SYS-CON
NOW INCORPORATING XML JOURNAL

Web Services JOURNAL

.NET J2EE XML

November 2004 Volume 4 Issue 11

Web Services: Monitoring and Management for Reliability

Do you need them?
pg. 18

Web Services + the Grid = Prime Time

Will the convergence of Web Services and Grid technologies complete the puzzle
pg. 26

Process-Centric Realization of SOA

BPEL moves into the limelight
pg. 34

Web Services in Action

Integrating with the eBay marketplace
pg. 38

See page 30 for details

EVOLVING JMX

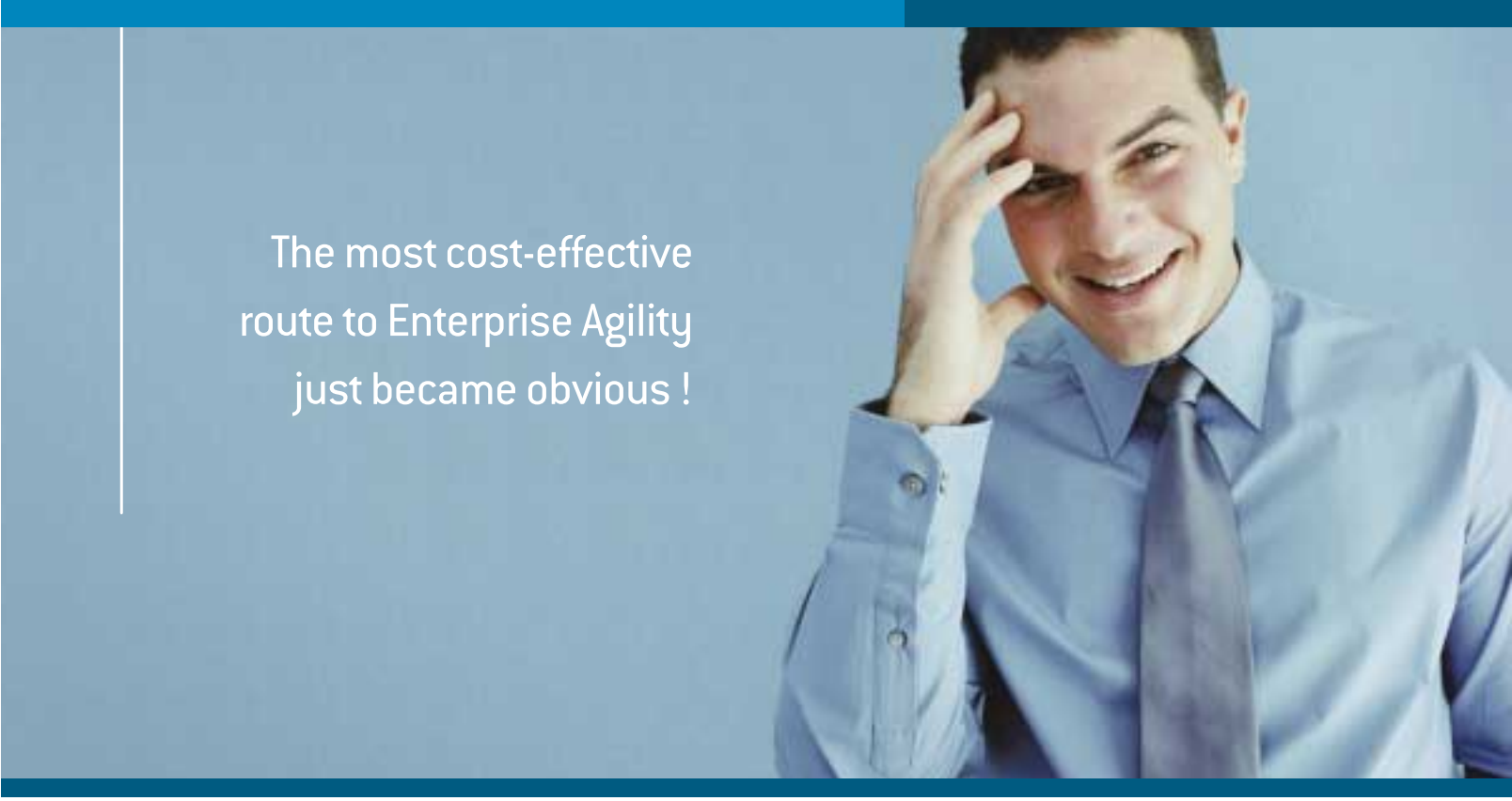
Supporting Web Services Management Technologies

pg 10

pg 8

Bright Future for J2EE Web Services Development

BY DALE L. FULLER



Implementing an Enterprise Architecture for organization-wide agility requires a clear understanding of intricate relationships across technology realms such as: Applications, Databases, Operating Systems, and Development Environments.

OpenLink Virtuoso is a new generation Business Integration Platform that cost-effectively unravels the potential of Enterprise Information Integration, Web Services, Service Oriented Architecture, and Business Process Management.

Virtuoso's industry acclaimed product architecture allows you to embrace and exploit the enterprise-wide heterogeneity of your IT infrastructure.

Empower your organization today – download a FREE evaluation copy from our Website or call us at 1 800 495 6322.

Database Support
Oracle, SQL Server, DB2, Sybase, Informix, CA-Ingres, Progress, MySQL, PostgreSQL, Virtuoso, and other ODBC or JDBC accessible databases

Standards Support
SQL, ODBC, JDBC, XML, XSL-T, XQuery, XPath, XML Schema, HTTP, WebDAV, SOAP, WSDL, WS-Security, BPEL, UDDI, NNTP, POP3, SMTP, and SyncML

Platform Support
Windows, UNIX (all major flavors), Linux, Free BSD, and Mac OS X

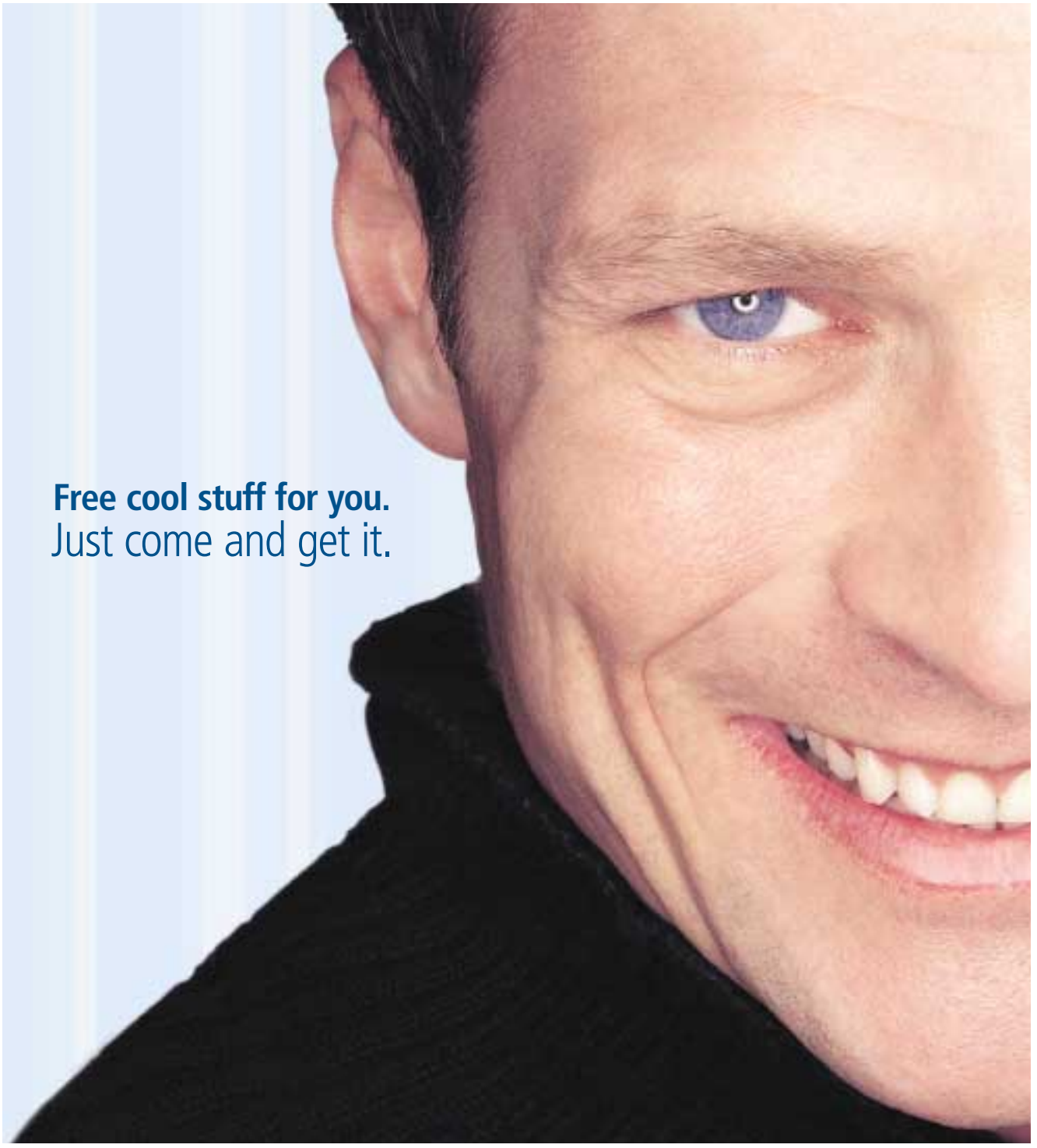
Runtime Environment Hosting
Microsoft .NET, Mono, Java, Perl, Python, PHP, and others

Programming Languages
Any .NET bound language (C#, VB.NET, others), Java, C/C++ , and others

Web Application Runtime Hosting
ASP.NET, Java Server Pages, PHP, Perl, and Python

http://virtuoso.openlinksw.com

OPENLINK SOFTWARE



Global Knowledge™
Experts Teaching Experts

Want a free Apple iPod Mini?

Purchase an Application & Web Programming training course through our online store by December 31, 2004 and get a free iPod.

Visit www.globalknowledge.com/wsj to discover expert Application & Web Development training. Find the course you need and get the Cool Stuff you crave. Act now: you must purchase by December 31, 2004 to qualify for the Apple iPod.

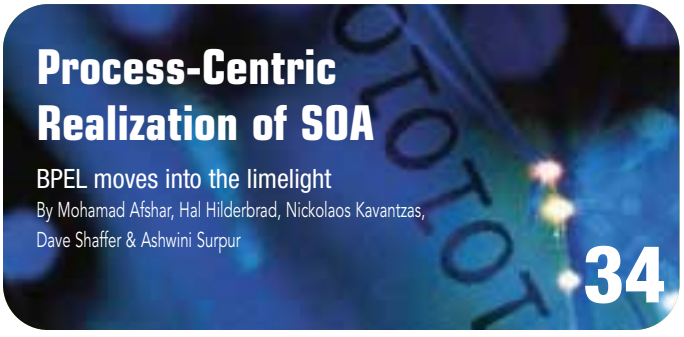
Web Services Journal
.NET J2EE XML
XML Journal
NOVEMBER 2004 Volume 4 Issue 11

Inside WSJ

WSJ: Features

Evolving JMX
Supporting Web services management technologies
By Pankaj Kumar & Chris Peltz
10

Web Services + the Grid = Prime Time
Will the convergence of Web services and Grid technologies complete the puzzle?
By Rickland Hollar
26



Process-Centric Realization of SOA
BPEL moves into the limelight
By Mohamad Afshar, Hal Hilderbrad, Nickolaos Kavantzas, Dave Shaffer & Ashwini Surpur
34

Achieve higher quality

in less time, with fewer resources.

Parasoft® Automated Error Prevention (AEP) Products:

Specifically designed for under-staffed, over-committed development organizations like yours.

Parasoft AEP Products ensure that comprehensive error prevention practices are applied consistently and uniformly across your entire team.

By automating compliance to a unified set of testing and coding standards, Parasoft AEP Products enable every team member to follow the same best practices and the same error prevention techniques – including coding standards analysis, unit testing, code review and regression testing.

Automated testing and standards compliance for any team configuration.

Parasoft AEP Products automate error prevention for Java, C/C++, Web Services, the Microsoft® .NET Framework, Embedded Development, Web Development, Database Development and more.

For details go to: www.Parasoft.com/AchieveQuality

Call: 888-305-0041 x3307 Email: AchieveQuality@Parasoft.com

PARASOFT®
We make software work.™

Founded in 1987, Parasoft's clients include IBM, HP, DaimlerChrysler and over 10,000 companies worldwide.

Availability

Parasoft AEP Products are available on Linux, Solaris, and Windows NT/2000/XP.

Contact Parasoft for details and pricing information.

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

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

NEW! 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 re-produced 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

Mindreef, Inc.
www.Mindreef.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

Going the Last Mile

WRITTEN BY
SEAN RHODY

One of the things I continually rail about in the Web services world is the lack of last mile technology – you know, the ability to get a Web service in front of a user. Even though a large number of Web services are designed solely for computer to computer communication the continuing reality is that more Web services are designed to interact directly or indirectly with human beings.

The problem, as I've stated before, is that there is no concept of user interface in Web services – no way to go that last mile and reach the end user. We've got a plethora of technologies to solve other problems, such as security or transactional integrity; sometimes we even have multiple standards for solving those problems.

Now, don't get me wrong, I know that having a transactional semantic, as complex as it sounds, is simpler than providing a generic approach to user interface. Computers have different operating systems, with different presentation constructs. Java took a shot at ubiquity of user interface with its write once/run anywhere approach, but the least common denominator approach they took with AWT and Swing left a bad taste in many people's mouths (not to mention that it left many people going to get a cup of coffee while they waited for the screen to respond). Regardless, Java failed to create a ubiquitous front end.

Ideally, we've abstracted the business rules into Web services, so all that's left for the front end is presentation logic – displaying data, acting as a data-gathering device that forwards all requests in coarse grained calls to a back end Web service.

Reality makes it a little more complex, as screen data population (think drop down list boxes, etc.) is rarely part of a pure Web services API design. So some additional Web services are usually needed for screen support.

Mix in things like validation (making sure the social Security number is nine digits, no alphabet characters), dependencies between screen controls (like filtering one drop-down based on the selection in another) and conditional displays (graying out options, or removing screen elements based on selections), and you have something that is more complex than just rendering screen elements. Then try to figure out how to do this in a neutral manner with regards to operating systems, and you have a good idea why no one has wanted to go here.

There is some hope however. Microsoft will be releasing a technology known as Avalon as part of their Longhorn release. Avalon will abstract the presentation logic into a format known as eXtensible Application Markup Language, or XAML. Based on XML, XAML describes the presentation aspects of an application's user interface. Like HTML, it is tag based, but unlike HTML, it's designed to be implemented on a native user interface (Windows, in this case), although it can also be utilized in a browser.

At the moment, no mention has been made of placing this language in open source or making a Web services standard out of it. Let me be the first to urge that this happen. Please, donate this to the community.

It would or should not be difficult to move a language that describes a user interface to other platforms. XAML will also work in Internet Explorer and for all intents and purposes that's a separate platform. And realistically, the language describes screen elements that are nearly identical across platforms anyway – a button is a button is a button, whether it's on a Mac, a Windows machine, a Solaris box, or a Linux PC. Sure there's always the chance of incompatibilities, or dummying down, but I'd take that (in the first release, at least) in exchange for being able to describe my application without code and let the operating system figure out how to run it, and what to display.

So I'm going to keep my fingers crossed. This is a good idea. Finally, the last mile.

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
ADVERTISING MANAGER
Megan Mussa megan@sys-con.com
ASSOCIATE SALES MANAGERS
Kristin Kuhnle kristin@sys-con.com
Beth Jones beth@sys-con.com
Dorothy Gil dorothy@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
VP, 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
Shannon Rymsza shannon@sys-con.com

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

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

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

SYS-CON MEDIA

Bright Future for J2EE Web Services Development

WRITTEN BY
DALE L. FULLER

Web services are already changing the integration model for enterprise applications, and they offer even greater promise for the future. It's not going to happen overnight, but there is an ever increasing focus on incorporating Web services into new applications and existing infrastructure in order to achieve standards-based information exchange. This expanding universe of Web services will allow not just J2EE applications, but all applications, to interact and exchange information freely, and permit development to proceed unencumbered by proprietary communications constraints.

Both J2EE and the Microsoft .NET framework have pushed the edge of the Web services envelope. Both platforms have already adopted Web services as their future integration model, demonstrating that Web services will become even more widely accepted and soon emerge as the dominant integration method.

Currently, J2EE is the prevailing enterprise development environment. More and more enterprises are already starting to expose their massive installed base of J2EE applications as Web services. The J2EE platform is now extremely powerful and broad in its capabilities, integrating features for both the server and client sides to enable existing enterprise Java components to be exposed as Web services and Java clients to invoke them. And more Java advancements are in the works.

The Java community is already at work on J2EE 1.5. The next release of this powerful platform incorporates new standards and language features from the recently announced JDK 5.0 and makes it even easier and more productive to create Web services. The JDK 5.0 language extensions support more productive Java development and deployment, enabling the Java code to be annotated with higher-level business process and other information. These improvements allow even higher-level abstractions to be built on top of Web services to solve real business problems, mapping directly onto solutions that come from business domain experts and enabling Java to more naturally support business process execution language (BPEL) environments.

As J2EE has advanced, so has the maturity and number of .NET applications. Soon Web services developed for these platforms will be able to communicate freely and without restrictions based on their platform. Development shops will be able to take advantage of each platform depending on their specific business needs, without having to worry as much about integration and interoperability issues.

Even with all this progress, Web services are still at the beginning of their evolution. There will be continual pressure to evolve specifications for both .NET and J2EE and create new specifications that can help IT teams better manage these complex transactions.

Security, reliability, and performance are the three primary areas where there is the most immediate need for advancing Web services. Some in the industry believe comprehensive Web services security, the kind to bet a business on, is nowhere near where they would like to see it. And reliability, especially of third-party Web services, is still elusive.

Following the natural progression of most computing paradigms, Web services security will likely be the first area of concentration for improvement, with high availability of Web services not far behind. However, until Web services and the platforms on which they are built advance in these areas, enterprises may decide to stick with the older integration model of components, especially enterprises that demand extreme levels of performance, security and reliability.

Much work is already underway by vendors, the open source community, and standards organizations to address these challenges. In the meantime, enterprises can manage system performance and employ tools to help them achieve better performance of Web services today.

