better suited for that sort of project. But for ESB fans that's okay, as there are relatively few brand new, stand-alone, component-based applications in the development queue in most enterprises.

On the contrary, most applications that are being built today are composite applications, consisting of a combination of a relatively small amount of new functionality along with a large amount of existing application logic and data. And, that design target is the sweet spot for ESB use.

So, what's required to build and deploy an ESB? Well, it all starts with the communications infrastructure that we dealt with in our definition – an integration platform that utiliz-

The Gilbane Conference

ON CONTENT MANAGEMENT TECHNOLOGIES

April 11-13, 2005
San Francisco, California
The Palace Hotel

Lighthouse Seminars

Content technologies have become mainstream and need to be part of all major enterprise applications and integrated into IT architectures and infrastructures. Join us to learn what you need to know to formulate a successful strategy from our gathering of content technology experts and practitioners.

What you will take away from the conference:

Attendees benefit from an unbiased, deep, and up-to-date understanding of content management technologies, vendors, trends, and best practices, from the most experienced and respected experts in the field. Our speakers have implemented every kind of content management system across all industries, and have written the books and reports that others depend on. We are strictly neutral with regard to vendors, analysts, enterprises, and consultants. We ensure the focus is on what you need to know to successfully plan and implement content technology solutions.

"The Gilbane Conference has provided me with crucial information that I was looking for as my company rolls out its content management solution. I have been able to network with industry experts, the major content management technology players and other companies, like mine, that are faced with the challenge of effectively managing content," says Angie Khumdee, Senior Software Engineer, Motorola CGISS E-Business

Register NOW!

Register now for the **CONFERENCE PLUS PACKAGE** and receive a **FREE iPOD! LIMITED QUANTITY, REGISTER EARLY!**

GET STARTED to make vendor choices, benchmark your progress, and make strategic decisions.

LEARN from industry leaders and anaylst about successful implementation of content technology through the unique combination of:

- **Project Management**
- **New Technology Trends**
- **Case Studies**

For more detailed information, please visit: **www.lighthouseseminars.com** or call **Joe Richard** at **781.821.6734**

Platinum Sponsor

INTERWOVEN

Media Sponsors

Business Wire · cms WATCH · EContent · The Gilbane Report · InfoWorld · WebServices · XML Journal es Web services to support a variety of communications patterns over multiple transport protocols while delivering value-added capabilities.

In reality, the ESB products in the market provide this to varying degrees. Some only support basic communications capabilities – enabling simple request/reply interaction with basic connection-level security. Other products enable both synchronous request/reply as well as asynchronous send/receive and some also add message-level security along with transaction management, logging, and even billing capabilities.

But, regardless of the communications functionality provided by the ESB, that's far short of meeting the complete set of requirements for developing and delivering a composite application. At a minimum, you need three layers of technology to address your integration needs, and the communications capabilities that we've just described only provide the middle layer in that model.

The bottom layer in our three-tier model provides the ability to link ESB communications to the wide variety of applications and data that need to be leveraged in the typical composite application. This layer includes tools or components that are sometimes referred to as adapters or wrappers, but that describes only part of what's needed here. You start with the ability to wrap or adapt a legacy program or data API to present a Web services interface, but you also need to be able to readily create and maintain the data transformation logic that's required to merge data from multiple applications into one new application.

You also have to be able to create APIs where none exist today – i.e., you need to be able to screen scrape those legacy programs that were never designed to be called by another program.

In the end, the functions needed at this bottom layer of our model can be met by products that have been traditionally referred to as EAI tools or integration brokers. There are also vendors who specialize in providing full function adapters (i.e., adapters that include transformation capabilities) that can be used with a range of integration tools from other companies. Regardless of where they come from, you can't really build an ESB-based application without these adapters, as there just aren't many legacy and packaged applications in the typical enterprise that natively supported Web services interfaces.

In addition to this adapter layer, we need a layer in our model that sits on top of the ESB's communications functions. Once data flows from your legacy and packaged applications through adapters and transformation logic and into your ESB, you've created a basic service-oriented architecture. But, that's a bit like having a Ferrari on cinder blocks. To take your SOA on the road, you need an application layer, including such capabilities as Business Process Management; Business Activity Monitoring; workflow; B2B connectivity; graphical user interface functionality; portal-based authentication, authorization and personalization; and master data management. In short, you need a comprehensive application development toolkit on top of your ESB and adapters/transformation tools.

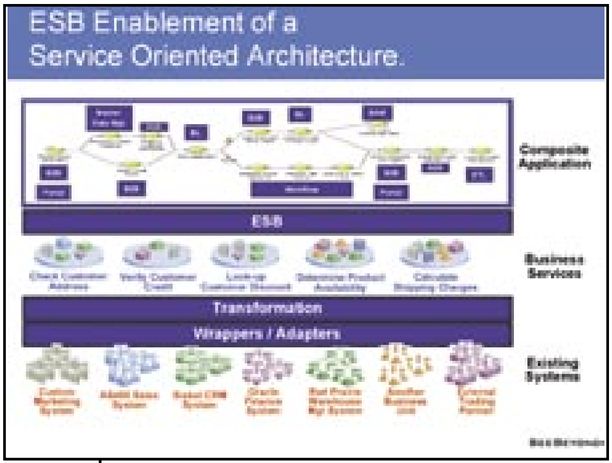


FIGURE 1 An ESB-based service-oriented architecture

Now, you could get an ESB from one vendor (i.e., an "ESB-only" vendor) and these development tools from various other vendors. Or, you can get all this stuff from a single vendor (i.e., an "ESB ecosystem" vendor). Which is best for you?

The ESB ecosystem vendors have several short- and long-term advantages over the ESB-only vendors.

1. You can buy everything from one vendor, which generally results in a shorter and simpler product evaluation and acquisition process, lower license and maintenance costs, and "one throat to choke" when issues come up around support, professional services, and customer care.
2. Basic ESB functionality is going to become commoditized, which will endanger the long-term viability of ESB-only vendors, who will find it increasingly difficult to differentiate their products.
3. As the Web services standards mature, both the perceived and the real value of an ESB will erode.

These advantages for the ESB ecosystem vendors call into question the long-term viability of the ESB-only vendors. They will either have to become ESB ecosystem vendors or sell out to one.

By the way, there are also issues that have to be evaluated when buying a solution from an ESB ecosystem vendor. If the vendor has simply pulled together the components of the ecosystem through the "late binding" of separately developed development tools or (more commonly) through merger and acquisition activity, then there will be no synergy in the product suite except, perhaps, the "one throat to choke" benefits previously noted.

On the other hand, if the various components and tools of the ESB ecosystem are actually integrated, then there are further benefits that can be realized. For example, a fully integrated set of ESB-based tools could share a single repository for development and a single runtime environment for deployment. This would allow developers to better implement reuse strategies by making it easier to find previously built components. Also, it would be easier to deploy and manage the solution at runtime if there were a single runtime console, a single debugger, a single runtime management environment, a single security framework, and a single transaction management engine.

Given all the hype that has surrounded the ESB marketplace, it's understandable that some of these issues have been lost in the turmoil. But, in the end, myths have nothing to do with getting your applications into the hands of users. It's the performance of your tools and platform software that's critical.

About the Author

Ross Altman is the CTO of SeeBeyond, where he focuses on the direction, development, and communication of the company's technology vision and strategy. Before joining SeeBeyond, Ross was vice president and research director at Gartner, where his research focused on application integration middleware.

Identity-Based Service-Oriented Architecture

Regulating business needs with Web services

■ Service-oriented architectures (SOA) have become the de facto architecture of choice for enabling agile business processes via reuseable, coarse-grained business services. Business services are integrated via exposed technical interfaces that increasingly support Web services and XML standards. By adopting the loosely coupled business service model, IT is better able to support the evolving business needs of customers, partners, and employees, who access these services over multiple channels such as browsers, devices, or other applications.

WRITTEN BY
SEKHAR SARUKKAI

In order to truly realize the promise of reuseable services, appropriate controls have to be put in place for protecting and ensuring service availability. Interestingly, while technologists are focused on XML intruders intercepting Web service messages, business people, who take this for granted, are focusing on tracking and controlling user and application accesses of these business services. These might otherwise expose businesses to significant risk, by exposing sensitive data to the wrong user, for example.

From a technical perspective, addressing this business need for protecting SOA assets is a complex cross-domain problem compounded by the loosely coupled nature of SOA. The solution to this problem necessitates the deployment of a new generation of SOA infrastructure domain services responsible for authoritative establishment and verification of identity. From a business perspective, this is critical to realizing the benefits of quicker application development and deployment, as well as increased system governance. These services should handle application, service, device, and other entity identities, just as user identities were managed in previous generation Web application deployments. It is also important that this service plug into an SOA-compliant federated architecture.