Tools available from Borland and other companies can already help manage and measure the performance of Web services, and help tame some of the complexities of developing a service-oriented architecture (SOA). For example, some of today's tools can simplify everything from modeling high-level views of enterprise SOA implementations, to applying best practices and successful patterns to implementations, to auditing software for security holes before the software is deployed.

Eventually, security and reliability, as well as performance, will be addressed at a more functional level within the technology platform itself. Vendors such as Borland, Sun, BEA, and IBM are already collaborating to push the technology

Continued on page 44

EVOLVING JMX

Supporting Web services management technologies

Increased demands to meet customer service-level agreements often put a great burden on IT to deliver a robust architecture for application management. Organizations wishing to deploy a management solution must consider a number of important requirements. For instance, sufficient monitoring must be in place to ensure the availability, performance, and system health of deployed applications and business processes. Additionally, management solutions must allow IT to control and configure applications. This control is vital to ensuring that the deployed assets can adapt to change over time.

Meeting these management requirements becomes even more difficult as organizations must manage components that are distributed and deployed on heterogeneous operating systems, platforms, and languages. IT management often requires the merger of a variety of management technologies, standards, and tools to successfully manage these IT assets. Some of these management standards, such as JMX and WMI, define standard interfaces for a particular platform that are resource specific and do not generally interoperate well.

WRITTEN BY
PANKAJ KUMAR &

CHRIS PELTZ

The Web Services Distributed Management (WSDM) Technical Committee was formed within OASIS to meet these challenges. This committee's charter is to define protocols for managing distributed resources using Web services standards and to develop a model for representing Web services as a manageable resource. As a result of this work, many organizations are asking questions such as "How can I leverage WSDM in my development?" and "Should I abandon JMX development in favor of WSDM?"

JMX and WSDM can and should fulfill an important role in the management of your IT environment. JMX is a key technology for instrumenting Java applications with manageability, and WSDM can serve as the overall interaction technology between your management systems and managed resources.

This article offers you further guidance on the use of these two management technologies in a single management ecosystem. We begin by reviewing the current state of JMX and WSDM. We then present specific challenges in mapping from JMX to WSDM. We present one potential solution to this mapping problem, followed by specific steps that you can take with JMX today to be better prepared for WSDM tomorrow.

Overview of Java Management

We will begin surveying the state of the Java Management Standards and the work done by the Java Community Process (JCP) to support management in both the J2SE and J2EE platforms.

The JMX 1.2 specification, defined in JSR 3, was initially created to specify an API and overall management architecture for the Java platform. The architecture, illustrated in Figure 1, consists of three distinct layers:

- ***Instrumentation layer:*** Responsible for encapsulating management resources using MBeans. MBeans are Java objects that implement resources and their instrumentation and follow the design patterns and interfaces defined in the JMX specification. The JMX specification defines four types of MBeans: standard, dynamic, open, and model MBeans.
- ***Agent layer:*** Contains an MBean Server that acts as a registry for MBeans, allow-

ing clients to locate and access them through a standardized interface. All management operations on MBeans are brokered through the MBean Server. Additionally, the Agent layer contains a series of additional services (timing, dynamic class loading, monitoring, and relationship) that can be dynamically loaded to extend the functionality of a JMX agent.

- ***Distributed layer:*** Contains a series of adapters and connectors that make the JMX agent available remotely. An MBean can be accessed either by local clients residing within the same JVM, or through remote clients running outside the address space of the JMX Server hosting the MBeans. The remote clients could be developed using a programming language other than Java. These clients access the MBean Server and the MBeans through a JMX Connector.

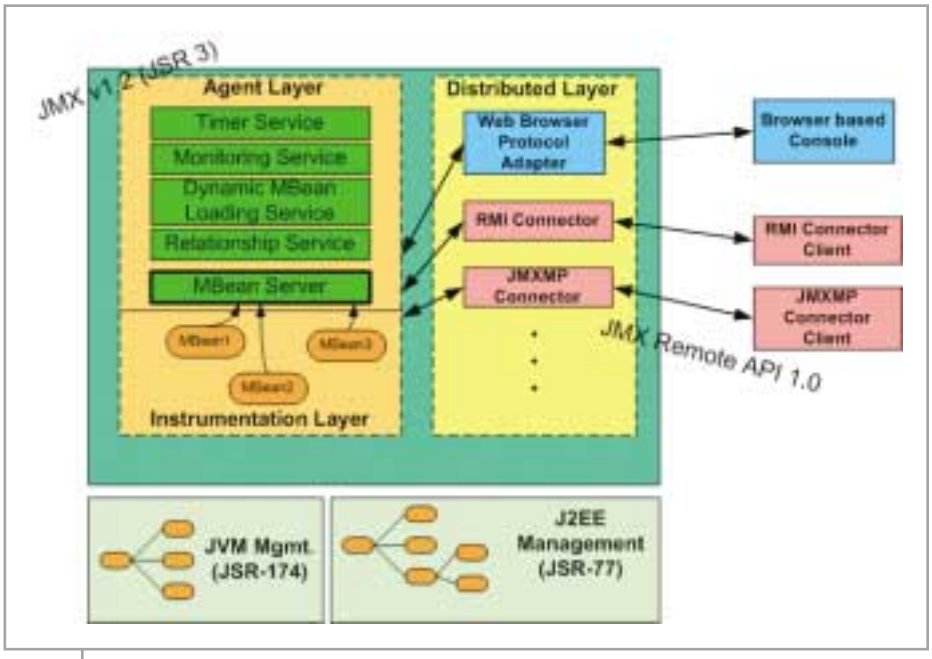


FIGURE 1 | **Management in Java**

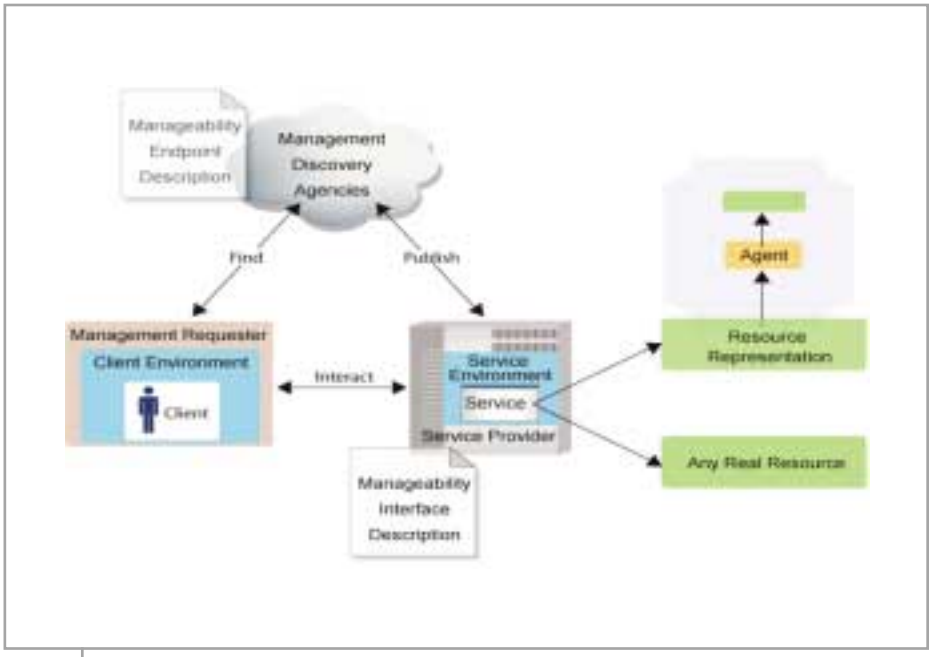


FIGURE 2 | **Management Using Web Services**

One important feature missing from the JMX 1.2 specification is a standard way for remote clients to access the management capabilities exposed through MBeans. The JMX Remote API 1.0 specification, defined in JSR 160, fills this gap and defines JMX connectors for RMI and an optional JMX Management Protocol (JMX-MP).

JMX-MP is a TCP-based protocol for accessing the MBean Server and MBeans from non-Java clients. A JMX-MP–based JMX connector would allow, at least in theory, non-Java management systems to manage Java applications. In practice, however, this approach is unlikely to find wide industry adoption because it requires implementing a new wire protocol for program-to-program interaction in different languages where many already exist.

You will notice that these specifications only outline the basic management mechanisms and do not define the specific management models, that is, the MBeans and their interfaces, for managing either a J2SE JVM or a J2EE Server. These tasks are addressed through two separate specifications, JSR 174 and JSR 77.

JSR174 defines a management model for a Java Virtual Machine (JVM). This JSR has added management and monitoring APIs in J2SE 5.0 that expose valuable information about the JVM, including health indicators such as memory usage, thread contention, class loading behavior, and garbage collection frequency. With the JMX support added in J2SE 5.0, JMX and RMI connectors are now part of J2SE, and a platform-level MBean Server is available within every JVM. Packaging of the JMX protocol within the JVM makes it much easier to instrument any Java application with manageability.

JSR 77 defines a management model for J2EE Server. This JSR defines a set of managed interfaces for resources exposed within a J2EE Server, including JMS, EJBs, and Web applications. These managed resources define a standard way to model and interact with the J2EE container, including dependencies between objects, and state and metrics on the components.

Overview of Web Services Management

Before making specific recommendations for how to evolve JMX toward supporting Web services management technologies,

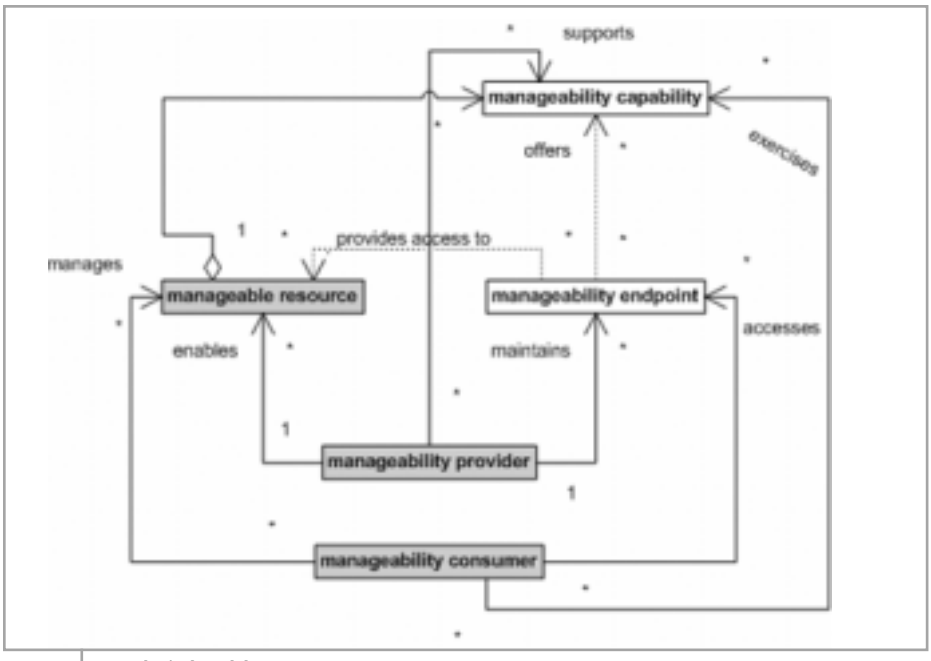


FIGURE 3 **MUWS logical model**

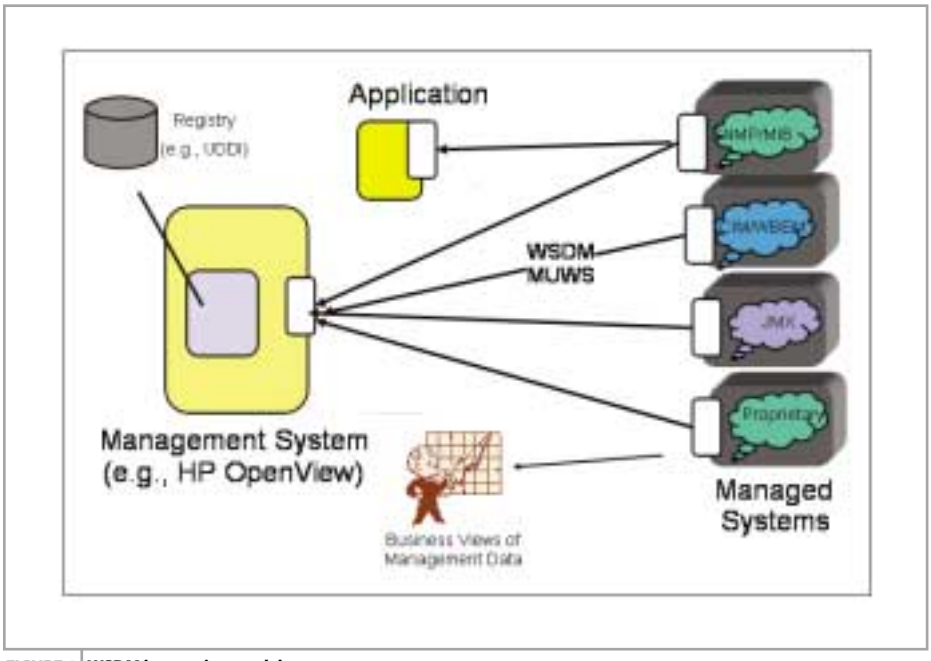


FIGURE 4 **WSDM interaction model**

we should spend a few minutes reviewing the state of Web services and Web services management. It would not be hard to argue that XML and Web services have emerged as the dominant distributed interaction mechanism across heterogeneous platforms over the Internet. Much of this adoption is due to their platform neutrality, extensibility, ease of use, and wide industry acceptance. Broadly accepted standards such as WSDL, SOAP, and UDDI are also helping to drive the use of Web services within and outside corporate boundaries.

The factors that make these technologies attractive for general-purpose distributed applications also make them an excellent choice for exchanging management information. As we have seen with Java management standards, two important things are necessary for developing management standards based on a new technology:

1. The basic mechanisms to format the management information
2. A model to manage the technology itself

To meet these objectives, the OASIS WSDM TC has been working on defining an approach for representing manageable resources using Web services interfaces, in addition to developing a management model of a Web service as a manageable resource. These capabilities are represented in the Management Using Web Services (MUWS) and Management of Web Services (MOWS) specifications, respectively. An important goal of the WSDM work is to provide an extensible architecture that complements existing management technologies (e.g., JMX, CIM, SNMP/MIB) while still offering model neutrality across these heterogeneous interfaces.

The basic architecture of a WSDM-based environment follows the typical SOA-based model of "publish-discover-interact," shown in Figure 2. A manageable resource is represented with a Web services interface using WSDL. This manageability interface is then exposed by a manageability provider. The manageability provider acts as a proxy between the client and the managed resource, enabling clients to execute management operations. The additional steps of describing the management interfaces, interacting between clients and management providers, discovering the endpoints, and securing the overall interaction are addressed by applying the corresponding Web services technologies.

MUWS defines a wire-level specification for managing distributed resources based on Web services technologies. It relies on a number of open standards, including XML, WSDL, SOAP, and UDDI. Additionally, the specification incorporates new standards such as WS-Notifications and WS-ResourceProperty.

Besides supporting traditional request-response interactions for invoking operations and accessing attributes, managed resources must also be able to send out one-way asynchronous notifications. Support for notifications is defined within the WS-Notifications specification.

While the Manageability Endpoint operations are describing using WSDL, the management capabilities, or attributes, are defined using resource properties as defined within the WS-ResourceProperty specification. A management capability could repre-

sent any number of aspects of a management interface for a resource, including:

- How do I distinguish among manageable resources?
- How do I get metrics (e.g., performance data) about a manageable resource?
- How do I monitor and configure manageable resources?
- How do I control (start, stop, restart) manageable resources?
- How is one manageable resource related to another manageable resource?

Figure 3 shows the relationships among some of the entities defined within the MUWS specification. Listing 1 outlines a few examples of how resource identity, state, and metrics can be represented using the MUWS specification. The Identity type is used to distinguish among manageable resources, ResourceState defines operations for starting and stopping the service, and the Metrics identify the metrics for the managed resource. These management capabilities help in exposing the management model for a resource, and MUWS offers the appropriate constructs for this model to be exposed in a standard way. It is important to keep in mind that WSDM MUWS does not define a management model but has constructs to expose such models.

JMX for Instrumentation, WSDM for Interaction

The WSDM MUWS specification is being developed primarily as a single, standards-based mechanism for management applications to interact with manageable resources. Yet it is also clear that MUWS is not intended to replace JMX-based instrumentation or other similar technologies. As Figure 4 illustrates, MUWS can serve as a bridge to these existing instrumentation technologies, including aspects of SNMP/MIB, CIM, JMX, or other proprietary management frameworks.

It should be evident that JMX continues to be a very relevant standard for instrumenting Java-based applications, even with WSDM in the picture. WSDM MUWS is not a programming language API, and developers still need the capability to instrument their applications with the appropriate messages, metrics, and control mechanisms; JMX clearly serves this purpose. But for more general management interactions (e.g., across applications, management systems, or managed resources), a more cross-platform, standards-based interface is required. For example, if you had a requirement to aggregate JMX resources with WMI resources in a distributed manner, this would not be a straightforward task. The WSDM MUWS specification provides the needed mechanism and platform neutrality to address this challenge.

The complementary nature of JMX-based instrumentation and WSDM-based interaction among managed applications and management systems is clear. However, to make integration a reality, developers need a mechanism to expose their JMX-instrumented applications using WSDM interfaces. Referring to the JMX architecture, this mapping could be addressed through a JMX-WSDM connector. Developers could use JMX to build manageability into their applications, and WSDM-compliant managers could be used to manage these applications through a JMX-WSDM connector.

Challenges

Creating a bridge between JMX and WSDM may appear to be a trivial matter. However, in practice, a number of factors complicate the mapping process. We have already discussed the lack of JMX APIs for exposing management models in standard manner. There are also some other difficulties in mapping from JMX to WSDM, as well as some specific WSDM constructs that are not present in JMX today.

- ***Inadequate support for exposing management models.***

As we noted earlier, the WSDM MUWS specification has the notion of a state model, metrics, and relationships that allow a management model to be exposed in a standard way. Today JMX does not support these capabilities at the same level, often leaving the architects or developers to create the necessary APIs on a per-project basis. This situation can be problematic, making it almost impossible to create a simple, automated way to translate any JMX-exposed management model through WSDM.

This isn't to imply that JMX cannot be used to expose a well-designed management model. But it is hard for a client to extract that model by simply using the available JMX APIs. The good news is that work is being done to expose management models via JMX. The J2EE Management specification, for example, exposes a J2EE Server management model built on JMX.

- ***Data type impedance mismatch.***

A JMX MBean in its most general form can use complex Java objects to represent attribute values, operation arguments, return values, and JMX notification fields. While WSDM offers support for complex types, automated translation from a JMX type to an appropriate XML Schema data type is not always straightforward.

In fact, this data type mismatch is problematic even in an all-Java world. The management application requiring access to the MBean needs to load the JAR files provided by the managed application. With such tight coupling, it can become quite challenging to manage two different applications instrumented with different versions of an MBean. The use of XML and Web services, by way of MUWS, can offer some assistance in shielding the client from being aware of these complex types.

- ***Complex mechanism for relationships.***

Relationships between managed resources are an important aspect of any management model. JMX supports the notion of relationships through its relationship service and Relation MBeans, but it is cumbersome to use in practice. Both the J2EE and JVM management models built on JMX do not currently take advantage of the relations construct. Although the details of how WSDM would represent relationships are not yet available, it is unlikely that there will be a one-to-one mapping with JMX.

Given these challenges, it would seem there are a few aspects of the base JMX specification that might make it difficult to support WSDM MUWS. Additionally, it may not be clear what you as a developer or architect should be doing in the JMX world to better prepare yourself for WSDM. To address both these concerns, let's turn our attention to some of the existing JMX-based management models, followed by a discussion of best practices that you can follow in your development today.

Towards the Solution

Earlier, we briefly discussed additional work being done in the JMX community to support management models. JSR 174 and JSR 77 define management models for the JMX and J2EE Server respectively. Let's take a minute to see how these models address some of the inherent limitations in JMX.

JSR 77 defines a set of conventions and classes for representing a J2EE Server management model. This includes support for a state model, notifications, and metrics:

- ***Mandatory attributes:*** Defines a set of mandatory attributes to allow a client to easily introspect the management model. These attributes identify whether the object supports a state model, metrics, notifications, etc. This support simplifies the step of automating the mapping between JMX and WSDM.
- ***State model:*** Defines the operations, state values, and valid state transitions for any managed resource that supports a state model. Supported operations might include deploy, undeploy, etc.
- ***Event notifications:*** Defines any notifications to be exposed by a managed resource, including state model transition and other changes in the management model.
- ***Metrics:*** Represents metrics to be exposed by a managed resource. To support this, a resource sets a mandatory attribute and implements an interface that allows a client to query about available metrics.
- ***Relationships:*** Describes containment relationships between objects. For example, a container-managed resource would typically store the list of contained managed resources in an array attribute.

Furthermore, the J2EE Server management model avoids the problem of data type impedance mismatch by restricting itself to only primitive types and data types defined in the specification itself.

For the JVM management model, JSR 163 introduced the notion of MXBeans. MXBeans are managed beans that conform to the JMX Instrumentation Specification and use only a specific set of data types. For example, MXBeans are provided to monitor and track memory consumption (via the MemoryManagementMXBean, GarbageCollectorMXBean, etc.) and CPU usage (via the ThreadMXBean, RuntimeMXBean, etc.)

One advantage of MXBeans is that they can only use a set of basic data types that can be marshalled and unmarshalled in a platform and programming language-neutral manner. These data types include primitive types (int, long, boolean, etc.) and their associated wrapper classes (Integer, Long, Boolean, etc.), enumeration classes (Enum), and classes that define a mechanism to convert from an input CompositeData class to an instance of that class. Additionally, List and Map types are supported as long as the contained objects are primitive types, wrapper classes, enumerations, or a class supporting CompositeData conversion.

Note that most of these data types are not specific to the JVM or J2EE Server management model, and they can help guide the definition of a management model for any Java-based application.

Mapping JMX to WSDM

Based on our discussion so far, let's take a stab at defining a very high-level mapping from JMX to WSDM. This mapping is shown in Table 1.

We acknowledge that we have glossed over many details and that the actual mapping of various JMX classes to WSDM MUWS structures might require a bit more work. But even this minimal mapping gives us a feel for how JMX and WSDM could work together.

However, before you rush to develop your own connector between JMX and WSDM, it is important to realize that several questions need to be answered:

- ***Should every JMX MBean map to a manageability provider?*** The existing JMX instrumentation mechanism does not use MBeans as a proxy for real managed resources. By blindly making every MBean a WSDM Manageability Endpoint, we may end up with a large number of MUWS Manageability Providers that do not conform to the spirit of MUWS.
- ***How do we handle Java data type mismatches?*** As per the JMX specification, developers are free to create their own Java classes for MBean attribute types, MBean operation parameters/return values, and JMX notification fields. The mapping doesn't specify what to do about Java classes that lack a straightforward counterpart in XML.
- ***How should we handle JMX relationships?*** Given the state of the MUWS specification today, is it appropriate to map a Relationship MBean to a Manageability Provider?
- ***How does this mapping account for a state model and metrics?***

JMX Entity	WSDM MUWS Counterpart
JMX MBean	WSDM Manageability Endpoint
MBean operation	WSDL described operation
MBean attribute	WS-ResourceProperties described property
MBean notification	WS-Notifications described notification
Relationship MBeans	MUWS Relationship counterpart

TABLE 1 | **Generic JMX to WSDM mapping**

Clearly, our mapping mechanism doesn't fully represent how JMX and WSDM would work together. There is still much to be done by the Java and WSDM communities to clearly define how these two management technologies would work together. In the meantime, it is still possible to adhere to an MBean design that would allow a much smoother transition to a WSDM-based management architecture.

JMX Development Guidelines

Many software projects focus on implementing the important functional requirements for an application. Often, these stated requirements do not consider manageability aspects of the application in the production environment. Applications "thrown over the wall" to production often leave the IT operations group with little information to guide them toward resolving application problems. In situations where the application has not been designed with manageability in mind, developers are frequently called in to help resolve these IT support problems.

We firmly believe that in today's complex and distributed world, manageability must not be an afterthought in the development process. Application manageability must be embraced early in the application development life cycle. And even if you are not considering WSDM in your design, the most important guidelines that we can provide are to first think about designing for manageability, and to understand which management technologies are most appropriate to address these concerns.

JMX defines a very powerful manageability architecture and API for Java applications. In

fact, most J2EE platforms are being built with robust JMX mechanisms. Still, there are cases where an additional level of instrumentation is required to expose application-level manageability interfaces. If you are thinking of using JMX, you can refer to the following guidelines. These guidelines help pave the way for embracing WSDM in the future as well as improve the overall quality of a JMX-instrumented application.

1. Do not fall into the trap of creating a JMX MBean for every single business object in your application. Instead, strive to cleanly separate MBean interfaces from the implementation-level business objects. Ideally, create MBeans as entities that expose management operations on resources that you would want to manage from an external management application. Use of the facade and proxy patterns can also help to expose the right management interface.
2. As we have pointed out here, there are many challenges involved in managing data type conflicts between WSDM MUWS and JMX. To address these challenges, avoid using complex data types wherever possible. JSR 174 and the MXBean interface define a strict set of guidelines for interoperable data type usage. We recommend that to minimize mapping issues, you follow those guidelines in the development of your own MBeans.
3. Finally, we have identified some gaps in JMX as it relates to support for state and metrics. Being able to define these manageability capabilities is essential for enabling an application to be managed, monitored, and controlled. Examine the conventions defined in JSR 77 and apply some of the same principles in your MBean design.

If you already have a JMX-instrumented application that does not follow the above guidelines, you can achieve a greater degree of interoperability through design patterns and frameworks. For example, you can create a Wrapper MBean around your existing MBeans to conform to some of these guidelines. By exposing your JMX instrumentation in a standard way, you can support WSDM management interfaces more easily in the future.

What's Next?

As demonstrated here, we are still far from the dream of using JMX as the instrumentation technology and WSDM as the interaction technology. To move toward making the dream a reality, we make the following recommendations to the Java community:

1. The JMX specification should be enhanced to support MXBeans and the capabilities from JSR 77 that help in representing a management model. This would allow application developers to reuse these APIs in a standard way. It would also be helpful to simplify the current JMX mechanism to support relationships.
2. A JMX-WSDM connector specification should be developed through the JCP so that any JMX-instrumented application becomes WSDM-manageable without any additional effort on the developer's part.

This article provided a solid background on JMX and WSDM as well as guidelines and best practices for using JMX in preparation for WSDM.

References

- For more on what HP is doing with JMX and WSDM, visit HP's developer resource portal (devresource.hp.com).
- *Java Management Extensions: Instrumentation and Agent Specification*: http://jcp.org/aboutJava/communityprocess/final/jsr003/index3.html
- *Java 2 Platform, Enterprise Edition Management Specification, JSR77:* http://java.sun.com/j2ee/tools/management/reference/docs/index.html
- *Java Management Extensions (JMX) -- Remote API 1.0 Specification*: http://jcp.org/aboutJava/communityprocess/final/jsr160/index.html
- *MXBean description:* http://java.sun.com/j2se/1.5.0/docs/api/java/lang/management/ManagementFactory.html
- *WSDM MUWS Working Draft:* www.oasis-open.org/committees/download.php/8464/wd-wsdm-muws-1.0-20040803.doc
- *WS-ResourceProperty specification:* http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceProperties-1.2-draft-04.pdf
- *Web Services Notification specifications:* www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsn

Listing 1. MUWS Description of Manageability Capabilities

```
<!-- fragment from MUWS.wsdl -->
<definitions>
  ...
  <portType name="Identity"
 wsrp:ResourceProperties="muws-
xs:IdentityProperties"/>

  <portType name="ResourceState"
 wsrp:ResourceProperties="muws-
xs:ResourceStateProperties">
    <operation name="Start">
      <input name="StartRequest"
   message="muws-wsdl:StartRequest"/>
      <output name="StartResponse"
   message="muws-wsdl:StartResponse"/>
    </operation>
    <operation name="Stop">
      <input name="StopRequest"
   message="muws-wsdl:StopRequest"/>
      <output name="StopResponse"
   message="muws-wsdl:StopResponse"/>
    </operation>
  </portType>

  <portType name="Metrics"
 wsrp:ResourceProperties="muws-
xs:MetricsProperties">
    <operation name="ResetAll">
      <input name="ResetAllRequest"
message="muws-wsdl:ResetAllRequest"/>
      <output name="ResetAllResponse"
message="muws-wsdl:ResetAllResponse"/>
    </operation>
  </portType>
</definitions>
```


About the Authors

Pankaj Kumar is a software architect within the Management Software Organization of Hewlett-Packard. He is responsible for management solutions based on Web Services Distributed Management (WSDM) and associated enablement efforts with partners. Pankaj is the author of the book *J2EE Security for Servlets, EJBs and Web Services*.

pankaj.kumar@hp.com

Chris Peltz is a software technologist at HP, helping to drive the adoption of emerging standards, technologies, and architectures for the Adaptive Enterprise. Chris has written a number of technical articles on application management solutions, and has done extensive research in the areas of Web services and service-oriented architectures.

chris.peltz@hp.com

Everything OLD Is NEW Again!

Imagine **XML-enabling** your existing **PowerBuilder** and **Visual Basic** applications

QUICK In hours, without changing a line of code

CHEAP Inexpensively, using your current skills

SAFE Safely, preserving all logic and security

NOW YOU CAN

www.active-endpoints.com

Web Services: Monitoring and Management for Reliability

Do you need them?

Organizations looking to reduce integration costs are increasingly adopting a service-oriented architecture (SOA) to maximize their IT investment.

WRITTEN BY
RAJIV TOTLANI

The preeminence of Web services as a tool that can support a wide range of dynamic business processes has made it the SOA tool of choice. Web services are easy to build but difficult and expensive to maintain. Monitoring and management costs weigh heavily on the ROI calculator, and in order to maximize ROI enterprises need to keep a keen eye on the support and reliability meter.

In the Web services world, an application is typically a chain of services, or "links," woven together in some sequence with a Web services front end. The chain itself is weaker than the weakest link in the chain. For example, if an application consists of three service calls, each with a reliability of 0.99, 0.96, 0.97 respectively, the overall application reliability according to the laws of probability and statistics is

```
Application Reliability =
0.99*0.96*0.97=0.92
```

The multiplicative effect of individual services tends to steeply reduce overall application reliability as the number of links in the chain increases.

Some of the biggest strengths of Web services, the HTTP and SOAP protocols, are also its weaknesses. HTTP is a stateless protocol that does not guarantee delivery of all the packets to the destination. Nor does it guarantee the order of the arriving packets. This makes HTTP an unreliable protocol incapable of meeting the delivery requirement of "Exactly once". If there is no bandwidth, the packages are discarded. SOAP is the wire protocol for Web services and has some inherent performance problems. Extracting the SOAP body from the SOAP envelope is time-consuming. Parsing megabytes of XML data with a lot of type information is slow and intensive. To increase the reliability of Web services and measure up to the more mature and robust middleware messaging standards, we need to fortify the managing and monitoring of Web services and enhance the reliability of the underlying protocols.

Monitoring and management are the two pillars of reliability. They are related in that the overall goal is to ensure that the QoS objectives are met. Monitoring is a "fault detection" mechanism that checks the health of a service in real time and tries to reduce application downtimes by detecting signs of failure. It ensures that the service is available, accessible, and capable of meeting the throughput and latency requirements. Management is a "fault avoidance" mechanism that lays down rules and policies that makes the service more reliable, usable, and robust. Management ensures that the services can be deployed in a consistent manner, configured from an easy-to-use user interface and meet the overall security and auditing requirements. Within an IT department, usually different groups are responsible for these two functions, so a degree of separation between them is desirable.

Monitoring: The Pulse of Web Services

Monitoring is essential to ensure the required QoS (Quality of Service). It tracks availability, accessibility, and performance of the Web service.

- ***Availability:*** Availability determines if the Web service is up and running. It can be determined by some sort of a "ping" mechanism that periodically executes a dummy request or some kind of a "push" mechanism built into the service that periodically generates heartbeats that can be monitored. Asynchronous push mechanisms work better in general as the system can be designed to perform a "health check" before publishing the heartbeat.
- ***Accessibility:*** Just because a Web service is "available" does not mean it is "accessible." The lack of accessibility may be due to reasons like an insufficient number of worker threads to handle the request under high load conditions, unavailable dependencies like a database, or other callable services. The "ping" mechanism works better to determine system accessibility. If the system is designed to perform a full periodic diagnostic of all system resources and dependencies, a push mechanism based on heartbeats may work, but the push mechanism cannot account for unforeseen exigencies.
- ***Performance:*** Performance profiles the execution of a Web service call and provides operational statistics. Its numbers measure both throughput and latency. Throughput measures the extent of usage of the Web service and determines scalability requirements. Latency is a measure of the round-trip time and can help identify bottleneck subcomponents or resources.

A Web service must be "coded" for moni-

Fix them fast.

The SIFT™ solution framework from Service Integrity™ is the leading monitoring and analysis software for XML and Web services applications. SIFT will allow you to rapidly identify and resolve performance issues in your distributed applications. SIFT installs in minutes and automatically discovers your Web services and provides complete visibility into HTTP and XML data flows — all without a single line of custom coding. SIFT will untangle your problems fast.

IT Applications of the SIFT™ Software:

- Intelligent Logging to save Time/Expense of Custom Coding
- Deep Visibility into Application Health and Security Status
- Improved Policy and SLA Compliance
- Fully configurable alerting
- Support for Billing and Metering
- And more...

"SIFT has been an invaluable tool and I don't know how we'd live without it. We've been able to monitor system performance and isolate a few bugs that had gone undetected in the past."

Director of Operations at one of the largest commercial banks in the United States

Service integrity

Visit www.serviceintegrity.com for more information and enter to win a FREE COPY of *Securing Web Services with WS-Security*, by Dr. Jothy Rosenberg, CTO, Service Integrity, Inc. Fill out our registration form and you will be automatically entered to win!

toring during development if the Web service development toolkit does not support these monitoring options. Nearly all the big Web service providers – like IBM, TIBCO, BEA, and Microsoft – either have built-in support for availability and performance or are planning it in the next release.

Management: The Nerve of Web Services

Managing Web services is a much more involved activity than monitoring. It deals with the following tasks:

- ***Deployment:*** Manages a multitude of Web services from a centralized console in a consistent manner throughout the enterprise. Managing deployment includes the task of configuring the service, deploying the service to a server, and displaying the status of all the services on all the servers.
- ***Versioning:*** Ensures backward compatibility by ensuring that the older versions of client requests are served by the older versions of the service instance. It allows rollout of newer tested versions to a limited user group before a full-blown release, reducing the overall risk of exposure to a new version.
- ***Security:*** Deals with encryption and decryption of messages and authentication and authorization of the Web service clients. Authentication and authorization typically involve some sort of identity management as well.
- ***Scalability:*** The ability of a system to meet performance requirements by optimizing its use of software and hardware resources. Managing scalability can be extremely complex and typically requires policies that look at the execution profile and determine if the throughput and latency requirements are being met and issue an alert if the performance metrics are not being met.
- ***Logging and auditing:*** Trace the life cycle of the Web service call. Logging and auditing require disk I/O and are expensive tasks. Web services should be able to perform role-based logging "on-demand" or "on-error". "On-demand" logging is the ability to turn logging on or off from a management console without the need to restart the service. "On-error" logging is a feature by which the application logs only the errors in a very descriptive mode.