Today's Solutions Don't Deliver Intended Value

An SOA service is all about open protocols and hidden implementation, while today's identity and access solutions are about exposed implementation (in the form of proprietary APIs) and hidden protocols, which usually do not rely on the standards that the SOA backbone relies on.

Solutions deployed today largely depend on proprietary or tightly coupled integrations to identity and access servers that support single sign-on (SSO) functionality and are integrated with portals or Web servers (see Figure 1). While this deployment does not address the issues around all participants in an SOA, this also breaks one of the fundamental premises of a loosely-coupled SOA by tightly coupling SSO solutions to applications that are protected. More importantly, not only is this non-compliant with standards, it also eradicates the value and corresponding technical and business benefits of adoption of an SOA.

In order to break this tight coupling, you need to rethink how Identity and Access services can be deployed as shared infrastructure services. Such an Identity service easily plugs into the adopted SOA service bus or messaging standards (such as XML or SOAP over JMS, HTTP, etc.); it is discoverable and addressable, just as is any other service in the enterprise.

Who Knows "Who is Who" in an SOA?

The establishment of the SOA-enabled Identity authority addresses the fundamental question of "who knows who is who" in the distributed network of applications, users, devices, and services. This authority acts as an authoritative SOA "system of record" that is not only an authority on all legitimate participants in the application network, but that can also manage and validate credentials associated with each of the participants in the SOA. This is the single logical authority that validates "who is who" in the SOA.

The legitimacy of each entity participating in an SOA is established by a Web service–enabled Domain Identity Service that supports authentication and authorization of services and users, while also acting as an attribute authority for all these entities. Existing SSO (Access) and IdM (Identity Management) solutions, put in place for securing Web applications, need to be enhanced with the following two SOA-induced enhancements:

1. Web service enablement of Identity in the form of a Domain Identity Service (DIS)
2. Identity enablement of Web services that seamlessly integrates with the DIS

Deployed in a standards-compliant manner, the shared domain-specific Identity Service becomes a ubiquitously available application-level Identity service analogous to DNS for Naming in the lower levels of the networking stack.

Web Service Enablement of Identity (DIS)

While establishing the security blueprint for SOAs, architects should explicitly address the following concerns:

1. How can IT directly address the business need of defining, tracking, and enforcing access to cross-domain business services?
2. How can IT help reduce business operational risk by eliminating vendor lock-ins via proprietary protocols in the enterprise backbone (especially for critical Identity-related services)?
3. How can IT architect better control, availability, and visibility into service-level access-related events that can be achieved only with standardization of interfaces to critical infrastructure services such as Identity and Access?

Business Benefits of Service Enabling Identity

1. A greater level of business-process agility is achieved by providing a mechanism to more quickly and securely monetize new business services and processes without tight couplings
2. Operational risk is reduced due to the elimination of vendor-specific protocol in the integration backbone
3. Governance is improved due to better auditability of authentication and authorization of service interactions

IT Benefits of Service Enabling Identity

1. Provides the ability to leverage existing WSSO investment by simply SOA enabling Identity without mandating SAML
2. Allows for the coexistence of existing deployments (for existing applications) while providing a means to deploy and support a productized DIS
3. Offers standards compliance with out-of-the-box support for SAML and WS-Security
4. Allows for the easy customization of DIS to enterprise XML/SOAP standards without custom coding

Web service enablement of Identity and its resultant deployment as Domain Identity Services is an approach that assess the above concerns.

DIS can have a significant impact on how securely and effectively applications, people, and devices can interact in an SOA by eliminating the otherwise proprietary protocol exceptions necessary to leverage the IdM and Access solutions. A loose, yet similar, analogy exists in the traditional world of systems networking, wherein an ubiquitous DNS service that is an authoritative naming authority significantly enhanced the value of the networked machines by simplifying addressability (as names rather than numbers), providing a scalable cross-domain architecture for addressing and increasing the value of networking these systems.

Similarly, in the DIS model, enterprises should view identity as a service that is ubiquitously available (as a shared infrastructure service necessary for application networking), rather than as being managed by a server (such as an Authentication or Access server). However, unlike a simple DNS-like naming authority, DIS acts at the application (or service) level and takes on a more complex role of not only being an attribute authority, but also supporting authoritative validation of named entities participating in an SOA, federation across domains, and token exchange.

DIS supports the following core interfaces:

- A SOAP/XML registration interface to manage entities (at a minimum, Web services and users) that are to participate in the SOA
- A SOAP/XML interface for querying attributes associated with entities
- A SOAP/XML interface for authentication and access queries

Plus:

- It should be deployable in a federated environment with support for SAML and WS-Federation
- It provides extensibility to support Security Token Service functionality (such as support for WS-Trust)

Each of the core interfaces of the DIS is available as services, via individually

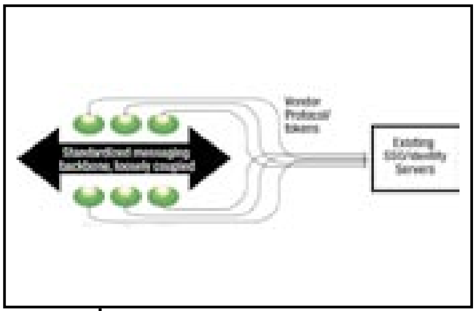

Standardized messaging backbone, loosely coupled
Vendor Protocol tokens
Existing SSO/Identity Servers

FIGURE 1 Traditional model

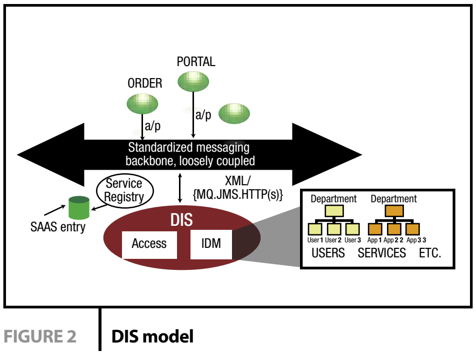


FIGURE 2 DIS model

addressable Web services in a standards-compliant manner. These services should be part of the SOA infrastructure, having been established by enterprise architects, and should be securely reusable by any application or service connecting to the same domain. Cross-domain security is transparently handled by the local DIS through communication between multiple DISs via standards such as WS-Federation.

DIS Functionality

In order to support all of the requirements listed above, the DIS should support both an extensible data model that allows for the management of various types of identities beyond the identity of people, and a well-defined functional interface via a documented protocol utilization, which can be queried for decisions or actions on behalf of an identity.

Data Model

A DIS manages various entity types (such as users, services, applications, devices, etc.) that are all associated with the metadata required for validation of a claim of identity, the types of tokens associated with the identity, and the arbitrary metadata that is specific to the type of entity under management. In addition, a set of entities (such as a list of unique users, applications, devices, etc.), each of which belongs to one of the entity types, is stored and managed by the Identity Authority Service.

Functional Interface

The functional interface to this service exposes four types of operations:

1. Administration
2. Issuance
3. Validation
4. Exchange

Administration: Allows administrators to provision new Identity types and Identities supporting the complete life cycle, including all the customary CRUD (Create, Read, Update, Delete) functionality, in a delegated and federated manner (see Figure 3).

Issuance: Given a claim, the DIS service returns a token in some configurable form that can later be presented by the holder of the claim to gain access to services in the SOA network (see Figure 4).

Validation: Given a token, the DIS returns a decision on whether the owner of the token is allowed to perform a function in the SOA network – including continued access to the application, or authorization to specific operations in the network (see Figure 5).

Exchange: Given a token, the DIS returns a token of an alternate form, enabling the support for heterogeneity in token formats within and across SOA networks (see Figure 6).

Standards such as WS-Trust and SAML provide the ideal interfaces to expose DIS service functionality for the issuance, validation, and exchange of tokens. Standards around the administration of different identities do not exist, but can be exposed via customizable SOAP interfaces.

Identity Enablement of Web Services

Once the DIS is put in place, the deployment architecture should support a declarative mechanism to enforce the use of the DIS services without developer involvement. In this model, all inbound messages to a Web service are automatically intercepted and authentication and authorization rules are enforced by leveraging the DIS via standard protocols. This level of seamless use of Identity services constitutes Identity enablement of Web services.