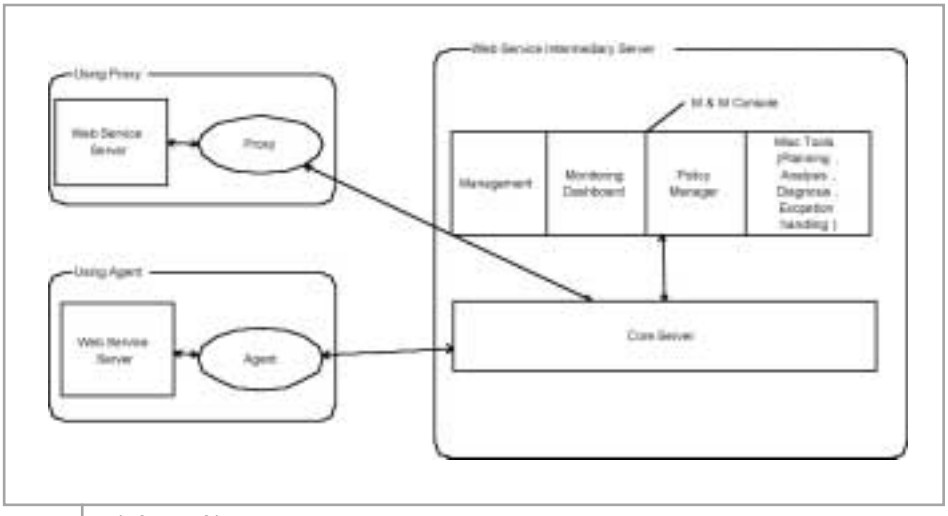


FIGURE 1 | **Typical WSI architecture**

Web services need to be "instrumented" for management if the service provider does not support some of the vital management functions. The management objective is the one in which the Web service providers need to catch up. Most support deployment from a centralized console. Some of them support versioning directly; most support it indirectly through namespaces. None of them support the entire spectrum of security requirements and most of them do not support management policies. This has led to the growth of a unique class of middleware in the Web services world called Web Services Intermediaries (WSI). These intermediaries (like Oblix, Actional, AmberPoint and Ensemble) specialize in monitoring and management activities.

> "WSDM and WS Security are the two key protocols that will influence how we manage Web services in future"

Management Brokers

WSIs (Web Service Intermediaries) provide all the features required for management and monitoring of Web services. There are a number of different tools out in the market. The tools differ significantly in how they approach the issue of monitoring and management but they all work in a similar manner to achieve their purpose. In spite of seeming differences, all the WSI products share some common design features. They all consist of an administrative GUI that serves as the monitoring and management console. Management is usually by way of policies that define which management tasks need to be performed and whether they need to be performed conditionally. The actual task of monitoring and management is achieved by either agents embedded in the Web services server or a proxy outside the server that interacts with the server in lieu of the client. In the proxy-type architecture, one endpoint of the intermediary is configured as a proxy for the provider and the other endpoint is defined for the consumer and a pipeline of activities are performed whenever a message is moved from the consumer to the provider (see Figure 1). The major players in the WSI world are Oblix, Actional, AmberPoint, and Infravio.

- Oblix offers COREsv as a comprehensive man-

agement product for Web services. The key components of COREsv are a policy manager, policy gateways, policy agents, and a policy dashboard. The policy manager of COREsv extends the identity management process of user-to-application interaction to application-to-application interaction and provides graphical tools to build security in these interactions. Policy gateways act as proxies that intercept inbound requests and implement policy steps in a non-intrusive manner. The policy agents plug directly into a Web service providers toolkit providing a more fine grained control over an application's operations than a gateway. The monitoring dashboard is a graphical tool that collects and displays data from the agents and gateways and executes rules that notify or correct problems based on the data. Oblix provides a custom built agent for TIBCO's Web service tool, BusinessWorks.

- Actional's Web services management product is "Looking Glass." Its essential components are an enterprise management console, an SOA planner (for analysis and planning) a customizable portal for service and SLA monitoring, and a management and policy server. Looking Glass provides "Service Stabilizers" that can be configured to define compensatory actions in case of failures so that the anticipated or known problems can be automatically handled. A root-cause analysis mechanism in Looking Glass provides a more comprehensive insight into failures and obviates the need to mine log files. Looking Glass excels in both monitoring and management.
- AmberPoint's AmberPoint Express provides logging, auditing, testing, exceptional handling, and a security mechanism for managing the Web services. Defining service-level objectives is the key to configuring the Web services for monitoring and management by Express. Analogous to the service stabilizers in Looking Glass, the product provides the capability to define compensatory transactions for service level agreement violations.
- Ensemble by Infravio uses contracts as the metaphor for defining relationships between the SOA provider and consumer. All the monitoring and management actions in the pipeline are terms in the contract. The product provides most of the management functions like logging, reporting, load balancing, versioning, fail over, authentication, and authorization. Ensemble is a management-heavy tool.

It is expected that some of the WSIs will be overtaken by the Web service development toolkit providers like IBM, TIBCO, BEA, and Microsoft in the future as these vendors reinforce their offering of Web services development tools with a management and monitoring stack.

Standards to the Rescue

A number of standards are emerging aimed at improving the manageability and reliability of Web services. These standards address some of the problems intrinsic to the HTTP protocol that make Web services unreliable.

- ***HTTP Reliable (HTTPR):*** An extension of the HTTP/1.1 protocol that offers reliable delivery of HTTP packets. Reliable delivery is "Exactly once" delivery of a message from the client to the server. If the delivery fails the protocol handles the retransmission of the message or reports the delivery failure reliably. A persistent storage is required to store the messages for retransmission. The concept of reliable delivery has been widely used with great success in the messaging world by MOM providers such as TIBCO and IBM. The design of the HTTPR protocol is still a work in progress administered by IETF (Internet Engineering Task Force).
- ***WS-Security:*** Ratified by OASIS as a full-fledged standard in April 2004. The specification proposes a standard set of SOAP extensions that can be used to build secure Web services by providing message integrity, confidentiality, and authentication. WS-Security can be used with a wide variety of encryption technologies. It is designed to be extensible.
- ***WS-Reliability:*** A SOAP-based protocol for exchanging SOAP messages with guaranteed delivery, no duplicates and guaranteed message ordering. It is defined as SOAP header extension and is independent of the underlying protocol. The specification contains a binding to HTTP. There can be four different types of reliable message contracts:
  - Guaranteed message delivery or At-Least-Once delivery semantics
  - Guaranteed message duplicate elimination, or At-Most-Once delivery semantics
  - Guaranteed message delivery and duplicate elimination or Exactly-Once delivery semantics
  - Guaranteed message ordering for delivery within a group of messages

  Routing of messages and transactions is not within the scope of WS-Reliability. The ordering of messages is based on a group id and sequence number. The choreography of "exactly one" and "in order" delivery of message proposed by WS-Reliability holds greater promise than the protocol level reliability enhancements of HTTPR.
- ***Web Services Distributed Management (WSDM):*** WSDM is an OASIS effort and the WSDM technical committee is working closely with the Web Services Architecture Group and the Distributed Management Task Force (DMTF) to incorporate the management work completed by these committees and develop the model of a Web service as a manageable resource.

WSDM and WS-Security are the two key protocols that will influence how we manage Web services in the future. WS Reliability is an effort to boost the reliability of the Web service by defining the QoS in the service call. The underlying protocol should be able to handle the QoS reliability requirements.

Build or Buy?

Chances are that most organizations won't need a comprehensive management solution to Web services. If your deployment hits critical mass (more than 20–30 Web services), you need to evaluate using a monitoring and management solution. The choice is between supplementing what the development toolkits offer with a planned and consistent code base that lays the foundation for management and monitoring or using one of the WS's. Using WSIs can cause some performance degradation in high volume, small message size situations. With larger message sizes, the compression feature offered by WSIs reduces the latency. The graphical nature of WSIs, drag-and-drop tools, and intuitive thinking to configuring contracts or service levels makes the task of monitoring and management easier.

A number of technical committees have been formed to address the deficiencies in monitoring and managing Web services. The future looks much better, especially with the Web service providers planning support for what the technical committees recommend. The monitoring and management services that a Web service provider will offer will serve to differentiate its product from competitors' products.

About the Author

Rajiv Totlani is an enterprise integration architect with TIBCO Software. He has designed EAI systems using TIBCO's Messaging, Web Services and J2EE Connector architecture for many of TIBCO's Fortune 500 clients. Prior to joining TIBCO, he worked for SABRE in the Airline Software Solutions group where he was responsible for managing their Day-Of-Operations software products.

rtotlani@tibco.com

Service Archaeologist

The hot job for 2005

Want to leverage your enterprise's Web services? Chances are you'll be enabling or exposing existing application services and not building new. This should come as no surprise to anyone. However, while we've been focusing on the development of new services, how to do it, and what tools to use, most of the work that I see coming is learning how to translate and expose legacy services. So, my prediction is that one of the best paid jobs in 2005 will be engineers specializing in Web services enablement of legacy systems. I'll call them service archaeologists.

WRITTEN BY
DAVID S. LINTHICUM

Indeed, most corporate data and business processing resides on mainframe computers; any attempt to move to an SOA has to include legacy/mainframe integration. Mainframes are not the mainframes of yesteryear. The good news is that most mainframes today are designed to work and play well with other systems within your IT infrastructure, and offer many points of integration, both at the information and the service levels. The bad news is that traditional mainframe-based applications were not designed to communicate outside of the processing space, and thus with some types of legacy systems Web service enablement is an unnatural act. However, there are always mechanisms to provide integration, it's just a matter of understanding what your requirements are, and selecting the correct technology.

Common Issues for "The Dig"

Legacy systems, such as mainframes, have three common issues that you must take into consideration: unstructured data, batch-oriented, binding of logic and data.

Mainframe information has a tendency to be unstructured, meaning that the information is contained in flat files or index sequential files, and not a database. This is challenging to those looking to expose that data into their SOA since a lot has to be understood pertaining to how the data is stored and retrieved. While this can make integration difficult, it's not impossible. In some instances changes must be made to mainframe programs to externalized unstructured data, and in some cases you can account for the unstructured data within the middleware layer through data abstraction facilities.

The fact that many legacy/mainframe systems are batch-oriented presents its own set of problems and opportunities. SOAs typically don't deal with batch systems; information moves between systems record by record, service by service. So, in many instances changes must be made to internal processing on the mainframe to support a more granular data exposure mechanism. However, in some cases it may make better sense to design your SOA in such a way as to provide the ability to accept and produce large amounts of batch data. Services can be written to accomplish this task.

Finally, many mainframe-based information systems do not have a clear separation of data and logic, such as those that leverage file-based information (e.g., ISAM/Cobol). This is more of a challenge to understand than integration. Indeed you must understand how the code is designed to access the information, and then make a decision to expose the mainframe code as a service, sometimes requiring changes to the code. Or, access the information on its own, if possible, binding the data to services outside of the legacy system.

Mainframe Access Mechanisms

Your choices for enabling technology to access mainframe-based systems are numerous. Typically your hardware vendor will assist you in selecting the proper integration technology for your requirements, but you need to understand all of the potential solutions available in the marketplace which are becoming numerous.

To access mainframe-based processes you may use Logical Unit 6.2 (LU6.2). This is IBM's device-independent, process-to-process protocol that provides the facilities for peer-to-peer communications between two programs and also supports asynchronous networking. This mechanism allows you to leverage an internal mainframe process, and if needed, expose it as a service using the LU6.2 as the access point, "wrapping" it as a Web service. You can build services on top of LU6.2 yourself, or leverage a middleware layer that manages the internal process to Web Services translation for you (using LU6., or other interfaces).

For instance, as I'm writing this column Cape Clear Software and NEON Systems, Inc., announced that the two companies are working together to enable organizations to simplify the integration of their mainframe applications and data using Web Services and service-oriented architecture (SOA). The agreement includes tight integration between Cape Clear's Enterprise Service Bus (ESB) and NEON Systems' Shadow z/Services product, allowing organizations to integrate their mainframe applications and to provide those applications as services to the rest of the network.

To expose mainframe data as services to your SOA, you may leverage database gateways. These are also known as SQL gateways and are APIs that use a single interface to provide access to most databases that reside on many different types of platforms. They are similar to virtual database middleware products, providing developers with access to any number of databases residing in environments that are typically difficult to access, such as a mainframe. For example,

Gartner

Application Integration & Web Services Summit

November 15 – 17, 2004 Orlando, Florida

gartner.com/us/aiws

FREE to every attendee:

Gartner 2004 Guide to Application Integration and Web Services Products and Vendors

Connecting the Dots

Key Ideas. Hot Topics.

- Application Integration
- Web Services
- Business Process Management (BPM)
- Event Processing
- Enterprise Information Integration
- Standards
- Service-Oriented Architecture and Applications
- Enterprise Architecture
- Portals
- Measuring and Improving ROI
- Security issues

and much more!

Event Features.

- 45 in-depth sessions led by 30 leading experts
- Real-world case studies
- The latest unbiased research and best practices
- Solutions from over 50 leading vendors

PLUS!

Special CIO Day

BPM 1/2 day Seminar

KEYNOTE and GUEST SPEAKERS

Dr. Mani Chandy

Professor of Computer Science, California Institute of Technology

Tom Davenport

President's Distinguished Professor of Information Technology and Management, Babson College

Peter Weill

Director of Center for Information Systems Research, MIT

Roy Schulte

Vice President and Distinguished Analyst, Gartner

David Smith

Vice President and Research Fellow, Gartner

For additional details and to register, visit: gartner.com/us/aiws

For sponsorship details, contact david.mcmahon@gartner.com or call +1 203 316 6603

Priority Code: WSJ2004

> "The good news is that most mainframes today are designed to work and play well with other systems"

using an ODBC, JDBC, or Web services interface and a database gateway, developers can access data that resides in a DB2 database on a mainframe, or in an Oracle database running on a minicomputer, and in a Sybase database running on a UNIX server. The developer simply makes an API call, and the database gateway does all the work, including distributing the query and assembling the data, perhaps exposed as a single virtual schema or view.

Database gateways translate the SQL calls into a standard format known as the Format and Protocol (FAP), the common connection between the client and the server. FAP is also the common link between very different databases and platforms. The gateway can translate the API call directly into FAP, moving the request to the target database, and translating the request so that the target database and platform can react.

Clearly, the use of legacy systems will continue, and as we move quickly to SOA, so will the enablement of these systems. The trick is to identify the services that are important to the enterprise now and create a plan for exposing those services including which enabling technology and standards to apply. Once you've done that prepare yourself for some complexity. None of these systems were designed to expose services, and it's not going to be an easy job no matter how good the translation and wrapping technology is. It could, in most cases, mean that you need to change and test code over 20 years old. Oh well, that's why service archaeologists will be making the big bucks.

About the Author

Dave Linthicum is the CTO of Grand Central Communications (www.grandcentral.com) and has held key technology management roles with a number of organizations including CTO of both Mercator and SAGA Software. David has authored or co-authored 10 books, including the groundbreaking and best-selling *Enterprise Application Integration* released in 1998. His latest book, *Next Generation Application Integration, From Simple Information to Web Services* was just released.

e-mail: linthicum@att.net

WebAppCabaret™

J2EE Web Hosting

http://www.webappcabaret.com/wsj.jsp

1.866.256.7973

Quality Web Hosting at a reasonable price...

<JAVA J2EE HOSTING AND OUTSOURCING>

JAVA Developers: Design and Architect with Struts. Develop with Eclipse.Build with Ant. Deploy to WebAppCabaret.
It is that easy with a hosting company that understands the Java J2EE standard. Web Services - a buzz word or reality.
How many web hosts provide the facilities to deploy and run applications written for Web Services? We Do.
At WebAppCabaret, we lead the way in providing comprehensive Java J2EE and PHP/Perl Web Hosting solutions.

Easy Application Deployment is a reality at WebAppCabaret. Each Account's J2EE Application Server is installed with its own instance and default settings so you have complete control. Our Web Control Panel makes it a breeze for JVM restarts. Such is an example of our Advanced Web Hosting Infrastructure and Tools. If you are a consultant, do you have a complex web hosting requirement for your client? Web Host or Reseller, are you looking to provide services without the headaches of managing your own network infrastructure? Look no more.

For JAVA J2EE we offer the latest versions of Tomcat, JBoss, and Jetty Application Servers. For Scripting programming we offer the latest PHP and Perl. We also offer the latest MySql and PostgreSql Databases. Commercial software, including Resin, JRun, and Oracle, is also available for an extra charge. In addition we provide valuable tools such as Bugzilla Bug Tracking, CVS Version Control plus a whole lot more. Does your service provider offer ENCRYPTED ACCESS for Email (POP3S/SMTPS/IMAPS), Shell, and FTP? WE DO. All of this is backed by our Tier 1 Data Center with 100% Network Uptime Guarantee.

Below is a partial price list of our standard hosting plans. (Reseller accounts also available). For more details please log on to http://www.webappcabaret.com/wsj.jsp or give us a call at 1(866)-256-7973.

$39/mo Enterprise

Latest JBoss/Tomcat/Jetty
Latest JSP/Servlets/EJBs
Private JVM
Choice of latest JDKs
Dedicated IP Address
NGASI Control Panel
PHP and Perl
Web Stats
1GB Disk
200MB DB
MySql
PostgreSql
Dedicated Apache
Telnet . SSH . FTP
5 Domains
100 Emails
Web Mail . POP . IMAP
more...

Options

The following are some of the add-on options with the associated extra starting at costs:

Resin: $10/mo

300MB Oracle: $40/mo

JRun: $20/mo

ColdFusion: $30/mo

$191/mo Dedicated

Managed Dedicated Server starts at $191 per month for:

Our J2EE-Ready dedicated servers make deploying complex applications a breeze.

NGASI Control Panel with your own Logo
Each dedicated server configured for standalone web application and web hosting (resellers) at no extra charge.
more...

$3000/mo 4Balance

4Balance is our entry level High Availability Load Balancing service comprised of:
1 Database Server
4GB RAM
Dual Xeon
2Application Servers
2GB RAM
Single Xeon
1 Load Balancer
1GB RAM
Single Xeon

VPN

Need to host a site outside your corporate network with secure access? Our VPN service is the solution for peace of mind.