As enterprises roll out Web services, they are often challenged by reasoning about the identities of applications or other Web services that are actually invoking the service via the published service interface; this is unlike the previous Web situation in which end users were typically involved in presenting credentials via browser forms in order to authenticate and get access to a URL. In the case of applications invoking services, the following issues need resolution:

1. How does a consuming application securely bind itself to its credentials?
2. How does this application communicate its credential for authentication to the Web service being consumed?
3. How does the recipient service validate the credentials and access rights of the application trying to invoke an operation on the service?
4. How do you support cross-domain Web service interactions?

Table 1 shows a simplified comparison between the process of protecting Web services in an SOA and the process that comprises traditional SSO solutions commonly deployed for protecting Web sites.

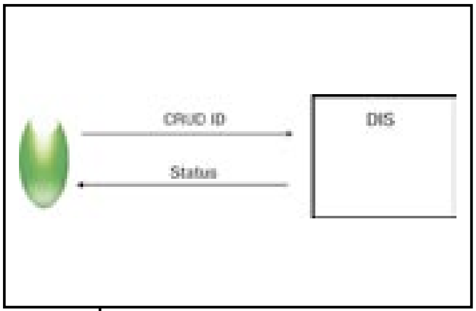


FIGURE 3 Administration

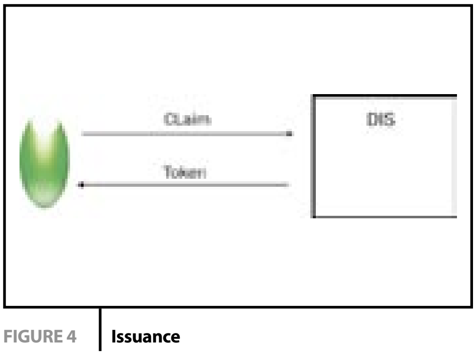


FIGURE 4 Issuance

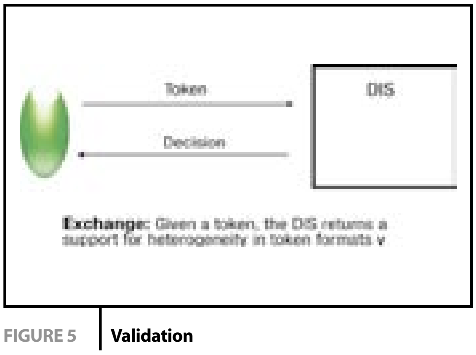

Token
Decision
DIS
Exchange: Given a token, the DIS returns a
support for heterogeneity in token formats v

FIGURE 5 Validation

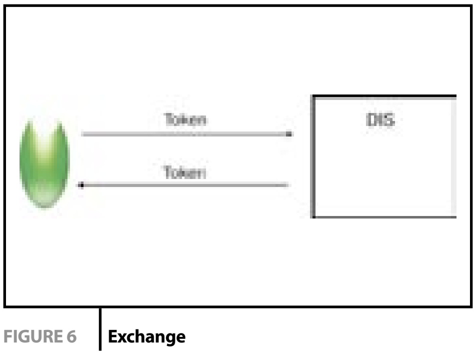
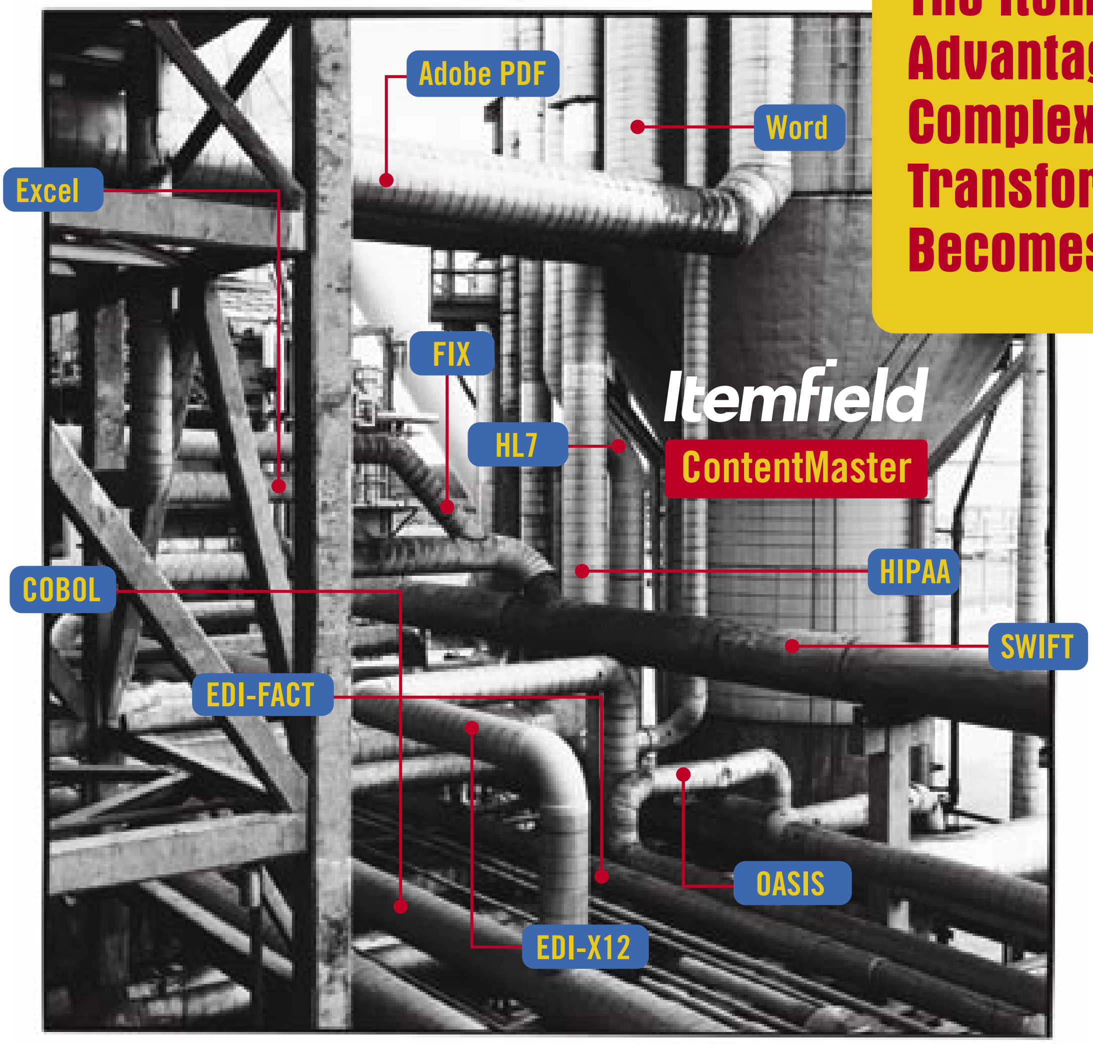


FIGURE 6 Exchange

There are additional considerations that need to be incorporated into the solution. In many practical deployment situations, services do not always authenticate and authorize other services only. Instead, they may need to authenticate or authorize the end user, consuming application, or a combination of both.

As shown in Figure 7, Identity-enabled

The Itemfield Advantage Is That Complex Data Transformation Becomes Simple



Achieve the "Complexity Advantage" in Business Integration

Find Out What a Growing Number of Global Fortune 500 Companies Already Know

Complex data is at best a significant cost factor in business integration—at worst it can cripple a business integration initiative. For many enterprises, integrating complex data remains an error prone and costly process that requires IT development.

ContentMaster™ is next-generation data transformation software that dramatically simplifies application development and integration—reducing implementation costs by as much as 80%. ContentMaster allows any form of enterprise data to be transformed to any other form in real-time via XML—without the need for costly IT development.

ContentMaster includes a powerful visual integrated design environment and embeddable runtime engine that is available standalone for deployment into virtually any software architecture, or as integration agents for deployment to IBM® WebSphere® Business Integration, Microsoft® BizTalk and webMethods®.

Visit our website **www.itemfield.com/itsolutions** to obtain a copy of our latest white paper *"The Complexity Advantage: Solving the Requirements of Complex Data Transformation in Business Integration"* and to view the ContentMaster demo.

Itemfield | 1 800 838 1055 | www.itemfield.com/itsolutions

Web Sites	Web Services/SOA
Entities (people) bind to a protected private secret (in the form of a password) and to a public identity (in the form of a username or e-mail address, etc.)	All SOA components (services) are associated with a unique Identity that is also registered in the DIS. In the absence of a DIS, a "ghost" user that represents the component is created in the same IdM solution used for managing people identities. The private and public information associated with the application component is communicated to the administrator of the service, who is responsible for making it accessible to the service – in many situations doing so by further protecting the application credentials with administrator credentials (such as password protection or encryption of the application credentials)
Credentials are typically communicated as username/passwords, propagated as http headers over a secure transport (such as SSL)	Credentials are inserted by a client-side agent or by the consuming application into HTTP headers or preferably as tokens in the WS-Security headers (or occasionally custom SOAP header variables), such as username token, binary tokens (for X509 certs or proprietary SSO tokens), or SAML assertions, etc.
A Web server plug-in extracts the credentials from the header and invokes an Access server that validates the user credentials, and on success returns an SSO token for future use	Credentials or claims are validated by a service-side policy enforcer (such as Gateway or an agent) that extracts the credentials/claims from configurable header variables and queries the DIS for an authoritative response for authentication or authorization
Integrate with an SAML service supporting browser and artifact profiles. Full-fledged federation via WS-Federation	Integrate with an SAML service supporting WS-Security profile. Full-fledged federation via WS-Federation

Table 1 **Protecting Web Services in an SOA versus traditional SSO solutions**

Web services can handle different types of subject credentials, depending on trust relationships. The three primary types of credentials used to validate access to a service are:

1. End-user credentials that are required for authentication of end users at the Web service. Independent of the path through which the user transaction indirectly reaches the service, the policy enforcers extract the credentials from configurable locations associated with the context of the message, and authenticate and authorize service access using the DIS.
2. If the service has a trust relationship with another application (such as a portal or a partner application) established via mutual trust based on X509 certificates, the service needs to only authenticate the application accessing the service, rather than the end user. End-user identity may be passed along in the message context (via SAML assertions, SSO cookies, etc.) for authorization purposes, but there is no need for user credentials to be propagated beyond the application that the user directly accesses.
3. In certain situations, the service may require both the consuming application and user credentials, and in this scenario, the policy enforcer is configured to both look for credentials and perform appropriate access controls.

In all these cases, if an SAML token or an SSO cookie is not already present, the policy enforcer may optionally insert such a token for future downstream use of the token for reduced sign-on needs. In certain other situations, fine-grained authorization needs may require communication with the DIS for querying subject attributes for further rule processing via pluggable entitlement engines.

By centrally managing both user and application identities in the DIS, the policy enforcers do not have to worry about separate infrastructures for allowing access based on user or application Identities, thus simplifying deployment and easing auditability of access to services.

Summary

Enterprises should view identity-enabling Web service and Web service–enabling identity as two sides of the same coin that forms the cornerstone on which to develop a strong security foundation for an SOA. By viewing Identity across the domain as a service that is ubiquitously available, enterprises can better handle the management of proliferating services in an SOA while meeting governance requirements and setting the right scalable architecture in place for the long-term reliance on identity as a core business asset.

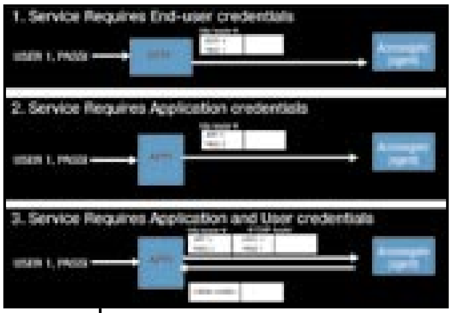

1. Service Requires End-user credentials
USER 1, PASS
2. Service Requires Application credentials
USER 1, PASS
3. Service Requires Application and User credentials
USER 1, PASS

FIGURE 7 **Identity-enabled Web services**

About the Author

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

■ ■ ■ ssarukkai@oblix.com

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

International Web Services Conference & Expo

web services conference&expo EDGE

Keynotes

Matt Ackley
eBay

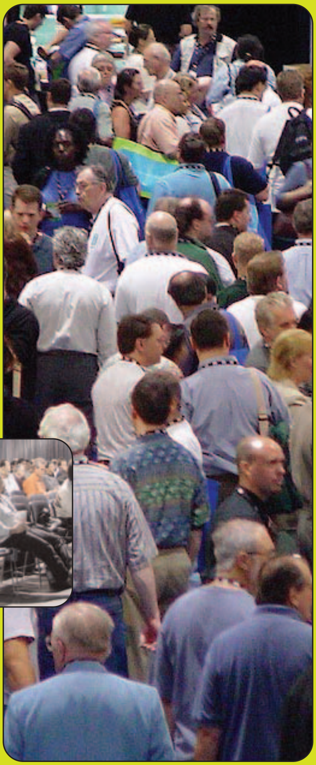
Ari Bixhorn
Microsoft

Mike Milinkovich
Eclipse.org

Register by FEBRUARY 11, 2005 SAVE Up To $200

WEB SERVICES EDGE 2005 EAST

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



New for 2005…

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

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

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

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

PRODUCED BY: SYS-CON MEDIA
PRESENTED BY: SYS-CON EVENTS

DIAMOND SPONSOR: Microsoft

GOLD SPONSORS: Novell. SAP

ASSOCIATION SPONSORS: INTEGRATION CONSORTIUM OASIS ineta eclipse

CORPORATE SPONSOR: Java Community Process

APP.-SERVER SHOOTOUT FACILITATOR: burton GROUP

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

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

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

International Web Servic

web services conference & expo EDGE

Web Services Edge 2005 East

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

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

Who Should Attend

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

Features & Attractions

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

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

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

ices Conference & Expo

Keynote Speakers

Tuesday, February 15, 11 a.m.

Matt Ackley
Senior Director, eBay
Developers Program

Web Services for eCommerce

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

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

Wednesday, February 16, 11 a.m.

Ari Bixhorn
Director, Web Services Strategies,
Microsoft Corporation

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

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

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

Thursday, February 17, 11 a.m.

Mike Milinkovich
Eclipse.org

An Open Development Platform for Web Services

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

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

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

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

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

SPECIAL INSERT: WEB SERVICES EDGE 2005

WWW.SYS-CON.COM/EDGE

FEBRUARY 15-17, 2005

JOHN B. HYNES CONVENTION CENTER • BOSTON, MA

DAY 1 FEBRUARY 15

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

DAY 2 FEBRUARY 16

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

DAY 3 FEBRUARY 17

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

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

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

Novell.

FREE Web Services Security Tutorial Presented by Novell

Using a Web Services Framework to Build SOA Applications

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

FREE Tutorial Presented by Java Community Process

Developing Web Services Using Java Technologies

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

Java Community Process

Free Tutorials with VIP Preregistration ONLY!

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

Microsoft®

FREE .NET Tutorial Presented by Microsoft

The Next Generation of Visual Studio
Thom Robbins

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

Free Tutorials with VIP Preregistration ONLY!

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

critical sites™

FREE .NET Tutorial Presented by Critical Sites

Security, The New Reality
Patrick Hynds

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

Free Tutorials with VIP Preregistration ONLY!

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

International Web Servic

WEB SERVICES TRACK

WS-1 Ensuring Web Services Interoperability

Chris Ferris, IBM

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

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

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

WS-2 Web Services Standards: Going Behind the Mask

Glen Daniels, Sonic Software

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

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

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

WS-2B Solving Complex Business Problems Though SOA

John Daly, netNumina

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

BIO: John Daly is a recognized leader in service oriented architecture, enterprise wide integration and business intelligence. As Managing Director of netNumina, a leading firm focused on financial service, pharmaceutical and other fortune 100 companies, he is responsible for helping companies solve complex business problems by delivering sophisticated technology solutions that combine proven technologies with cutting edge thinking. John's ability to deliver technological solutions specifically designed for the user experience separates his approach from those of netNumina's competitors.

WS-3 The XML Data Challenge

Jonathan Bruce, DataDirect Technologies

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

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

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

WS-4 How To Bulletproof Your Web Services

David McCaws, Parasoft

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

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

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

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

Toufic Boubez, Layer 7 Technologies

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

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

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

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

Alistair Farquharson, Digital Evolution

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

One of the great benefits of a service-oriented architecture is the ability it gives you to extend programmatic, integration capabilities to business partners. Going beyond simple sharing of data with partners, SOA enables true B2B application integration. At the same time, this capability creates a vexing security policy enforcement dilemma. How can you be sure that a user from a partner organization is actually authorized to integrate with your applications? How can you authenticate that user? Do you even want that headache in the first place? This session will discuss the issues that arise in B2B security policy enforcement and explore several proven approaches

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

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

to solving the problem. In particular, it will focus on the emerging technology of XML Virtual Private Networks (XML-VPNs) and their potential to mitigate security policy enforcement issues in B2B SOA implementations.