$99/mo Reseller

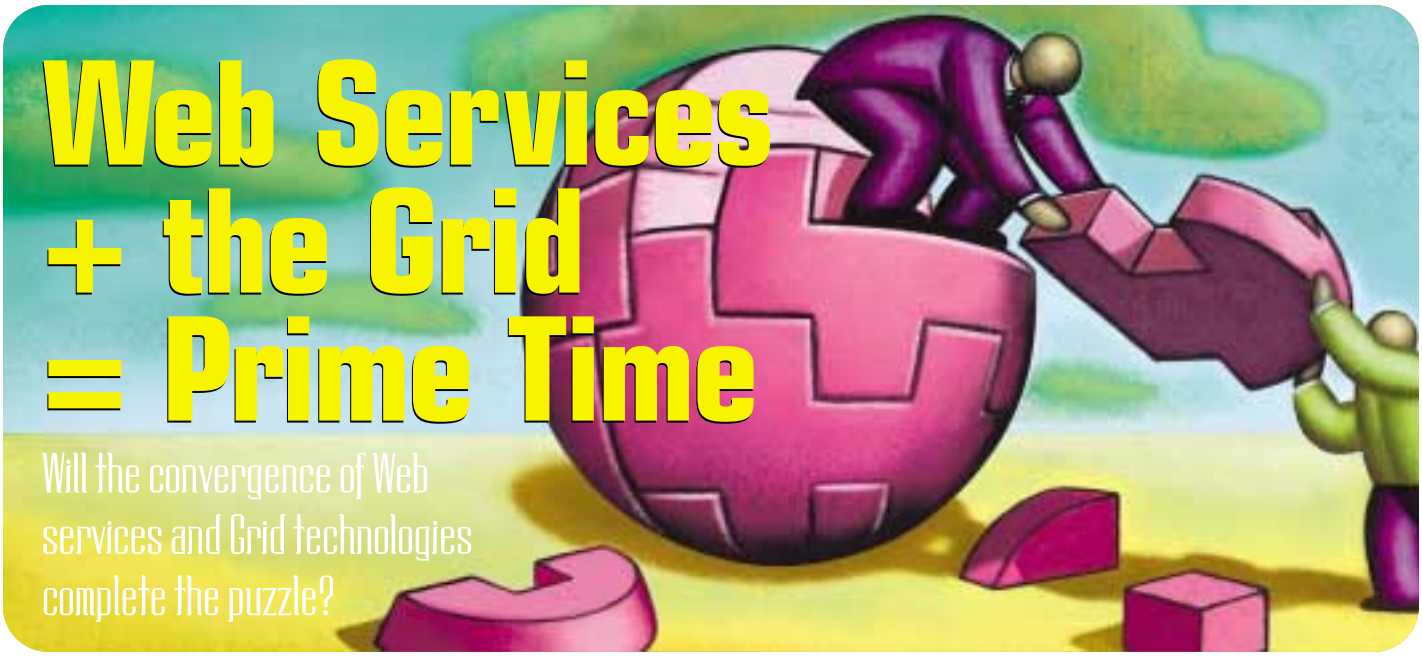
If you cannot afford a Dedicated server for reselling web hosting, for $99/mo the Shared Reseller plan gives you:

10 Professional Plans Your Web Site
NGASI Control Panel with your own Logo
50% Discount
No System Admin
Easy Account Mgt

more...

Prices, Plans, and Terms subject to change without notice. Please log on to our website for the latest Prices and Terms.

Web Services + the Grid = Prime Time



Will the convergence of Web services and Grid technologies complete the puzzle?

Web services and the Grid are converging! The prospect of grid-based, commodity computers delivering run anywhere, anytime Web services across the Internet has hype-o-meters showing a speedy rise and marketing departments gearing up everywhere. Standards are still winding their way through community processes and early adopter products are just coming to market, but that hasn't stopped some industry watchers from proclaiming "Grid services" the next big thing. The Butler Group, for example, sees the coming boom in Grid services dwarfing even the Internet in terms of its impact, as they transform IT from a products-based to a services-oriented industry. Irrational exuberance? Maybe not. The coupling of Web services' strong standards heritage with the Grid's experience securing and managing resources in heterogeneous environments could create a boon for SOA, B2B and eCommerce. The answer ultimately lies in the Web services community's ability to turn the Grid's strengths into standards, reference implementations, and products as part of the convergence.

Ben Worthen, in his September 2002 article for *CIO* magazine, titled "Web Services Still Not Ready for Prime Time," singled out security and reliability as two major hurdles Web services needed to overcome before being ready for prime time. He saw the lack of standard security protocols as an issue for all but a few firms wanting to use Web services on the Internet, and the absence of an "open-line, telephone-like connection" derailing B2B and e-Commerce applications by undermining the reliability of machine-to-machine transactions. Two years, dozens of standards later, and the problems remain. Can the Grid's influence change the equation?

WRITTEN BY
RICKLAND HOLLAR

Addressing the Security Problem

Security on the Internet, by its very nature, is problematic. The Internet operates as a confederation of independent domains where each domain potentially has its own domain authority and security policies. Figure 1 illustrates some of the problems this creates for Web services applications. In the example, a user interacts with a Web service, Service A, running in security Domain A. Domain A uses X.509 certificates for authentication and authorization. Service A uses a second service, Service B, running in security Domain B, that uses Kerberos. Service B, in turn, uses Service C, running in security Domain C, that uses user-ids and passwords. Reconciling the user's identity, privileges, and credentials across the domains is the problem.

SAML and XACML are approaching this problem by abstracting the differences out of the services themselves and into Policy Decision Points (PDP). This decreases the number of nodes involved and introduces a common mechanism, assertions, for bridging domains. SAML and XACML are relatively new standards and implementing products are just beginning to emerge. That leaves a large "legacy" base and long transition period unaddressed. The interim solution is to integrate existing security mechanisms in a coherent way. For the foreseeable future, that means bridges between domains and proxy credentials representing real credentials in situations where there is

no common credential mechanism or authority.

In my article "Who's Master of Your Domain? Web services security in an unfriendly world," (***WSJ***, Vol. 4, issue 6), I discussed how Web services standards (WS-Policy, WS-Security, and WS-Trust) are now in place for implementing basic security services, such as authentication, authorization, confidentiality, and integrity, in Web services environments. The standards provide the grammar for communicating security policies and the flexibility necessary to support any of the traditional security strategies. They also support SAML and XACML. As you would expect, the standards specify "how" to convey security information, but not "how" to implement any specific service or bridge domains. They leave that to implementers.

The Grid's experience operating in heterogeneous environments goes directly to solving the bridging problem. The Grid Security Infrastructure (GSI), built on the Transport Layer Security (TLS) protocol and the Internet X.509 Infrastructure, creates a basic security framework that does not require a centralized certificate management authority. The Grid community has successfully worked out the problems of integrating non-X.509 based authentication and authorization mechanisms into this infrastructure by creating bridges and methods for global-to-local identity mapping, proxying credentials, and delegating privileges. Some refactoring is necessary, but the Grid community's work provides the roadmap needed to advance the discussion within the Web services community beyond abstract models to concrete reference implementations and products. The Grid's experience could cut years off product introduction cycles and provide bridging solutions between both traditional approaches and between those approaches and next generation, SAML and XACML-based products.

Addressing the Reliability Problem

The Internet poses even greater reliability challenges – particularly for Web services integrating components, belonging to multiple owners, using the Internet as the backbone.

When you pick up a phone, you get a dial tone. The telephone is always on; it's always available. That is the level of reliability Ben

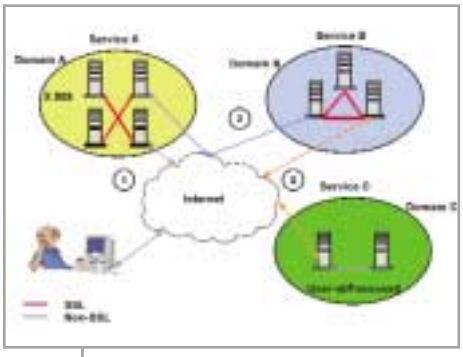


FIGURE 1 | **The problems**

"ility"	Definition
Reliability	a measure of a Web Service's ability to produce a result. The time-between-failures (TBF) is a common reliability metric — the higher the number, the more reliable the service
Availability	the degree to which a Web Service suffers degradation or outages due to failures. A probability measure reflecting the percentage of time the service is available for handling requests — the higher the probability, the better
Serviceability	the ease with which corrective or preventive maintenance can be performed. Time-to-repair (TTR) is one serviceability measure — the smaller the value, the better
Accessibility	the degree to which a Web Service is capable of servicing a request. The probability the service will successfully handle any given request at any point in time is one way to measure accessibility
Scalability	the ease with which the Web Service's accessibility can be improved by adding resources to meet increasing demand — high scalability is desirable
Integrity	a measure of how well a Web Service maintains the correctness of its interactions
Extensibility	an indicator of how easy it is to extend a Web Service, allowing for customization, while retaining complete or partial compatibility with the original service — high extensibility is desirable
Manageability	a reflection of how easy it is to diagnose and resolve capacity and resource problems and fault conditions — high manageability is also highly desirable
Maintainability	how easy it is to modify a Web Service to correct faults, improve performance, and respond to new requirements
Performance	how well a Web Service performs in terms of throughput and latency. Throughput is a measure of how many requests the Web Service can service within a given period of time.

TABLE 1 | **Quality of Service Metrics**

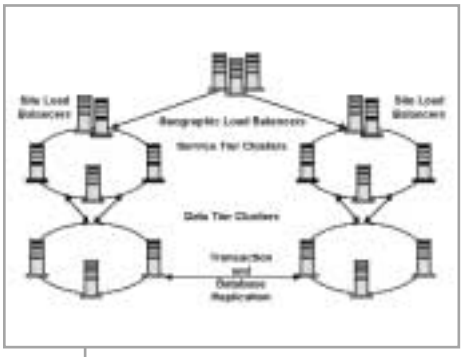


FIGURE 2 | **Example problem**

Worthen was talking about when he said there isn't an "open-line, telephone-like connection" for Web services. When Worthen used the term reliability, he was actually using it as shorthand for the "ilities" – a broader set of Quality of Service (QoS) metrics characterizing an application (see Table 1 for a more comprehensive list). Like a dial tone, when the "ilities" are present, an application is flexible, stable, reliable, and there when you need it. When they are missing or unevenly implemented, all bets are off.

A "dial-tone"–quality service must be able to handle both scheduled and unscheduled outages. It must also be able to deal with surges in demand both in terms of the number of clients requesting to use the service and in the resources any particular request may need. From a client's perspective, it doesn't really matter whether a service is "down" or simply too busy to respond, it is either there or it isn't. The keys to building "dial-tone"quality services are redundancy, managed environments, and high-quality Web services. To better understand this point, let's look at what it means to build a "dial-tone"–quality service for a simple, self-contained environment, and then extend the discussion to include the Internet.

Redundancy, redundancy, redundancy. The first key to building a "dial-tone"–quality service is including enough resources to eliminate single points of failure and resource bottlenecks. The resources in question include hardware, software, networks, and Web service instances. The level of redundancy needed depends on the level of failover the service must provide, not only during normal operations, but also during peek demand and routine maintenance outages.

The three levels of failover possible are cold, warm, and hot (stateful). Cold failover brings up a new component, from a down or offline status, in response to unexpected demand or an outage. Cold failover is generally only an option at the applications, or Web service, level for "dial-tone" –quality services because of the time needed to bring up system or network components cold. Warm failover keeps spare components or capacity online. With warm failover, a service can gracefully deal with both scheduled and unscheduled outages by either dynamically restarting or switch-

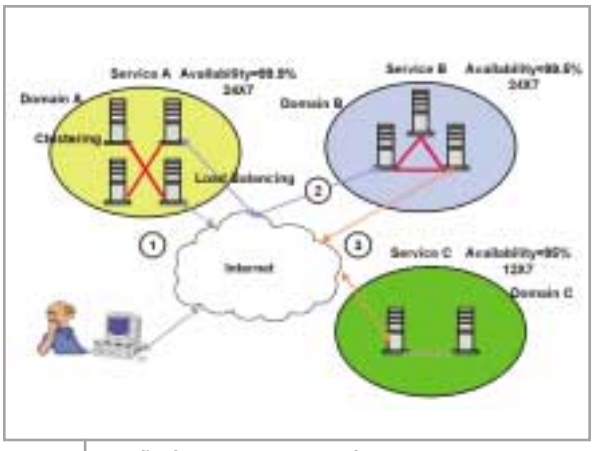


FIGURE 3 **Centralized management approach**

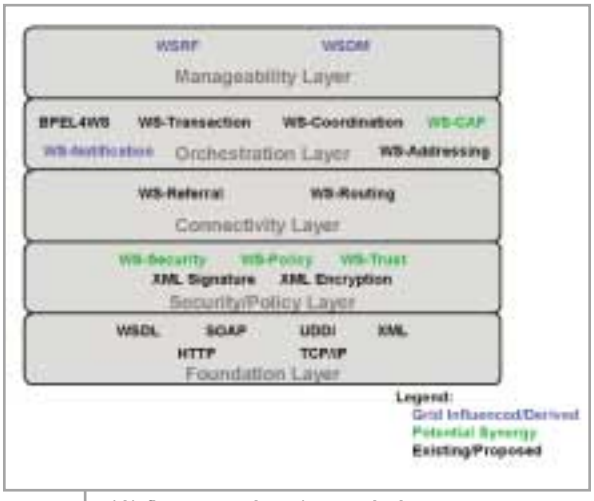


FIGURE 4 **Grid influence on Web services standards**

ing components, and with surge, by distributing load. Stateful failover goes a step further by being able to transparently switch from a failing or degraded component to a backup while maintaining application state.

The second key to achieving "dial-tone"–quality services is managed environments. In a managed environment, management products control different components and levels of the infrastructure, detecting, recovering, and repairing, or at least compensating for, failing or degraded components. Management products include load balancing, clustering engines, and network, systems, and applications management services – which are available from firms such as BEA, Cisco, Computer Associates, IBM, HP, Oracle, Quest, and Veritas. The management products monitor key metrics, resource status, and fault diagnostics, and use redundant resources to maintain consistent levels of service.

The third key is "dial tone"–quality Web services. A "dial tone"–quality Web service is well written, error free, does not contain any unwanted or unknown side effects, and does not request resources beyond those needed, nor hold resources longer than necessary. The service abstracts out lifetime, location, and resource management functions and incorporates interfaces for interacting with management products that provide those capabilities. For stateful failover, the service also maintains the application "context" necessary for "failing over" to another instance.

Figure 2 brings these concepts together into a notional, "dial-tone" quality service. Load balancing provides scalability, clustering provides availability and reliability, transaction and database replication provide failover, geographic separation provides disaster recovery, and server and network redundancy provide the underlying resources necessary to eliminate any single points of failure and to meet surge requirements. The implementation includes not two, but three instances of critical components to ensure there is enough redundancy to guarantee high availability and reliability even when some components are offline for maintenance. Gartner estimates that it costs approximately 3.5 times as much to create such an environment as one for standard applications.

How does the Internet affect this picture? The "ilities" are difficult to build into distributed, Internet-based Web services applications because Web services applications are compositional by nature, i.e., you create larger applications by linking together smaller services or components, and the Internet introduces diversity in terms of the underlying systems, network, and manageability infrastructure. The overall reliability of such a service is a function of the reliability of the individual component services and their supporting and connecting infrastructures.

Figure 2 illustrates this problem. In the example, a user connects to Service A, which is 99.9% available on a 24x7 basis (three nines, by the way, equates to less than 2 minutes downtime per day, and to less than 9 hours downtime per year). Service A connects to Service B, which is only 99.5% available but still 24x7. Service B, in turn, connects to Service C, which is 95% available but only 12x7. Assuming all three services are critical to the application's functionality, the overall availability of Service A is only 49.2% (.999*.995*.95*.5) when you look at it from a 24x7 perspective.

While the challenges are different, the keys to building "dial-tone"–quality services for the Internet remain redundancy, managed environments, and high-quality Web services. The X-factor in this equation is managed environments. Management products geared to the Internet cannot take the centralized management approach shown in Figure 3. Instead, they must take a decentralized, federated approach. This presupposes standards for products communicating with one another, with infrastructure-level resources – such as servers, memory, and network devices – and with application-level services such as Web services. It also presupposes standards for products requesting and providing the metrics, status, and resource information necessary for management products to relocate and dynamically control resources within the environment. Web services is the logical choice for providing these standards, but stateful resources, context maintenance, and standard Web services interfaces for interacting with management products and resources, have, up until this point, been missing. This is where the Grid comes in. Part of the convergence between Web services and the Grid has been getting the "right" interfaces and standards in place to specifically address the manageability question.

Adding Stateful Resources to Web Services

Fault-tolerant and stateful failover solutions assume the ability to interact with stateful services and resources such as servers, disk drives, and memory. Web services is built on a family of stateless protocols: HTTP, SOAP, WSDL, etc. They are stateless in the sense that each client/server interaction is discrete, with no assumption one message relates to another. While higher level protocols such as WS-Coordination,

WS-Transaction, and BPEL4WS provide patterns for implementing stateful message sequences, they also assume stateless interactions at the message level. So, up until recently, none of the existing standards accounted for stateful services or resources.

The Web Services Resource Framework (WSRF) proposal corrects this omission by defining a WS-Resource as a combination of a Web service and a stateful resource expressed as an XML document and a WS-Addressing endpoint. WSRF creates the framework for message exchanges between Web services components and a stateful resource, thereby paving the way for exposing and sharing such resources as Web services and for creating stateful Web services.

Adding Context

Web Services Composite Application Framework (WS-CAF) compliments WSRF by adding the context necessary for managing state aware, compositional applications in stateful environments. WS-CAF introduces the concepts of participants, sharing a common context, and coordinators, orchestrating and managing that context, thereby enabling them to ensure a common outcome across the application.

Adding the Management Product Interfaces

WSRF, Web Services Notification (WS-Notification), and Web Services Distributed Management (WSDM) complete the picture by defining the Web services interfaces and metrics necessary for managing Web services and for integrating Management Products across heterogeneous environments.

WSRF defines the interfaces for managing service lifetimes, properties, and faults. WS-Notification adds a publish and subscribe interface that Management Products can use for overseeing and reporting state and property changes. WSDM builds on WSRF introducing the concept of "manageable resources" and identifying the interfaces necessary for Management Products to interact with those resources.

WSDM is a two-part specification. WSDM Management Using Web Services (MUWS) defines the interfaces for management products to manage resources using Web services messages. WSDM Management of Web services (MOWS) defines the use of MUWS for managing Web services resources to:

- Monitor QoS
- Enforce SLAs
- Control tasks
- Manage resource life cycles

Basic WSDM products can achieve simple fault tolerance; WSDM products acting as WS-CAF coordinators can monitor and manage the context necessary for providing stateful failover.

When will WSDM products be available? They are here today with CA and HP on the leading edge. CA's Unicenter WSDM is one of the first products available, and HP is working closely with the WSDM standards committee, contributing elements of its Web Services Management Framework (WSMF) to the WSDM standard. Smaller companies, such as AmberPoint, Blue Titan, and Infravio, also have products in this space. Bigger players such as BMC, IBM, Microsoft, and Veritas are rapidly following.