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

WS-7 Driving SOA Governance

Brent Carlson, LogicLibrary

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

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

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

WS-8 SOA: From Pattern to Production

David Chappell, Sonic Software

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

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

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

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

Sameer Tyagi, Sun Microsystems

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

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

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

WS-9B Effective Risk Abatement and Success in a Service-Oriented World

Paul Lipton, Computer Associates

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

IT leaders are hoping to leverage the benefits inherent in Web services and Service-Oriented Architectures (SOA) to enable their businesses to be far more competitive and to find new operational efficiencies. But, can we depend on these new technologies and approaches? Management and security are a common concern today and this session provides the necessary background and perspective on both the business and the technical issues. We will examine important principles and recommendations using real-world examples to illustrate key concepts.

BIO: Paul Lipton is a senior architect in the Web services and application management team at Computer Associates (CA) as well as a Strategist in the Office of the CTO. He has been an architect and developer of enterprise systems for over 20 years, and has worked closely with key CA customers to solve important business challenges through the creation of manageable, mission-critical distributed solutions.

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

Eric Newcomer, IONA

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

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

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

WS-11 Four Abilities SOA Will Lack Without a Registry

Luc Clement, Systinet

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

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

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

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

WEB SERVICES SECURITY TRACK

WSS-1 Identity in SOA

Sekhar Sarukkai, Oblix

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

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

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

Attend a FREE* Tutorial

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

Using a Web Services Framework to Build SOA Applications

Ashish Larivee

Ashish Larivee, Novell

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

Novell.

* Free Tutorials with VIP Preregistration ONLY!

WSS-2 Securing Web Services with WS-Security

Dr. Jothy Rosenberg, Service Integrity

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

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

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

WSS-3 Anatomy of a Web Services Attack

Mamoon Yunus, Forum Systems

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

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

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

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

Dr. Michael Juntao Yuan, University of Texas

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

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

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

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

Ashish Larivee, Novell

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

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

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

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

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

WSS-6 XML Content Attacks

Girish Juneja, Sarvega, Inc.

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

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

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

WSS-7 The Interoperability Challenge of Web Services Security Standards

Eve Maler, Sun Microsystems

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

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

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

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

Dan Foody, Actional

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

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

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

RESERVE YOUR HOTEL ROOM NOW! AT THE SHERATON BOSTON HOTEL

THE OFFICIAL CONFERENCE HOTEL OF EDGE 2005 EAST

THE SHERATON BOSTON HOTEL
PRUDENTIAL CENTER
39 DALTON STREET
BOSTON, MA 02199
PHONE (617) 236-2000
FAX (617) 236-1702

Hotel Arrangements Are Easier Than Ever! Web Services Edge 2005 East has secured special group rates, well below regular rates, at one of the finest hotels in Copley Square, The Sheraton Boston Hotel. The hotel is directly next door to the Hynes Convention Center, and is connected via The Prudential Mall. To learn more about hotel savings, or to make your individual or group reservations please contact Meeting & Travel Consultants:
Toll free (866) 549-3985
Direct (732) 549-3985
Fax reservations to (732) 549-3987.
Make online reservations by Jan. 17, 2005
visit: http://www.mtcreservations.com

SPECIAL CONFERENCE RATES

Single	Double
$196	$196

Credit card information is required to guarantee reservations and expedite confirmation. Confirmations will be mailed directly from the hotel, time permitting. All changes and cancellations should be made directly through MTI.

DRIVING DIRECTIONS TO THE SHERATON BOSTON FROM LOGAN INTERNATIONAL AIRPORT

• Follow direction sign from terminals to Sumner Tunnel/Rt. 93 North • Take the Rt. 93 North ramp as you exit the Sumner Tunnel • Exit at Storrow Drive (26 B) • Follow Storrow Drive approximately 2 miles to the Fenway/Kenmore Exit (1st exit after Massachusetts Ave., on left) • Stay left as exiting Storrow Drive going towards Fenway • Continue to 1st set of lights, staying left, going onto Boylston Street • Go through 4 sets of lights on Boylston Street • The Hynes Convention Center's main entrance driveway will be on the right immediately after going through the 4th set of lights.

WEB SERVICES EDGE CONFERENCE PAST SPONSORS & EXHIBITORS

Java and Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc., in the United States and other countries. SYS-CON Publications, Inc., is independent of Sun Microsystems, Inc. All brand and product names used on these pages are trade names, service marks or trademarks of their respective companies.

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

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

International Web Services Conference & Expo

Register by FEBRUARY 11, 2005 SAVE Up To $200

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

CONFERENCE & EXPO: February 15 – 17, 2005

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

THREE WAYS TO REGISTER FOR CONFERENCE

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

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

Please complete sections 1, 2, 3 and 4

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

First Name ______________________ Last Name ______________________

Title ______________________

Company ______________________

Street ______________________

Mail Stop ______________________

City ______________________

State ______________________ Zip ______________ Country ______________

Phone ______________________

Fax ______________________ E-Mail ______________________

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

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

Check #______________________ Amount of Check $ ______________

Charge my □ Visa □ MasterCard □ American Express □ Discover

Name on card______________________

Card #______________________ Exp. Date______________

Signature______________________

Billing Address (if different from mailing address)

____

3 PLEASE INDICATE YOUR CONFERENCE CHOICE:

Total Registration fee $____________

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

SPECIAL GROUP RATES

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

4

A. Your Job Title

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

B. Business/Industry

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

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

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

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

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

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

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

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

01 □ Yes 02 □ No

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

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

____

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

SYS-CON EVENTS

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

CANCELLATIONS, SUBSTITUTIONS, REFUNDS

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

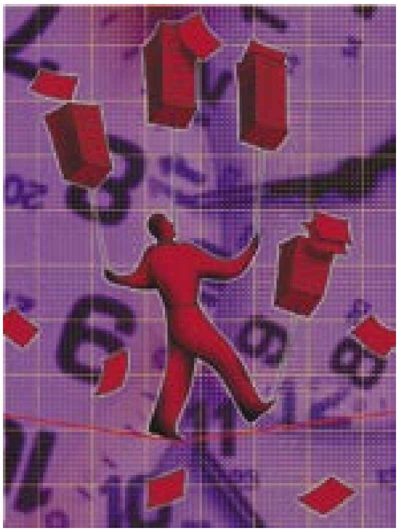
Speakers, sessions and schedule are subject to change without prior notice.

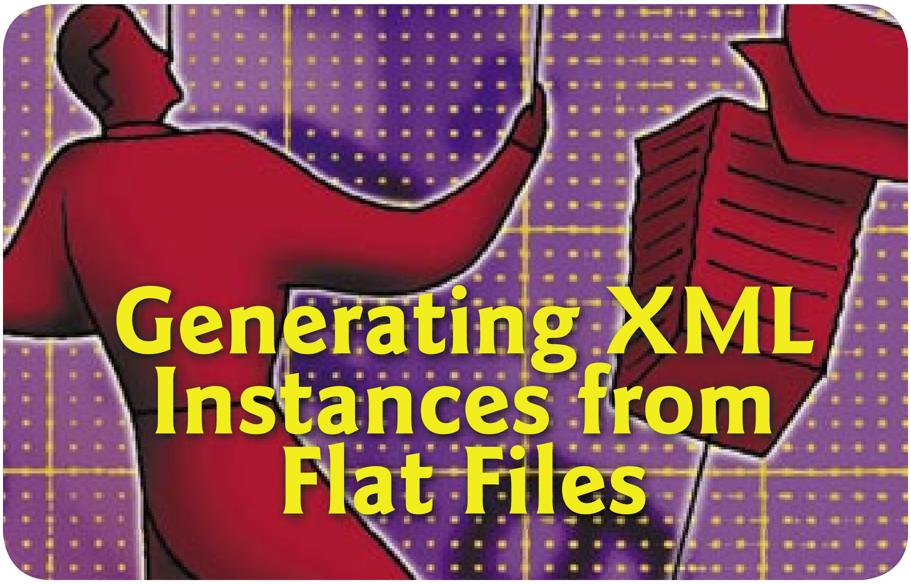
No solicitation by anyone other than official exhibitors, sponsors or marketing partners is permitted. Such behavior is cause for expulsion without refund.

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

XML JOURNAL
SUPPLEMENT

This Month





XML-Based Interop, Close up

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

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

Content Management:
Organization, dissemination, and presentation of information

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

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

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

Generating XML Instances from Flat Files

WRITTEN BY
Indroniel Deb Roy

A schema-based approach

Enterprise applications such as banking, healthcare, and so on still use flat files to import/export data between applications. Flat files contain machine-readable data that is typically encoded in printable characters. There is a growing need for these applications to interact with XML-aware applications and Web services, and to satisfy this need these applications must convert flat file data to an XML format.

XML is suited for the interchange of data as XML documents are tagged, easily parsed, and can represent complex data structure. The conversion of a flat file to an XML format requires proper representation of the data embedded in the flat file in some template form so that it can be converted to XML. There are custom solutions based on XML templates and XML DTDs made to capture the data structure of flat files to be converted to an XML format, but here a new schema-based approach to parse flat files to an XML instance will be discussed.

Why XML Schema?

The XML Schema provides a means to define the structure, content, and semantics of the data contained in XML documents. Flat files contain data, and in order to convert the data to XML, the underlying data structure and data validation rules should be captured in an XML Schema representation. XML Schema is now a well-known standard and there are various XML parsers available to parse and validate documents against an XML Schema. Moreover, it provides the flexibility of describing the data more effectively using the rich W3C schema language, which can be used to validate the generated XML document.

Several kinds of commercial software are available that convert flat files to XML instances based on proprietary templates and conversion routines. These solutions are tailored to meet specific needs and do not scale to fit the requirements of generic flat-file-to-XML-instance generation.

This approach is based on open standards such as W3C schema, API, and XERCES XML parser's schema implementation. It is suitable for any Java project or custom XML instance-generation project using open source technologies.

Process

XML Schema is the best way to represent data structure and validation rules in XML- aware applications. In order to parse a flat file to create an XML instance, information about the data and its hierarchy needs to be understood properly and then captured in the schema definition. Once the data structure is defined correctly in the schema, the parsing instructions for the flat file need to be introduced properly in the schema so that after producing an XML instance from a schema definition, the instance may be populated with live data from the flat file.

The following steps explain the conversion of a flat file to the XML Schema.

- ***Data representation:*** Present the flat file in a schema form with required information on data structure and validation
- ***Parsing logic implementation:*** Develop a concept of container (record) and contained objects (fields) in the flat file and place relevant control attributes for them in the schema definition
- ***Default instance generation:*** Convert the schema into a default XML instance governed by schema rules
- ***Populating the instance with data:*** Parse the flat file with the control attributes information provided in the schema and

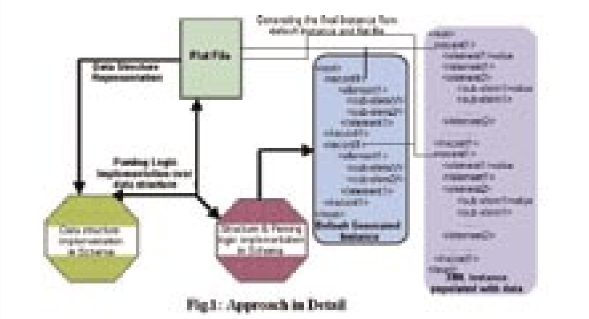

Generating the final instance from default instance and flat file
Flat File
Data Structure Representation
Parsing Logic Implementation over data structure
Data structure implementation in Schema
Structure & Parsing logic implementation in Schema
Default Generated Instance
XML instance populated with data
Fig1: Approach in Detail

Figure 1 • The conversion of flat files to an XML instance populate the generated instance with live data from flat file

To elaborate on the process in detail and explain the underlying Java implementation to convert flat files to an XML instance, a "," delimited flat file and a fixed-length field flat file are shown below.

A "," delimited flat file:
Social Security Number, Name, Salary
123456789,Ram Singh,100000.00
444556666,Barr Clark,87000.00
777227878,Simi? D Roy,123000.00
998877665,Charr Lee,92000.00

A fixed-length field flat file:
Social Security Number, Name, Salary
123456789Ram Singh   100000.00
444556666Barr Clark   87000.00
777227878Simi D Roy   123000.00
998877665Charr Lee   92000.00

In these flat files, the data and the data structure remain the same but the representation changes depending on the flat file type. Now let's go through the steps to convert these flat files to an XML instance. Figure 1 shows the approach in detail.

Data Representation

The main focus of converting data from flat file to XML instance depends on proper understanding of the underlying flat file data structure, which can be easily captured as a schema definition. In all of the preceeding examples the data structure remains the same, only the representations are changed depending on the flat file type. So, the basic data structure in these cases may be captured in the same schema definition shown in Listing 1.

For all the examples, any one of the two schema representations may be adopted. Both representations exhibit the flexibility to describe the data structure using the W3C schema language. In our example the first schema definition is considered. The examples here are simple but the same approach applies to complex cases as well.

Parsing Logic Implementation

The raw schema definition defined in Listing 1 captures the data structure of the flat file. To parse the flat file and to populate an XML instance, more parsing logic needs to be built inside the schema definition. To implement the parsing logic in the schema definition, annotations and control attributes with namespace, xmlns:t2xml="http://xmlns.oracle.com/t2xml" are introduced. These control attributes and annotations are added in the element declarations in the schema definition to mark the XML root element, physical container definition (records), and contained objects (fields) as per the flat file structure. Table 1 shows the list of main control attributes required to build the parsing logic in the schema.

ontrol Attribute	Description
t2xml:rootelem	Specifies the root element. Add this to the schema element that is specified as the root element. As there is only one root element in an XML instance, only one schema element that qualifies as the root of the XML instance should have this attribute.
Container-Specifi	
t2xml:container	Marks a schema element as a container i.e., it will contain other elements inside it. It is similar to a record in the flat file that contains the fields inside.
t2xml:container_endtoken	Marks a container to fetch data from the flat file using the container end token as a delimiter. If this attribute is present in a schema element, it guides the engine to get records from the flat files to populate objects contained in that container.
t2xml:container_def_id	Tells about the definition id of a container. This might be a value that identifies a container. If a container does not have a particular ID to identify it, this attribute might be omitted or "generic" might be assigned as the attribute value.
t2xml:container_position	Tells the position where the "t2xml:container_def_id" might be found in the container. For the delimited container case, it's the position of the object from the start of the container that holds the "t2xml:container_def_id." For the fixed-length case it's the absolute position of the "t2xml:container_def_id" from the start of the container.
t2xml:container_type	Tells the container type, i.e., whether it's a delimited or fixed-length type of container. The only two allowed values currently are "delimited" and "fixed."
t2xml:object_sep	Should also be defined in the container-schema element in case of a delimited container. It tells about the object delimiter used to separate objects in a container.
t2xml:escape_char	Marks the escape character in the container definition.
Contained Object Specific Attributes	
t2xml:object	Mandatory while defining fields (objects). This attribute marks an element for a field (object) representation inside a container (record).
t2xml:object_len	Mandatory if the container (record) consists of fixed-length objects (fields). This attribute should contain the exact length of the field in the text file. In case of a mismatch, parsing will fail or provide unexpected results.
t2xml:ignore	Sometimes the flat file may contain fields and records that must be suppressed while creating the XML instance. If present, this tag will suppress the element in the generated XML instance.
t2xml:keep-current	Required if an object value needs to be reused in the subsequent object value population. It just keeps the current value after populating the current object.

Table 1 • Control attributes and their meanings

These control attributes are pivotal to parsing the flat file and are defined in the t2xml.IParseProperties interface. This interface is implemented in the class xerces.xs.instance.T2xmlInstance to implement all the parsing logic to populate the XML instance with flat file data discussed in detail in the "Populating instance with data" section. See Listing 2 for the control attributes.

Apart from these control attributes, the attributes minOccurs and maxOccurs play a crucial role in determining repetitions of containers (records); depending upon the value of the minOccurs and maxOccurs, the optional and required containers are decided. For example, if minOccurs is "0," the container is optional; if it is more than "0," the container is mandatory. If maxOccurs is "unbounded," the number of containers is decided depending upon the records in the actual flat file. However, if a number is prespecified in the schema, that number of records is anticipated in the flat file.

Now let's see how these control attributes are used in the schema definition to mark parsing logic instruction for the flat file.

Delimited Flat File Case

For the "," delimited case let's consider a record from the flat file.

```
777227878,Simi? D Roy,123000.00
```

This shows an employee record with three fields: Social Security Number (SSN), name, and salary. The name field can be subdivided into first name and last name and separated by a “ ” delimiter. The “?” is considered an escape character in the name field. Figure 2 shows the basic mapping for a delimited record to a schema definition.

In Figure 2 the full record is mapped to the Employee element. Since the record is a delimited one, the following control attributes are added to the Employee element:

- *t2xml:container="true"*: Added to tell if it's a container or record
- *t2xml:object_sep=","*: Added to tell the field delimiter
- *t2xml:container_type="delimited"*: Added to tell the record type is delimited
- *t2xml:container_endtoken="os:linesep"*: added to tell the os-specific line separator is used as a record terminator
- *t2xml:escape_char="?"*: Added to tell the escape character in the record definition

Apart from these control attributes, maxOccurs=" unbounded" is found in the Employee element declaration. It says to produce as many Employee elements as are encountered in the flat file.