Conclusion

As Figure 4 summarizes, the Grid is significantly influencing Web services standards. Web services can finally clear the hurdles Ben Worthen identified by leveraging the Grid's experience in the areas of security and service manageability. The Internet is still a challenge, but even there the necessary pieces are starting to fall into place. Standards, however, are still in their infancy, many are either at Version 1.0 or lower, and some deconfliction still needs to occur. Management products built to the standards are relatively new or still on the drawing boards, meaning widespread adoption is still several years off. It may be 5–10 years before we fully understand the Grid's full impact on Web services, but the convergence is still a thread worth watching. If the Web services and Grid communities leverage their synergy, this is one instance where the reality may live up to the hype and given the – potential, this one could turn out to be the big kahuna.

References

- Worthen, B., "Web Services Still Not Ready for Prime Time", *CIO Magazine*, Sept. 1, 2002. www.cio.com/archive/090102/prime.html
- Foster, I., et.al, "The Anatomy of the Grid." Globus Organization. 2001. www.globus.org/research/papers/anatomy.pdf
- Foster, I., et.al, "The Physiology of the Grid, Globus Organization." June 2002. www.globus.org/research/papers/ogsa.pdf
- Tuecke, S., "Grid Security Infrastructure (GSI) Roadmap." Internet Draft, Global Grid Forum, February 2001. www.gridforum.org/Security/ggf1_2001_03/drafts/draft-ggf-gsi-roadmap-02.doc
- Tuecke, S., et. al. "Open Grid Services Infrastructure (OGSI) Version 1.0." Global Grid Forum, 2003. www.ggf.org/ogsi-wg
- Web Services Resource Framework Version 1.0, March 5, 2004. www-106.ibm.com/developerworks/library/ws-resource/ws-wsrf.pdf
- Bunting, Doug, et. al (2003). "Web Services Composite Application Framework (WS-CAF) Ver 1.0." Arjuna Technologies, Ltd; Fujitsu Limited; IONA Technologies Ltd; Oracle Corporation; and Sun Microsystems, Inc. http://developers.sun.com/techtopics/webservices/wscaf/primer.pdf
* Graham, Steve, et.al. "Web Services Notification (WS-Notification)." International Business Machines Corporation, Sonic Software Corporation, SAP AG, Hewlett-Packacrd Development Company, Akamai Technologies Inc, and Tibco Software Inc., 2003–2004. ifr.sap.com/ws-notification/ws-notification.pdf
- DeCarlo, J, et. al. "Management Using Web Services: Architecture." OASIS, 2003. xml.coverpages.org/WSDM-MUWSArch20031208.pdf
* Dharmawan, A., et. al. "Web Services Distributed Management: Management Using Web Services (MUWS 0.5)." OASIS Open, 2003–2004. www.oasis-open.org/committees/download.php/6234/cd-wsdm-muws-0.5.pdf
* DeCarlo, J, et. al. "Web Services Distributed Management: Management of Web Services (WSDM-MOWS 0.5)." OASIS Open, 2003-2004. www.oasis-open.org/committees/download.php/6201/cd-wsdm-mows-0.5.20040402.pdf

About the Author

Rickland Hollar is a senior applications architect with the Central Intelligence Agency with over 30 years experience in the industry. The views expressed in this article are his own and not necessarily those of the Agency. Prior to joining the CIA, he was president of a Virginia-based software development firm.

rick_hollar@yahoo.com

web services conference & expo EDGE

Web Services Edge 2005 East

International Web Services Conference & Expo

New for 2005:

- **Colocating with LinuxWorld Conference & Expo**
  badges give access to both shows
- **Guaranteed Minimum Attendance 3000 Delegates**
- **New, Hot Sessions & Seminars**

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

Tuesday, February 15
11 a.m.

Matt Ackley
Senior Director, eBay Developers Program

Topic:
Web Services for eCommerce

Keynote Speakers & Featured Guests

Wednesday, February 16
11 a.m.

Ari Bixborn
Director, Web Services Strategies, Microsoft Corporation

Topic:
Indigo and the future of Web Services

Anne Thomas Manes
Burton Group

Application Server Shoot-Out Facilitator

Hynes Convention Center
Boston, MA
February 15-17, 2005

For Exhibit and Sponsorship Information:
Jim Hanchrow, 201-802-3066, jimh@sys-con.com

Register Now! Hot Sessions & Seminars

- Ensuring Web Services Interoperability
- Web Services Standards: Going Behind the Mask
- The Role of Policy in Web Services Integration – It's More Than Just Security
- Four Abilities SOA will lack without a Registry
- Driving SOA Governance
- BPEL Best Practices from Real-World Projects
- SOA: From Pattern to Production
- Lessons From the Front Line – Building Interoperable Web Services
- Developing E-Commerce Applications with Web Services
- Developing Enterprise Class Web Services
- Orchestrating FORCEnet Engagement Packs with BPEL for Web Services
- CPI: A Global Integrated Problem Tracking and Resolution System using Java Web Services
- Using Service-Oriented Architecture and Web Services to Issue Business Licenses in the District of Columbia
- Developing Web Services with Eclipse
- The Interoperability Challenge of Web Services Security Standards
- Securing Web Services with WS-Security
- Anatomy of a Web Services Attack
- Using a Mobile Phone as an SSO Authentication Device in SOA Solutions
- XML Content Attacks
- XACML and Agnostic Authorities

Register Today: www.sys-con/edge2005east/registernew.cfm

Come and see what everyone is talking about...

The Application Server Shootout!

Find out which server best fits your performance and ROI metrics

Join us in delivering the latest, freshest, and most proven Web services solutions...

At the *Fifth Annual Web Services Edge 2005 East – International Conference & Expo* as we bring together IT professionals, developers, policy makers, industry leaders and academics to share information and exchange ideas on technology trends and best practices in secure Web services and related topics.

Exhibit in *The Web Services Pavilion* and see thousands of buyers!

Becoming a Web Services Edge Exhibitor, Sponsor or Partner offers you a unique opportunity to present your organization's message to a targeted audience of Web services professionals. Make your plans now to reach the most qualified software developers, engineers, system architects, analysts, consultants, group leaders, and C-level managers responsible for Web services, initiatives, deployment, development and management at the region's best-known IT business address– The Hynes Convention Center in Boston.

3-Day Conference & Education Program features:

- Daily keynotes from companies building successful and secure Web services
- Daily keynote panels from each technology track
- Over 60 sessions and seminars to choose from
- Daily training programs that will cover Web Service Security, J2EE, and ASP.NET
- FREE full-day tutorials on .NET, J2EE, MX, and WebSphere
- Opening night reception

PRODUCED BY SYS-CON EVENTS

www.sys-con.com/edge

Sponsored by: wldj JDJ WebServices SD Times .NET JOURNAL Information STORAGE+ SECURITY JOURNAL WebSphere XML JOURNAL MX eaIndustry asp.netPRO

sforce.com

The salesforce.com development and integration platform

sforce.com surfaces the services used by the saleforce.com CRM solution to provide an exhaustive development platform used to integrate new functionality and develop applications. While the services provided using industry-standard Web services protocols (XML and WSDL) represent one of the most complete Web service framework implementations to date, sforce.com provides a number of additional tools to streamline the development process by allowing developers to focus on the development of business functionality and enabling them to work productively in the environments they are most comfortable in.

What Is sforce.com?

WRITTEN BY
AHMED SAKO

salesforce.com is the leading hosted enterprise-class Customer Relationship Management (CRM) solution, providing sophisticated CRM functionality (e.g., authentication, role-based access and visibility controls, data and document management, support and management tools for all components of the sales life cycle, collaboration, forecasting, workflow, reports, analytics, integration to office and contact management tools [Act!, Goldmine], etc.) while avoiding the customer ERP/CRM system management headaches by hosting the solution. To avoid the development and integration headaches associated with ERP/CRM systems, salesforce.com provides sforce.com 4.0 a powerful development and integration platform.

The key components of the multi-tiered sforce.com are:

The Studio

A set of wizards and tools customizes the salesforce.com UI (see Figure 1), object/entity model, and workflow without coding or the need for end users to involve IT staff. The studio is used to create new objects, add custom fields to the sforce.com entities (e.g., leads, contracts, etc.), create new relationships between objects (see Figure 2), create new tabs, lay out the contents of tabs, and even modify existing tabs via a flexible layout editor with DHTML drag-and-drop support. Once the modifications are complete the resulting customizations are virtually indistinguishable from the base application (while some limitations exist, for example custom objects cannot be included in the workflow, you cannot create a custom object from the API, etc., most users are unlikely to be affected by these). Also, new User Interface (UI) controls (e.g., Java Applets, ActiveX controls, etc.) can be uploaded to the platform and used for UI customizations. While the wizard approach can seem cumbersome at times, the importance of the decisions being made warrants a deliberate process.

Web Service APIs and related WSDLs

Considering the richness of the functionality that it provides, the API is simple, robust, well thought out, and mature. The API focuses on Create/Read&Query/Update/Delete (CRUD) operations enabling developers to focus on business functionality. It show signs of having been used extensively and includes several high-end features (request throttling, session time-outs, HTTP chunking, response batching and compression, etc.). The WSDL is regenerated from the object model stored on the hosted platform and does not need to be maintained manually.

Two flavors of the WSDL are provided: the Enterprise and the Partner. The Enterprise API is geared toward strongly typed languages such as Java and C# and whereas the Partner API is loosely typed it supports scripting languages such as Perl and Python.

sforce Object Query Language (SOQL) and Object Search Language (SOSL)

SOQL and SOSL are used in conjunction with the API similar to the way SQL is used to access the data held in databases (while the two languages only support queries subsequent API calls can be used to update the retrieved items).

Development Environment Toolkits and Examples

Development toolkits are available to jump start the development effort for a large number of development and deployment environment Frameworks, including Sun's Sun ONE, Microsoft's Visual Studio .NET, Borland JBuilder, BEA WebLogic Workshop, and plain Java. The toolkits typically consist of tools and components to integrate into the development environment and a robust set of examples. The examples are well written and use industry standards and best practices (for example, the Java toolkit uses Apache Axis), making them an excellent development resource.

In addition, the use of industry-standard

Company Info

salesforce.com, inc.
The Landmark @ One Market
Suite 300
San Francisco, CA 94105
United States
E-mail: info@salesforce.com
Main: 415-901-7000
Fax: 415-901-7040
Sales: 1-800-NO-SOFTWARE

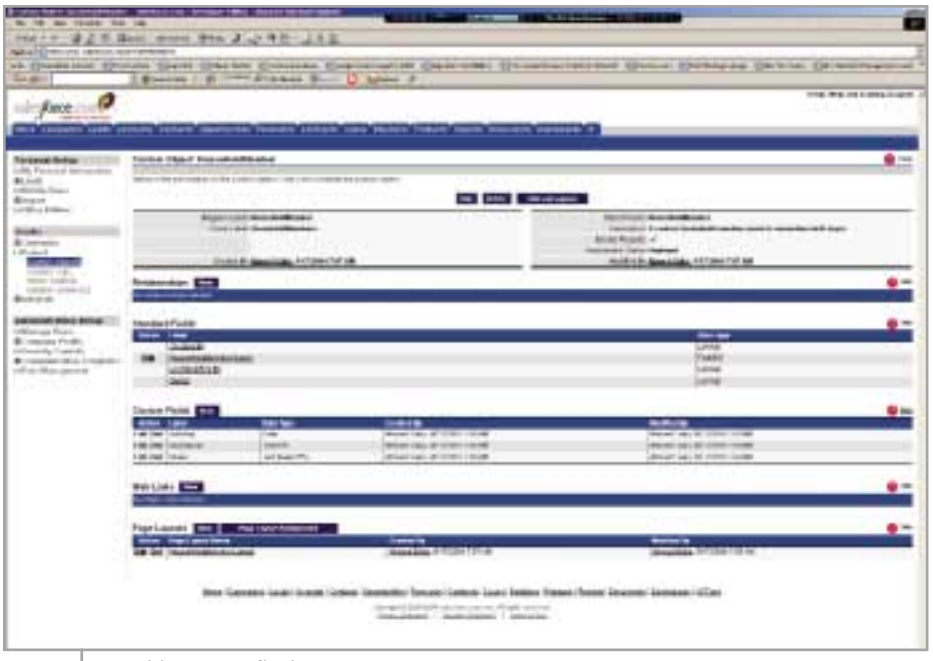


FIGURE 1 **Customizing your application**

FIGURE 2 **Custom object creation (adding a relationship)**

> sforce.com represents a significant step forward for CRM application developers

protocols like XML and WSDL makes it easy to integrate other tools. During this review I easily used a tool from one of my previous reviews (DreamFactory; Vol. 4, issue 9) to create a small application.

This approach is different from that of the leading CRM vendors that tend to direct you toward the use of their proprietary tools (or a single tool) to leverage the bulk of their functionality efficiently.

SourceForge Projects (http://sforce.sf.net/)

A number of open source tools and utilities are targeted, including:

- Explorer: A .NET schema (and object model) browser used to create and test SOQL queries. Extremely handy for development
- A JSP sample built in Eclipse 3.0
- An Excel connector providing bidirectional access to the sforce.com API via the sforce.com Toolkit for Office

The project range of platforms and runtime environments is a reflection of sforce.com's commitment to its development platform agnosticism.

Developer Support, Training and Forums

Extensive documentation and tutorials are available on the sforce.com web site. Online training is also available for free, along with an extensive set of moderated forums where a number of knowledgeable sforce.com resources (e.g. Developer Program Managers, sforce.com Expert Developers, etc.) provide a significant amount of support.

Final Thoughts

sforce.com is a banner ERP/CRM enterprise-class development platform that provides a rich, comprehensive, easy-to-use feature set using a standards-based approach and excellent online documentation. sforce.com represents a significant step forward for CRM application developers.

However, the platform ease of use should not lull you into a false sense of security. As with any complex system, analysis should be performed before undertaking and performing changes, lest you find yourself in a costly "dead-end." Further, despite the ease of use, as with any platform it takes time to learn key idioms and development patterns for the platform and navigate the extensive option set effectively.

About the Author

Ahmed Sako is CTO of an agency-only securities trading firm. His areas of expertise include distributed systems, systems integration, transaction processing, and algorithmic trading.

asako@asako.info

Process-centric Realization of SOA

BPEL moves into the limelight

Agile and adaptive business processes and supporting IT infrastructure are the holy grail of enterprise applications. The industry is heading in the right direction to start delivering on this promise. SOAs (service-oriented architectures) promise to enable businesses to align their business processes to customer needs, and optimize them to improve customer responsiveness and drive efficiency. A process-oriented realization of SOAs is necessary to deliver on this promise.

WRITTEN BY
BY MOHAMAD AFSHAR, HAL HILDERBRAD, NICKOLAOS KAVANTZAS, DAVE SHAFFER, & ASHWINI SURPUR

The process-oriented model is based on an SOA component model augmented with an underlying formal model in which business processes are expressed through orchestration and choreography. This model blends the bottom-up framework of SOA with a top-down, process-centric view. By simplifying the set of activities that are part of the life cycle of business processes, we'll outline how this approach is an effective and efficient way to develop, maintain, and improve best-in-class inter- and intra-enterprise business processes. We'll also emphasize that enterprises looking to automate their business processes around an agile platform can start building, service-enabling, deploying, securing, and managing services – both internal and external – with Business Process Execution Language for Web Services (BPEL4WS or BPEL) business flows today.

The SOA Promise

Traditionally, business information systems have been developed with a functional orientation, often resulting in silos of services and information. However, end-to-end business processes that must span silos can't adapt to change as business needs evolve – they're fragmented and embedded deep within systems. SOA is a software architecture that facilitates the development of enterprise applications as modular and loosely coupled business services.

SOA's intent is to integrate software components, or services, in new runtime contexts and business processes. It creates a unification layer on which business processes and enterprise dashboards can be built. SOA promises to deliver greater responsiveness to business change and provide real-time visibility into business processes. When effectively implemented, it improves business agility by letting you modularize legacy, packaged, and custom applications, and orchestrate them in easily changeable business flows.

The Web services platform, including SOAP and other protocols accessed through Web services binding frameworks, acts as the component model and ubiquitous network fabric through which newly built and existing applications cooperate. Information can be exchanged much more seamlessly, unconstrained by the hardware, operating systems platforms, and programming languages used to implement services.

Whereas a Web services model and SOA often take an essentially bottom-up view, business processes are inherently top-down activities. Bridging the capabilities envisioned by SOA with the requirements of supporting distributed business processes – including trading partner collaborations – is the focus of the process-oriented realization of SOA discussed here.

SOA Realization

The process-centric realization of SOA can provide the glue between modular business services for the end-to-end business processes that companies rely on, such as order-to-cash, procure-to-pay, store order-to-sales compensation, and supplier-invoice to settlement. To complete this picture you must define all forms of business processes, including those of a collaborative nature, and deliver these processes reliably so that they, in turn, can be leveraged by other business processes and services (see Figure 1)

Today, mature standards such as BPEL and Web services effectively enable businesses to implement private processes and collaborative processes based on ad hoc collaboration definitions. The process-centric model of SOA considers processes and collaborations as first-class citizens, much as an object-oriented language

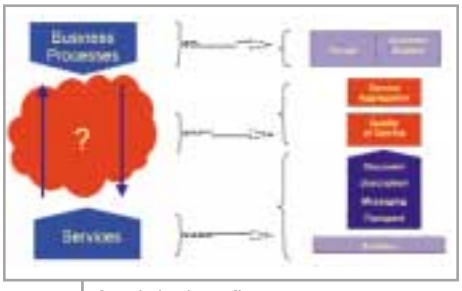


FIGURE 1 **The missing ingredients**

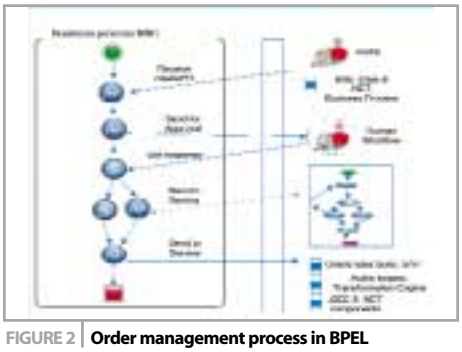


FIGURE 2 **Order management process in BPEL**

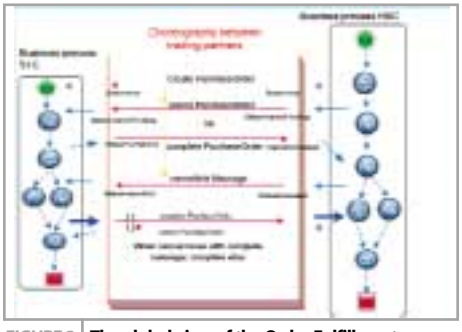


FIGURE 3 **The global view of the Order Fulfillment collaboration**

would consider objects as first-class citizens, and encourages the same maturity of global standards for the definition of B2B processes. With this vision, business processes, collaborations, information exchanges, and work activities are expressed through service aggregation – a combination of BPEL orchestration and WS-CDL choreography, with operational semantics based on the pi-calculus:

- *BPEL orchestration* is concerned with process-flow coordination and implements the behavior of a single participant/ business entity. For example, consider an order-to-cash that implements a business process that handles orders. Orchestration defines the execution logic of application components. Typically, a controller executes a definition of the business process. Standards have tackled process-flow coordination over the years, with the most recent industry consensus occurring around Web services orchestration standards like BPEL. BPEL is used to define the coordination of a process flow as well as related semantics, including parallelism, sophisticated exception handling, data transformation, and transaction compensation facilities.
- *WS-CDL choreography* is concerned with multiparticipant collaborations, such as the Order Fulfillment collaboration between a buyer and a seller. The Web Services Choreography Description Language (WS-CDL) standard is emerging as a way to specify contracts of interoperable business processes collaborating to accomplish a common business goal. Examples of the business protocols that can be easily modeled using WS-CDL are the FIX and TWIST financial protocols and the RosettaNet PIPs. Choreography is built on top of BPEL. BPEL and WS-CDL together can express any business process, including trading partner interactions.

In order to deliver quality of service to enterprise-class business processes a number of supporting services are required. With quality of service, a number of supporting services are required. These include security, reliable messaging, transactions, context and coordination, message translation and transformation services, message validation, and content-based routing. Although these services are vital for the vision of networked-enabled business processes, we'll focus on the service aggregation/business process aspects.

BPEL: Automating Business Processes

In the SOA architecture, and within the Web services platform in particular, services must be combined into business processes/business flows. BPEL provides a standard, portable language for coordinating the flow of business process services and builds upon a decade of progress in the areas of business process management, workflow, and integration technologies. Traditionally, such technologies have suffered from proprietary business processes definitions; the resulting applications are seldom portable across implementations, and skills are not transferable. Since BPEL is a standard, it creates a common language for developers. It's built from the ground up around XML and Web services and is supported on both the .NET and Java platforms.

Asynchrony, parallelism, sophisticated exception handling, long-running processes, and a need for compensating transactions change the fundamental nature of what we think of as an application. In order to support the required executable behavior, BPEL provides constructs for control-flow (conditionals, sequencing, parallelism, loops), variables, and constructs for manipulating them; event handlers, timeouts, exceptions, and forward recovery; and nested scoping units of work.

The process-oriented realization of BPEL also adds the following capabilities on top of standard BPEL:

- Advanced data handling and transformation activities that are used to manipulate variables using XQuery
- Advanced process life-cycle management capabilities such as versioning, Suspend Process instance, and Resume Process instance
- Activities that use a business rules framework, which lets you model business rules in a purely declarative way
- Advanced human workflow activities such as assigning tasks to specific users, sending notifications, and approving requests through mechanisms such as task lists

BPEL provides the basis for implementing enterprise-wide business processes by combining existing business applications, custom applications, packaged applications, non-native services, and partner services into end-to-end business flows.

WS-CDL: Collaborations Between Peer-to-Peer Business Processes

In complex integration scenarios, capturing the collaborations between two or more business processes engaging in peer-to-peer business transactions, potentially across trust boundaries, requires a higher-level model. This is the fundamental thrust of WS-CDL, which is based on a variant of pi-calculus. It enables the formal modeling of the common observable behavior of all collaborating participants from a global viewpoint.

WS-CDL describes the global message exchange of participants during their joint interaction, global reactive rules for declaratively prescribing normal/abnormal progress, common agreement of the outcomes, and a recursive composition model that lets you build choreographies incrementally by combining existing

choreographies. In this way, complex inter-organization business processes can be implemented as a collection of services, which, in turn, implement complex business processes themselves.

Choreography perfectly complements the effort of orchestration and BPEL, offering a global viewpoint into the peer-to-peer interacting business processes that are defined and implemented in BPEL.

To understand how orchestration and choreography come together, let's consider an example.

Case Study: Order Management Business Process

For any business interaction to work, the rules of collaboration must first be agreed upon. These rules can be represented in English. However, since English is unstructured and frequently vague, the rules may not be represented precisely. Choreography, and WS-CDL in particular, lets us write the above rules in a precise and unambiguous manner. So how do orchestration and choreography fit into the larger picture? The case study below demonstrates their complementary role by presenting the example of a localized business process that's concerned with handling purchase orders in a really big corporation (RBC).

This business process leverages several trading partners and links to a range of back-end services and legacy applications into an end-to-end business process. As Figure 2 shows, the RBC process interacts with different applications such as J2EE, .NET, portals, human workflow, ERP financials, trading partners such as subcontractors, and rules engine interfaces.

What makes this process interesting is that it requires both the integration of existing applications and new functionality. In our example, a clerk takes an action that causes a PO to be sent out. This is logged against a custom application (which could be J2EE or .NET) that registers the PO and ensures that the total amount does not violate the employee's sign-off authority level. If management approval is required,
a workflow is initiated.

Upon confirmation, the PO needs to kick off a number of business processes, including one that updates the total POs that have been approved by the employee and manager, and one that checks the availability of goods from an existing warehouse or nearby distribution center and requisitions them if they're not available.
If the goods aren't available at any location, then the system updates the financials application.

Notice that in this case, the overall business process is implemented in BPEL. Each of the services can publish a Web service interface that is then orchestrated in BPEL. Using SOAP, WSDL, XML Schema, and BPEL, the business process can be built in a vendor-independent fashion. RBC's business process is shown in Figure 2.

Using BPEL and other open standards, the task of integrating these systems into an end-to-end business process is straightforward. At any stage, the business process may be changed; for example, by adding extra approval steps, or by updating the BPEL definition of the business process.

Now consider what's needed when the RBC order process interacts with an outside supplier, STC, which manufactures and distributes t-shirts. RBC and STC are engaged in a collaborative fashion to achieve their common business goal: order fulfillment. For the collaboration to work successfully, RBC must provide the terms under which it's willing to do electronic business with suppliers such as STC.

RBC has the following simple business rules for collaborating with STC:

1. RBC places an order for STC t-shirts.
2. STC acknowledges the purchase order. This initiates a business process in STC that handles the PO. For all RBC knows, STC has an employee at a computer handle the PO, or a sophisticated BPEL process that's linked with back-end manufacturing, logistics, and procurement systems.
3. After the internal processes within STC regarding the PO are completed – which may take days or even months – STC sends a Purchase Order Completed message back to RBC.
4. RBC can send a Cancel Order message any time before it receives the Purchase Order Completed message.
5. If the Cancel Order message arrives at STC before the Purchase Order Completed message is sent, STC aborts its business process and acknowledges this to RBC with the Purchase Order CancelAck message.
6. If STC has already sent the Purchase Order Completed message, it ignores the Cancel Order message because RBC has agreed that it will honor POs when cancellations are not sent out within an agreed-upon timeframe.

Based on these business rules of collaboration, the two parties also agree to the following rules of how they exchange messages:

1. When a PurchaseOrder message is sent from the buyer to the seller and the acknowledgement is received by the buyer, the OrderStatus at both parties is set to New.
2. When the OrderStatus at both the buyer and seller is New, either a Cancel Purchase Order Message can be sent by the buyer to the seller or a Complete Purchase Order Message can be sent by the seller to the buyer.
3. When the buyer sends a Cancel Purchase OrderCancel message, the buyer sets itself to CancelPending; when a CancelAck mes-

sage is sent from the seller to the buyer, this sets both parties to OrderCancelled.

4. When the seller sends an OrderCompleted message, the OrderStatus at both parties is set to OrderCompleted.

These terms form the basis of the choreography. The resulting choreography is shown in Figure 3.

Today, these rules are frequently described in English, and each trading partner's business processes that interact collaboratively can be defined in BPEL or in any other language. What WS-CDL offers is a standardized way to define these interactions, leaving unambiguous documentation of the roles and responsibilities of each. Also, in the future, products that support business process implementation languages such as BPEL could provide tools that would generate, for example, an implementation skeleton for a particular role or validate behavior. Developers looking to define collaborations in B2B situations should keep an eye on WS-CDL. Today, however, BPEL provides the basis for automating business processes on both sides of the firewall.

Summary

The maximum value of SOA architecture is realized by blending SOA with a process-centric view that explicitly defines trading partner collaborations. The basic building blocks to do this are already in place and standards like WS-CDL are emerging to support the full life cycle of collaborative business processes.

Enterprises looking to automate their business processes around an agile platform can build, service-enable, deploy, secure, and manage services (both internal and external) with BPEL business flows today.

Looking further down the road, we expect the ways that business-to-business collaborations can be defined will be greatly enhanced through standards such as Web services choreography and we're actively working in these areas. What's clear today is that there is much value to be gained by streamlining your business around SOA with a process-centric view. Go build!

References

- *Regarding BPEL engines and the code examples in this article, see:* http://otn.oracle.com/bpel

About the Authors

Mohamad Afshar, Ph.D., is director of product management for Oracle Application Server 10*g*. His focus includes middleware architecture, SOA, Web services vision, and evangelism. Mohamad holds a Ph.D. in parallel database systems from the University of Cambridge.

mohamad.afshar@oracle.com

Hal Hilderbrad is a Web services architect at Oracle working in the Oracle Application Server core infrastructure group.

hal.hildebrand@oracle.com

Nickolaos Kavantzas is a Web services architect at Oracle working in the Oracle Application Server core infrastructure group. He is the architect of the Web Services Choreography Language and the lead editor of the Web Services Choreography Language in the W3C Choreography Working Group. Currently, he is designing Oracle's Web Services Orchestration/Choreography framework.

nickolas.kavantzas@oracle.com

Dave Shaffer has been helping customers use the Oracle BPEL Process Manager since 2001, managing implementation projects, providing technical training, and ensuring successful implementations. Prior to joining Oracle, he was a principal consultant at Collaxa, a managing director at Eleven Acceleration, and manager of a professional services group at Apple Computer.

david.shaffer@oracle.com

Ashwini Surpur is a principal member, technical staff working on Web Services Choreography at Oracle.

ashwini.surpur@oracle.com

Web Services in Action

Integrating with the eBay marketplace

Web services have served as a topic of interest for IT professionals since at least 2001, and yet there have been few examples of significant, successful deployment. With this in mind, developers and IT decision makers should be aware of the major, business-defining role of eBay's successful Web service deployment. With roughly a billion Web service requests served a month, eBay has demonstrated clear leadership in the real world deployment of Web services.

This article shows how eBay has been able to use Web services to dramatically grow both acquisition and third-party development efforts, transforming a Web site into a Web platform. It will examine the business benefits of Web services for eBay, its partners, and independent developers, and describe eBay's efforts to support this growing community. Finally, it will describe the future direction of eBay's Web service implementation.

WRITTEN BY
WILL IVERSON

Business Advantage of Web Services

How long does it take a business to put a product up for sale? How many people view that product? How long does it take to close the order after placement? Simply put, the eBay Web services platform is an integration engine that enables a business to dramatically improve all of these metrics.

eBay's Web service platform affords a proven, real-time integration opportunity for several different types of business. Most obviously, businesses that have grown beyond an occasional listing can use the eBay Web service platform to dramatically decrease the overall costs associated with managing inventory and working with eBay. This usage of the eBay platform enables integration and automation.

By using eBay Web services, businesses can build integration directly into their existing channel infrastructure for moving products. For example, a business may wish to leverage an existing inventory system to automatically generate eBay auction listings for products that have been sitting idle for ninety days. By integrating with eBay's Web services, an analyst can generate listings for these products with a single click. Similarly, integration with eBay Web services could allow for automatic notification of a successful closed integration with a business's fulfillment center, or trigger a customer satisfaction system 30 days after an auction has closed.

In the integration and automation model, Web services provides a highly efficient mechanism for improving the execution of business processes. Businesses can dramatically reduce the time to list and fulfill auction listings, increasing the rate of return.

This usage of eBay's Web services is in broad deployment today. eBay has found that sellers using Web service–enabled software applications generate an average of 595 new listings and $10,202 in GMS (gross merchandise sales, the value of goods sold) per minute on eBay. In addition, eBay sellers that transition from entering information by hand via the HTML forms to a Web service–enabled selling application typically report a listing volume improvement of over 60% within 6 months.

One additional side benefit of an organization looking to build on the eBay Web services platform – it gives an IT organization a chance to work with a successful, broadly deployed Web service platform, building experience in an arena that increasingly serves as a model for future solution development. By examining eBay's best practices, a savvy business can transfer knowledge into an existing IT organization.

Growth Through Acquisition

Web service–based integration is not limited to organizations outside of eBay. For example, eBay recently acquired www.mobile.de, the largest provider of automotive classified in Germany. The engineer assigned to integrate with eBay's Web services platform was able to build a working prototype of integration over a weekend.

The most dramatic example of this integration is PayPal. Acquired in October, 2002, PayPal can be considered the most significant and successful of all of the eBay Web service integrations. Of eBay's billion-plus Web service requests served monthly, roughly 40% serve PayPal.

PayPal Web Services

PayPal, in turn, has also invested heavily

> "businesses can build integration directly into their existing channel infrastructure"

in a developer framework and Web services. Part of its merchant tools, PayPal Web services offers four informational and transactional APIs: TransactionSearch, GetTransactionDetails, RefundTransaction, and MassPay. These APIs enable developers to create payment applications that integrate with the PayPal platform. One additional interesting component, Instant Payment Notification (IPN), allows real-time integration of the PayPal payment processing system with a third-party application. For example, a developer can use IPN to respond to an incoming payment by e-mailing a registration code to an end-user and sending a fax to a fulfillment partner.

PayPal has grown in a similar fashion to eBay, from simple interfaces based on HTTPS, to the recent addition of SOAP and WSDL-based Web services. PayPal has recently added a test environment called the PayPal Sandbox, which is very similar to the eBay Sandbox. In addition, PayPal Developer Central offers information about how to set up developer certificates, get started with PayPal APIs, and access developer forums for discussion and questions. Given the challenges inherent in building a secure system for working with financial transactions over the Internet, PayPal's advances in the Web services arena represent a significant step forward for e-commerce.

As eBay continues to grow and acquire new entities, the Web services framework serves as an important strategic solution.

For more information see www.paypal.com/pdn.

Conclusion

"A platform is something that other people build on top of. Hundreds of thousands of people have built successful businesses on eBay. Millions have built relationships with others over shared areas of interest. eBay has been a platform long before we had an official Platform or Developer program. You know you're an effective platform when someone builds something you never predicted." – Pierre Omidyar, eBay founder and chairman

eBay's need for Web services grew out of a need to grow the company while retaining the flexibility for others to integrate and innovate. From there, Web services have grown to serve an important strategic role in the growth of the company. By focusing on the pragmatic, critical needs of the company, partners, and users, eBay has become a silent giant of the Web services world.

Through Web services, eBay has been able to transform itself from a popular Web site into a Web platform. Through the use of core technologies such as XML, HTTP, SOAP, and WSDL, eBay has been able to provide highly scalable, well-performing Web services. The business benefits are real today, and are expected to serve as a critical engine for growth into the foreseeable future.

About the Author

Will Iverson previously served as developer relations manager for VisualCafe at Symantec and as the Java Product Manager at Apple Computer. He currently serves as the application development practice manager at SolutionsIQ (www.solutionsiq.com). Will is the author of *Real World Web Services* (O'Reilly), as well as upcoming titles on Hibernate and Jakarta Commons.

wiverson@cascadetg.com

J2EE + .NET

Greater than J2EE versus .NET

A few years ago, a heated debate raged over the benefits of two competing technologies: .NET versus J2EE. Microsoft advocates affirmed that .NET was superior and likewise Java enthusiasts asserted similar observations about J2EE. As it turns out, both camps were at least partially correct.

WRITTEN BY
ED COHEN

The progression has been interesting to watch: both technologies have carved out niches on either side of the corporate firewall. For example, the large enterprise software makers – especially the major ERP vendors such as Oracle, PeopleSoft, and SAP – have developed business software applications almost exclusively based on J2EE. However, a quick glimpse behind the firewall shows that application customers are selecting .NET as the internal development tool of choice.

While each technology has a set of strengths and weaknesses, the advent of Web services has changed the paradigm because it facilitates communication between the two technologies. What's more, from the enterprise perspective J2EE and .NET are beginning to look less and less like competitive technologies and more like complementary languages working hand in hand.

.NET Preferred for Internal Custom Development

Microsoft's decision to support so many different developing environments to ensure that all are capable of producing .NET code was nothing short of brilliant. For many companies and customers of enterprise applications, this event was the single most compelling reason to transition toward a .NET environment for internal development.

Microsoft's support enabled enterprises to leverage the existing expertise of programmers with skill sets in languages such as Visual C++ and Visual Basic because these programmers could immediately begin producing applications in .NET. To add to the benefits, Microsoft provided tools and guidelines for taking the entire legacy client/server and desktop applications and porting them to Web-based environments.

As with many choices in technology, advantages gained generally also come with drawbacks. The adaptation of .NET technology comes with dependencies on platforms, browsers, and licenses from Microsoft. This fact has been the impetus for application developers to use J2EE more often than not.

Flexibility Out-values Productivity

What a developer gains in productivity from .NET is lost in flexibility and the businesses that develop enterprise applications have taken note. Although Java in a J2EE design is a considerably more difficult environment in which to program, once built, the flexibility of the application deployment and the scalability of a J2EE architecture far outweighs the additional development costs.

To put it another way, in order to develop and sell an enterprise application to the widest possible customer base, the application must run on any hardware platform, with any operating systems and without licensing fees. This must include everything from the smallest handheld device to the largest cluster of servers. To this end, there is one choice: J2EE.

Truce Between Giants

So why are these two technologies becoming complementary? The most overlooked component contributing to this trend is the unprecedented willingness of Microsoft and Sun Microsystems to agree to work together on a common standard for Web services. This now allows enterprise application customers – many of whom have already made substantial investments in one technology or another – to easily use both J2EE and .NET. Traditionally, the only choice for enabling various applications, built within these separate environments, to communicate was with extensive – if not expensive – custom development.