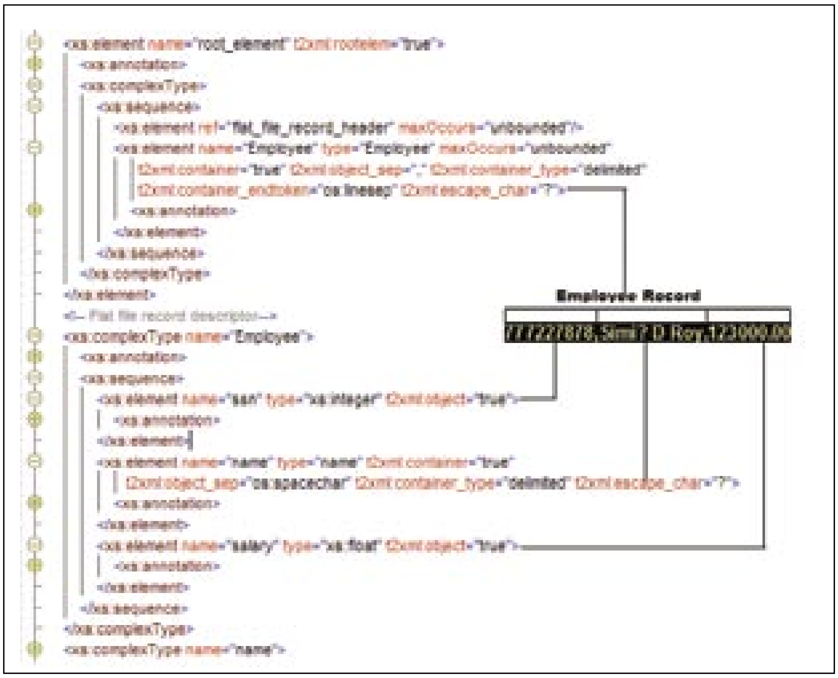
For the SSN and salary fields the mapping is simple, as these are contained within the Employee container and do not have any additional contained objects inside. The following control attribute is added for them:

- *t2xml:object="true"*: Added to tell that it's a contained object

For the name field the mapping is a little complex as it contains the subfields first name and last name. Thus it's a container as well as a contained object itself. The following control attributes should be added for it.

- *t2xml:container="true"*: To tell if it's a container for first name and last name
- *t2xml:object="true"*: To tell if it's also a contained object inside Employee
- *t2xml:object_sep="os:spacechar"*: To tell the os-specific space character as delimiter
- *t2xml:container_type="delimited"*: To tell the Container type as delimited
- *t2xml:escape_char="?"*: To tell about the escape character

The complete schema definition is located in the file delimited-sample.xsd in the source JAR (the source code for this article is at www.sys-con.com/xml/sourcec.cfm).



```
<xs:element name="root_element" t2xml:rootelem="true">
  <xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="flat_file_record_header" maxOccurs="unbounded"/>
      <xs:element name="Employee" type="Employee" maxOccurs="unbounded"
        t2xml:container="true" t2xml:object_sep="," t2xml:container_type="delimited"
        t2xml:container_endtoken="os:linesep" t2xml:escape_char="?">
        <xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!-- Flat file record descriptor-->
<xs:complexType name="Employee">
  <xs:annotation>
  <xs:sequence>
    <xs:element name="ssn" type="xs:integer" t2xml:object="true">
      <xs:annotation>
    </xs:element>
    <xs:element name="name" type="name" t2xml:container="true"
      t2xml:object_sep="os:spacechar" t2xml:container_type="delimited" t2xml:escape_char="?">
      <xs:annotation>
    </xs:element>
    <xs:element name="salary" type="xs:float" t2xml:object="true">
      <xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="name">
```

Figure 2 • Delimited record mapping to a schema definition

Fixed-length Flat File Case

For the fixed-length case too let's consider a record from the flat file.

```
777227878Simi? D Roy
123000.00
```

The fixed-length field case is almost the same as the delimited case; the only difference is that here the field lengths are fixed and not separated by any delimiters. Therefore, the control attributes for the Employee element are a little different from those of the delimited case.

- *t2xml:container="true"*: Added to tell it's a container or record
- *t2xml:container_type="fixed"*: Added to tell the record type is fixed
- *t2xml:container_endtoken="os:linesep"*: Added to tell the os-specific line separator is used as a record terminator

The attributes t2xml:object_sep and t2xml:escape_char are abolished here as they are specific to the delimited case only.

For the contained objects there is an additional attribute to specify the object length. The respective lengths of the SSN, name, and salary fields were updated in the t2xml:object_len attribute. So, for contained objects here are the required attributes:

- *t2xml:object="true"*: Added to tell it's a contained object
- *t2xml:object_len="9","30","9"*: For SSN, name, and salary fields

For the name field, mapping the other control attributes remains the same as in the delimited case.

The complete schema definition for the fixed-length case can be found in the file fixedLength-sample.xsd in the source JAR.

Default Instance Generation

Based on the schema defined for the flat file data structure, a default XML instance that follows the rules defined in the schema definition is generated. Generating this XML instance from the schema definition requires the proper identification of the root element to start the instance generation. This XML instance generation from the schema is implemented in the xerces.xs.instance.SchemaInstance class. This class can generate an XML instance from any schema definition, provided the schema contains at least one element declaration. If there is more than one element declaration, it calls for the element with no reference from other elements as a potential candidate for the root element. A generic root element finder is implemented in the xerces.xs.instance.RootElementFinder class. The XML instance generation in the SchemaInstance class starts with the root element and comes up with XML elements on traversal of complextypes and elements defined in the root element definition. It converts each complex type/simple type

continued on page 58

HOME
ENTERPRISE SOLUTIONS
CONTENT MANAGEMENT
DATA MANAGEMENT
XML LABS

WRITTEN BY **DAN DUBE**

Top Five Traps in a "Content Supply Chain"

Don't mangle your publishing issues

Before the information age, car manufacturers only made cars, libraries only stored books, and newspapers only printed the news. Now, however, companies from all industries are realizing that in addition to what they do, they are also publishers, and there is a learning curve.

Carmakers have to create, manage, and produce manuals, catalogues, and brochures. Libraries have to preserve collections and make their stacks available to a digital readership. Newspapers have to splice and dice their daily report millions of different ways to fit the distinct consumer tastes. In addition, these organizations have to publish in multiple languages and multiple formats: CD-ROM, PDF, Web page, hard copy, etc.

In the information age, all companies are publishers, and that's not even counting internal demands for content that every company has: memos, annual reports, newsletters, Intranet sites. However, in this rush to transform content products, many companies are actually losing ground, credibility, and money due to the innovation choices they have made.

The speed and sweep of these new choices – from plates and ink to digital rendering, from product development to knowledge development, from product selling to task support – have caused technology whiplash for corporate America. Sparkling new techniques are being implemented with less-than-sparkling outcomes, causing those promised returns on investment to look puny to the shareholders.

An all-too-common example is the poor implementation of a new vocabulary like Extensible Markup Language (XML) to reflect a historical need (like page layout and typesetting). A shoddy job can cause more complicated, unwieldy systems, and products that still lack agility. This example outlines the difference between page-based views (the old way) versus meaning-based views (the new way).

When a legal publisher "typesets" its law journals in open source codes like XML, the implementation involves rendering words by meaning. By tagging an italicized word as <italic>, the publisher shuts off that content from the advantages of XML. If the word has been italicized because it is a title, the proper tag is <title>. If done correctly, the publisher could later search the full database for titles, an impossible task if the word is simply rendered <italic>. This common mistake, along with dozens of others that occur in content manufacturing systems, is the reason why many global companies are struggling to compete in the information age.

XML does not just make things look a certain way on paper or on the Web. It's a way to build meaningful languages that companies use to specify and reflect the true meaning of the content being developed into information products. Typesetting conventions are used to imply deeper meaning; XML can be used to make that meaning explicit, accurate, and precise. The typesetting, or online rendering, comes later, with the added benefit that the content now reflects a higher knowledge quotient, enabling more targeted search. The tech-savvy consumers' needs are met alongside the companies' traditional buyers, and future reuse, retasking, and redeployment of content are built in with little or no added cost.

The first step is to understand that even if you are not in manufacturing, you depend on a supply chain, a content supply chain, and its proper function is vital to your bottom line. Otherwise, you're putting a drain on ROI from both directions, escalating costs and lagging innovation.

Having evaluated many content supply chains for publishers, manufacturers, institutions, and government agencies, I have ranked the top five traps in content supply chain management:

1. ***Inefficient processes:*** Don't get bogged down in manually intensive tasks (like troubleshooting format problems in a word processing program) or redundancies (like composing the same content twice for a hard copy and a Web

AUTHOR BIO

Dan Dube is director of business analysis at Innodata Isogen, a provider of content supply chain solutions. Innodata Isogen (www.innodata-isogen.com) optimizes content supply chains, helping clients realize significant cost savings and productivity gains from content-related operations, achieve better outcomes, and compete more effectively in demanding global markets.

site). In addition, hundreds of hours per year can be lost recreating information that already exists (like copying and pasting content from one document to another), or converting Word files to PDFs or HTML. Languages like XML offer a simple way around this kind of time-sapping redundancy.

2. ***Organizational boundaries:*** Don't let separate departments duplicate content production activities (like posting the same information on another page of the company Web site). In-house experts from each department of an organization will often achieve the departmental goal but never see the big picture. When the right hand doesn't know what the left is doing, you aren't coordinated.
3. ***System limitations:*** Over the years, different departments have ended up with different systems that don't talk to each other, or don't translate information from one to another. Proprietary systems architectures also make it difficult to move data from legacy systems to newer technologies. Don't let your organization be beholden to a software maker that may be out of business in twenty years. You've got to be able to control your content in perpetuity.
4. ***Technology limitations:*** Don't use a wrench to hammer a nail. Don't extend an application beyond its capability (like using MS-Word for final composition when it usually leads to extra labor hours troubleshooting formatting errors). Many companies pay internal IT staff to write internal "patches" to transfer data from one system to another, which ends up costing more because they perform worse than integrated systems, and you have to keep applying new bandages.
5. ***Inadequate use of resources:*** Assign your high-skill/high-cost labor only to the most important tasks, not to support activities that could easily be outsourced. For example, a production editor whose primary skill is page layout and Web site design may not need to be in-house because the skill requires no expertise in that field. Expensive resources should not spend time doing repetitive tasks.

Let me stress that these are just the top five traps. Many others exist, and companies are discovering new and terrible ways to mangle their content supply chain every day.Proper use of XML and other formats in your business systems requires that you first read the manual, so to speak. I also urge you not to delay. If you don't offer additional value like this in modern content and modern information products, customers will go across the street and get it from your competitor. In a marketplace this competitive, proper XML implementation could make the difference between leading and following.

dan.dube@innodata-isogen.com

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

continued from page 54

element to a default XML instance node during schema traversal. If some elements are optional, it produces one default element. If some elements have maxOccurs="unbounded," it goes with the value defined in Instance.MAX_UNBOUND_OCCURANCE. The most important methods of SchemaInstance are shown in Listing 2.

The source JAR contains the full class declaration for SchemaInstance class. All the handler methods in the class are recursively called to generate the XML instance. Below is the sample code that demonstrates how to use this class to generate an XML instance from a schema definition:

```
    SchemaInstance lSchemaInstance= new Schema
Instance(aSchemaFilePathOrURL);
   lSchemaInstance.generate(aOutputFilePath);
/* or */
   lSchemaInstance.generateInstance(aOutputSt
ream);
```

The XML instance generated from the schema example above looks like this:

```
             <root_element>
<Employee>
         <ssn/>
         <name>
             <fname/>
             <lname/>
         </name>
         <salary/>
</Employee>
 </root_element>
```

Populating the Instance With Data

As populating the XML instance with data comes after the XML instance is generated, one point worth mentioning here is that the XML instance to be filled up with data does not go on with the whole schema definition at one time; rather, each schema element contained in the schema definition under the root element is converted to the XML element and then filled up with data. When the XML instance generation starts from the root element, the control attributes are examined for each schema element that is generated as XML.

These control attributes tell if an element is a container or contained object, and also tell about the container end token, object separator, etc. Therefore, depending upon the control attributes, after the XML instance is generated for a particular schema element, the physical record is read from the flat file and the instance is populated with live data from flat file. This process is repeated for each record defined in the flat file. Only after traversal of the full schema definition (starting from the root element) will a filled-up instance representing the full schema definition be created. If the maxOccurs attribute is "unbounded" for a schema element, the number of XML instances for this element is created as per the availability of records in the flat file; otherwise the actual number is regarded in the schema definition.

The lookup for the control attributes and their correct handling is very important when filling the XML instance with data. To start, the implementation class xerces.xs.instance.T2xmlInstance implements the t2xml.IParseProperties and extends the xerces.xs.instance.SchemaInstance class. In the SchemaInstance class, the barebones XML elements were generated, but in the derived T2xmlInstance, with the help of the IParseProperties, these XML instances will now be filled up with data from the flat file. In Listing 3 you can see a skeleton representation of the T2xmlInstance class.

The last three methods in the class skeleton were overridden in the T2xmlInstance class from the SchemaInstance class to fill up the XML elements generated in the SchemaInstance class.

The getRootSchemaElement(.) method is overridden to find the root element in accordance with the control attribute "t2xml:rootelem."

The handleParticle(.) method is overridden to look up specific control attributes for container and container type, so that the filler object that fills the data to the generated XML instance is set up properly.

The fillupData(.) method is overridden to fill up data in the XML instance based on the control attribute for object marking and the type of filler object passed in.

The other methods in the class are helper methods to get the instance fill-up mechanism working. The full source code for T2xmlInstance may be found in the source JAR.

The T2xmlInstance class uses the control attributes explained in Table 1 to populate the default XML instance with data. To run T2xmlInstance as an application, download source.jar, unzip the contents, and try the following commands for the delimited sample case and the fixed-length case, respectively.

```
{your jdk home}\bin\java -cp .;classes;lib\xerces\
resolver.jar;lib\xerces\xercesI
mpl.jar;lib\xerces\xml-apis.jar;lib\xerces\xml-
ParserAPIs.jar xerces.xs.instance.
T2xmlInstance test\delimited-sample.xsd test\
delimited-input.txt test\delimited-output.xml
```

```
{your jdk home}\bin\java -cp .;classes;lib\
xerces\resolver.jar;lib\xerces\xercesI
mpl.jar;lib\xerces\xml-apis.jar;lib\xerces\xml-
ParserAPIs.jar xerces.xs.instance.
T2xmlInstance test\fixedLength-sample.xsd test\
fixedLength-input.txt test\fixedLength-output.
xml
```

The XML instance for the delimited and fixed-length case is shown in Listing 4. Because the data structure is exactly the same for these two cases, the generated XML is also identical.

Scope for Future Enhancement

This article demonstrated the concept of converting flat files to an XML instance and provides a solution to prove the concept. Fairly complex flat files may be converted to XML instances using this approach, but the solution needs to be improved for industry-standard XML-instance generation because the support for doing custom calculations and mapping data directly to attributes is not yet established. Enabling XPath expressions might add some more goodies to the approach.

Conclusion

This approach is used to parse flat files and create an XML instance given a schema representation of the underlying data in the flat file. Many flat file–based systems such as EFT/EDI, custom database migration, and backup utilities will find this useful. Since this approach is based on open standards such as W3C schema and Xerces implementation for schema, it may be widely used.

References

- *XML Convert 2.2 to transform flat files into XML and vice versa:* www.unidex.com/xflat.htm
- *XML Instance from TIBCO Software*: www.xml.com/pub/p/259
- *A description of an XSL-based approach to convert a schema to an XML instance:* http://incrementaldevelopment.com/papers/xsltrick/#schema-to-instanc
- *Sun's XML instance generator:* wwws.sun.com/software/xml/developers/instancegenerator/
- XML instance generator*:* http://xml-xig.sourceforge.net/
- *W3C schema:* www.w3.org/XML/Schema

Author Bio

Indroniel Deb Roy works as a project leader for XML Publisher with Oracle. He was previously involved in developing Novell's XML integration server and has worked with various XML and J2EE technologies for the past six years.

indroniel.roy@oracle.com

WebRenderer™

Standards compliant embeddable web browser component

WebRenderer is a cutting edge embeddable Java™ web content rendering component that provides Java applications and applets with a fast, standards compliant HTML and multimedia rendering engine. WebRenderer provides a feature-packed API including complete browser control, a full array of events, JavaScript interface, DOM access, document history and more.

Why WebRenderer?

- Standards Support (HTML 4.01, CSS 1 & 2, SSL, JavaScript, XSL, XSLT etc.)
- Exceptionally Fast Rendering
- Predictable Rendering
- Scalability (deploy in Applications or Applets)
- Security (based on industry standard components)
- Stability and Robustness

Embed WebRenderer to provide your Java' application with standards compliant web content rendering support.

To download a 30 day trial of WebRenderer visit
www.webrenderer.com

JadeLiquid™ Software