While some might argue that Web services is not the best possible solution for every interface, the fact that users now have a convenient method for exchanging data between enterprise applications developed in different technologies cannot be overlooked. For example, with Web services a major ERP application developed in J2EE now has the capacity to communicate with a Web portal developed in .NET.

For those that doubt the significance of the truce between Sun Microsystems and Microsoft imagine this: without the standard the enterprise software industry would be looking at years of competing technologies similar to the browser battles between Netscape and Internet Explorer that reigned in the 1990s.

The Corporate Customer Wins with Standards

Regardless of which technology is superior, the standardization of Web services brought with it the dawn of a new era in software development. End users no longer have to sink all of their hopes into one technology or the other – the freedom to choose from the best applications without concern for the core technology opens the possibility for new advancements in technology and the function of technology. In summary, two formerly competitive technologies have begun to look more and more complementary, and the end result means that the corporate customer wins.

About the Author

Ed Cohen is the chief technology officer for Plateau Systems (www.plateau.com), the leading provider of enterprise software that manages learning and organizational readiness.

ecohen@plateau.com

The Gilbane Conference

ON CONTENT MANAGEMENT TECHNOLOGIES

November 30-December 2, 2004
Boston, Massachusetts
The Westin Copley Place

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.

WHO SHOULD ATTEND:

IT Professionals

- IT Strategists, Managers, Staff
- Content Management Project Managers
- Content Management System Designers
- Intranet, Internet, Extranet, Portal Managers
- Webmasters, Developers, and Administrators
- Enterprise Architects

Business Managers and Strategists

- Product Data Managers
- Information Architects/Knowledge Managers
- Technical Documentation Managers
- Business, Market, and Technology Analysts
- Consultants and Integrators
- Marketing and Product Executives
- Brand Managers

Register NOW!

Register now for the **CONFERENCE PLUS PACKAGE** and receive a **FREE iPOD!**

LIMITED QUANTITY, REGISTER EARLY!

LEARN about successful implementation of content technology through the unique combination of

- **Case Studies**
- **Implementation Techniques**
- **Technology Analysis**

"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

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

Gold Sponsors

EMC² | documentum · GMC · INTERWOVEN · STELLENT™ · VIGNETTE the e:/iciency experts

Media Sponsors

Business Wire · CMS WATCH · EContent · The Gilbane Report · InfoWorld · TRANSFORM · WebServices · XML Journal

Association Sponsors

IDEAlliance · OASIS

Best Practices for Transition Planning

Taking control of SOA

Despite the magnitude of a migration to a service-oriented platform, the continuing uncertainty of critical WS-* standards, and the often thundering impact of large-scale SOA deployments, now is the time to start considering the move. The key to a successful transition is to find a spot of calm amidst the storm of activity surrounding SOA, and develop an intuitive plan that will guide your organization through a path of technical obstacles, organizational resistance, and ever-shifting industry trends.

WRITTEN BY
THOMAS ERL

Count yourself lucky if you've had the foresight to coordinate your migration efforts around this type of plan. Many organizations stumble their way into the world of Web services and what they perceive to be SOA. This "inadvertent migration" is often the result of blindly adopting updates to established development tools and following the vendor's lead with regards to how the next generation of applications should be built this time around. The danger of allowing architecture to evolve in this manner is that you never attain a sufficient level of control over the underlying technology. Little or none of your environments become standardized, and new product platforms further entrench layers of proprietary technology.

One of the core benefits of SOA is its ability to unify previously disparate environments. In planning a migration to a standardized adoption of SOA you therefore have an opportunity to erase some of the neglect from the past. Web services and the vendor-neutral communications framework they establish can bridge disparity and unify interoperability on both data representation and messaging levels. SOA fully leverages this framework by introducing standard service design characteristics that foster interoperability between services, solutions, and organizations.

The potential upside is obviously significant, and well worth the battle that lies ahead in achieving this transition. Your best defense against the potential disruptions, costs, and risks that can follow SOA into an organization is the creation of an SOA Transition Plan. This article provides a set of best practices that will assist you in preparing for and developing such a plan.

Define SOA Within Your Own Organization

Surprisingly, one of the greatest obstacles to adopting SOA is a lack of understanding of what actually constitutes a service-oriented architecture. If you were handed a blank piece of paper and asked to write a generic definition of SOA in the next 10 minutes, what would it be? You (and any other decision makers who are part of a migration project) need to be able to clearly describe concepts such as service autonomy, process encapsulation, and service composition, while distinguishing the differences between SOA and traditional distributed architectures. You'll also need to understand how SOA affects business process modeling and introduces a new integration model.

Without a clear vision of what SOA is in the abstract, you will be in no position to contribute to or even assess the merits of a transition plan. This is because the core of the plan will need to consist of a high-level design in which you define how and to what extent SOA will become part of your organization. This definition will then become the ultimate goal of your migration effort, followed by detailed technical designs, organization processes, and strategic decisions that will be determined in support of this goal.

Invest in an Impact Analysis Before Developing the Transition Plan

In order to assess the feasibility of a transition to SOA, you first need to estimate the real-world impact such a migration will have. Therefore, you should consider holding off any sort of planning until an initial impact analysis is complete.

SOA's reach is broad. Hence the scope of this report may exceed what you are accustomed to associating with a traditional impact analysis. Specifically, this type of research effort tends to include in-depth assessments of current and upcoming development tools, infrastructure requirements, skill-set and training requirements, proposed new middleware, changes to organizational processes, changes to security models and policies, and a list of recommendations associated with architecture, custom standards, and project management approaches.

Keep in mind that if you decide to postpone your transition, much of this analysis work may lose its value. While SOA design principles will remain relatively static, a great deal of the underlying technology will continue to evolve and change. Even a delay of one year will require a significant re-analysis. On the other hand, if you do decide to proceed with a transition now, this upfront analysis effort will have already laid the groundwork for your project. (However, if you are planning a long-term migration, re-analysis may still be required, and should be a task scheduled as part of the plan.)

Set the Scope of the Transition

Using the impact analysis results as your primary guide, and factoring in budget constraints, related project requirements, and other external drivers (such as strategic business goals), you should be able to determine the scope of your planned transition. It is not uncommon for an SOA transition plan to apply only to a subset of an organization's technical environment. For instance, there may be several legacy areas of an enterprise that do not warrant any intrusion by service encapsulation. Perhaps

your goal is to build a dedicated hosting environment intended only to support new service-oriented applications. More often than not, however, integration requirements drive SOA transitions, in which case the scope of your project could easily see the introduction of SOA affect a majority of your IT enterprise.

Setting the scope of your migration pretty much determines the structure and size of your plan. The scope also dictates a great many parts of your plan's deliverables, including the amount of new technology introduced, the extent to which existing organizational processes are modified, the number of new or revised standards required, and the migration model that is chosen. The use of a migration model, such as the Layered Scope Model, is highly advisable, as it provides you with a customizable template for building project phases around specific migration milestones.

> "one of the greatest obstacles to adopting SOA is a lack of understanding of what actually constitutes a service-oriented architecture"

Change the Project Team's Mindset

Service-oriented principles themselves are not complex, however, the application of these principles can result in relatively complex automation solutions. This is especially the case when services from different solutions are shared and composed to support new or modified business processes. If you're going to live in a service-oriented world, your project team will need to change the way it thinks about fundamental aspects of common architecture, such as componentization, integration, and process automation.

One of the more significant revelations team members will be faced with is how much SOA differs from an architecture that uses Web services only peripherally. Even a large-scale deployment of Web services does not necessarily constitute an SOA. This new design paradigm brings with it a set of distinct principles that need to be applied in order for an architecture to be considered truly service-oriented.

There also needs to be a common acceptance of the fact that service-oriented theory is not solely related to the eventual implementation of technology. Concepts, such as process encapsulation and service composition, extend beyond technical environments, and can change the manner in which business processes are structured and modeled. This has significant implications, as the introduction of SOAs can force change beyond the realm of technology, affecting business analysts and existing modeling standards.

Without a sound understanding of these and other changes to how distributed solutions are traditionally designed, you will introduce risks to your project. Foremost of these is the danger of laying a weak foundation built around the use of Web services, but omitting the unique characteristics that define SOA.

Expect Evolution to be Part of the Migration

It may read like a tired line, but investing in SOAs is "building for the future." Whether you need it or not, you will be creating an environment in which interapplication (and even intra-application) communication becomes nonproprietary, and therefore provides the constant potential of open interoperability. You will also be designing solutions consisting of autonomous units of application logic (services) that can be efficiently composed and recomposed to accommodate changes in the business processes they automate.

These, and other SOA characteristics, can lead to significant benefits and an overall superior enterprise. Achieving all of this, however, comes at the price of serious commitment and investment. It also requires that you stay open to the sometimes unpredictable evolutionary nature of SOA. In larger scale transition projects, it is not uncommon for the environment you end up migrating toward to be very different from what you originally defined in your transition plan. Further, it will likely continue to change soon after the migration project finishes. This is the nature of SOA.

And, while WS-* standards continue on their volatile path, products that implement these standards will undergo continuous refinements. This can turn a physical target architecture into a moving target. It is easier to simply accept this ahead of time than to get frustrated at having to continually adapt. The bottom line is that when you document the environment you want to establish upon completion of your migration, realize that what actually gets implemented will likely differ. Then, compensate for this by incorporating a means of facilitating regular change (through iterative design cycles, for example, into your plan.

Use Speculative Analysis to Build Toward a Future State

In order to best leverage the potential of Web services within SOAs, it is often advisable to build an environment capable of providing more than what your immediate requirements ask for. A speculative analysis entails a realistic prediction of how your technical environment will change over the immediate future following the completion of your transition. This analysis is based on current corporate goals and anticipated changes to an organization's business areas. Typically, you won't want to venture out more than 6–12 months with these predictions. What often results is a set of new requirements (commonly integration-related) that can be greatly facilitated by introducing design changes ahead of time.

With this approach you obviously run the danger of wasting time and money on features that end up not getting used. Of course, anything can change and no speculation can be guaranteed. However, if a little more up-front design and development effort can spare the significantly larger amount of effort required to redesign and redevelop solutions (which incurs associated testing and deployment overhead), then it's worth the gamble.

For example, imagine that the results of your analysis indicate a strong likelihood of

two systems having to communicate with each other within a year after you introduce SOA to your organization. In this case you are better off taking that potential requirement into account ahead of time. It may not even necessarily require that you build new functionality into your current system; you can often accommodate future state architectures simply by designing increasingly autonomous services with highly generic interfaces, and by positioning these services as future application endpoints.

Prepare for Post-Migration Growth

A common mistake is to limit the scope of a transition plan to the cycle of the migration process. Even if the ultimate goal of your plan is to establish Web services based on service-oriented principles as the primary computing platform within your organization, you still need to take the post-migration growth of this new environment into account ahead of time.

Many organizations focus almost exclusively on the introductory phase of the migration, ignoring what it will take to maintain and expand these environments. This oversight can be costly. For example, IT managers are often shocked when they realize how little of their existing infrastructure can be used to manage large-scale Web services deployments (the second shock generally occurs when the cost of upgrading the administration infrastructure is presented).

The best way to avoid this situation is to speculate on the growth rate of Web services and SOAs within your organization, and to then dedicate part of the up-front analysis to an evaluation of current administration tools. Match up licenses with expected administration requirements, add a contingency for changes in the marketplace and growth variance, and put this figure in your initial budget. Finally, even if the anticipated usage of SOAs is minimal upon completion of the transition, ensure that an extensible support infrastructure is delivered as part of the migration project.

Plan Transition Phases Around the Introduction of Custom Standards

Nothing is more important to realizing the benefits of SOA than creating and enforcing custom standards. SOA promises to establish an environment that is open, adaptive, and fosters interoperability through a common, globally accepted communications platform. A primary benefit of this architecture is its ability to bridge platform and data disparity. Without standardizing this effort, however, the benefits actually realized will be superficial. Nonstandardized SOAs will, in fact, end up contributing to existing disparity.

"Phasing in SOA means phasing in standards." Base your approach on a motto similar to this, and you should see the delivery of standards documents take a prominent place in your migration project. Planning the delivery of standards, however, is only the first step. The enforcement of a standardized SOA will not only introduce changes to technology and the organization as a whole, it will also impose new constraints and limitations. It should therefore be recognized that phasing in SOA and associated custom standards in a controlled manner requires the support and cooperation of the entire IT department.

Though an entirely separate topic, also worth noting is the creation of standards for distributed applications that will not be immediately migrated to SOA. There are a number of steps you can take to build service-oriented characteristics into a solution in preparation for a future migration. Not only will this ease an eventual transition to the Web services platform, it will greatly facilitate any integration with service-oriented applications in the interim.

Other Best Practices

These best practices have focused almost exclusively on the groundwork of your transition plan. Unfortunately that is all I've been able to provide here. Once you actually start authoring the details of your plan, more best practices come into play, such as:

- Assigning new roles and responsibilities
- Creating subprojects based on project phases
- Defining transition architectures
- Incorporating a service-oriented security model
- Periodically revisiting original impact analysis results
- Planning around application endpoints
- Structuring plans in support of ROIs

Summary

A good transition plan must coordinate the strategic positioning of new concepts and technologies within an organization. With SOA, the execution of the plan achieves an alignment of proprietary technology and open industry standards with an organization's business goals. In other words, you need to seize ownership over how SOA is phased in on corporate, organizational, and technical levels. Approaching migration in this manner minimizes dependencies on external drivers, and places the evolution of this architectural platform in your hands.

About the Author

Thomas Erl is the president and chief architect of XMLTC Consulting Inc. (www.xmltc.com) in Vancouver, Canada, specializing in the delivery of service-oriented and XML-centric solutions. He is the author of two books for Prentice Hall/PTR: *Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services* and the upcoming *Service-Oriented Architecture: Concepts and Technology* (see www.serviceoriented.ws). Thomas has published over 30 papers, and established an integration framework for XML and Web services (see www.thomaserl.com/technology).

terl@xmltc.com

Continued from page 8

forward. Yet as powerful as these new platforms promise to be, this power will still come at a cost. It's the cost of complexity – today's systems are very robust but very complex.

The level of complexity involved in SOAs today requires enterprises to look beyond just platform advancements and more into the way they manage their software and IT systems. Businesses need to demand the same level of rigor and predictability in how they develop and deliver software as they do in other critical aspects of their business. Today it's too chaotic. There must be a higher level of predictability in building SOAs, more visibility into the risks and costs involved, the impact on business and development teams, and a better understanding of the benefits and return on investment a business hopes to acheieve with these plans.

This is an issue no matter what platform an enterprise relies upon for Web services.

About the Author

Dale Fuller is president and chief executive officer of Borland Software Corporation, with complete management responsibility for the company. He joined Borland in April 1999 with more than 20 years of experience in general management, marketing, and business development in the technology industry. Presently, Fuller focuses on extending the company's leadership in both the development marketplace and enterprise deployment.

A LIMITED TIME SAVINGS OFFER FROM *SYS-CON MEDIA*

SUBSCRIBE TODAY TO MULTIPLE MAGAZINES

AND SAVE UP TO $340 AND RECEIVE UP TO 3 FREE CDs!*

RECEIVE YOUR DIGITAL EDITION ACCESS CODE INSTANTLY WITH YOUR PAID SUBSCRIPTIONS

3-Pack

Pick any 3 of our magazines and save up to $210.00

Pay only $99 for a 1 year subscription plus a FREE CD

- 2 Year – $179.00
- Canada/Mexico – $189.00
- International – $199.00

6-Pack

Pick any 6 of our magazines and save up to $340.00

Pay only $199 for a 1 year subscription plus 2 FREE CDs

- 2 Year – $379.00
- Canada/Mexico – $399.00
- International – $449.00

9-Pack

Pick 9 of our magazines and save up to $270.00

Pay only $399 for a 1 year subscription plus 3 FREE CDs

- 2 Year – $699.00
- Canada/Mexico – $749.00
- International – $849.00

CALL TODAY! 888-303-5282

Pick a 3-Pack, a 6-Pack or a 9-Pack			
☐ 3-Pack	☐ 1YR ☐ 2YR	☐ U.S. ☐ Can/Mex	☐ Intl.
☐ 6-Pack	☐ 1YR ☐ 2YR	☐ U.S. ☐ Can/Mex	☐ Intl.
☐ 9-Pack	☐ 1YR ☐ 2YR	☐ U.S. ☐ Can/Mex	☐ Intl.

TO ORDER •Choose the Multi-Pack you want to order by checking next to it below. •Check the number of years you want to order. •Indicate your location by checking either U.S., Canada/Mexico or International. •Then choose which magazines you want to include with your Multi-Pack order.

☐ **LinuxWorld Magazine**

U.S. - Two Years (24) Cover: $143	You Pay: $79.99 /	Save: $63 + FREE $198 CD
U.S. - One Year (12) Cover: $72	You Pay: $39.99 /	Save: $32
Can/Mex - Two Years (24) $168	You Pay: $119.99 /	Save: $48 + FREE $198 CD
Can/Mex - One Year (12) $84	You Pay: $79.99 /	Save: $4
Int'l - Two Years (24) $216	You Pay: $176 /	Save: $40 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

☐ **JDJ**

U.S. - Two Years (24) Cover: $144	You Pay: $99.99 /	Save: $45 + FREE $198 CD
U.S. - One Year (12) Cover: $72	You Pay: $69.99 /	Save: $12
Can/Mex - Two Years (24) $168	You Pay: $119.99 /	Save: $48 + FREE $198 CD
Can/Mex - One Year (12) $120	You Pay: $89.99 /	Save: $40
Int'l - Two Years (24) $216	You Pay: $176 /	Save: $40 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

☐ **Web Services Journal**

U.S.- Two Years (24) Cover: $168	You Pay: $99.99 /	Save: $68 + FREE $198 CD
U.S. - One Year (12) Cover: $84	You Pay: $69.99 /	Save: $14
Can/Mex - Two Years (24) $192	You Pay: $129 /	Save: $63 + FREE $198 CD
Can/Mex - One Year (12) $96	You Pay: $89.99 /	Save: $6
Int'l - Two Years (24) $216	You Pay: $170 /	Save: $46 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

☐ **.NET Developer's Journal**

U.S. - Two Years (24) Cover: $168	You Pay: $99.99 /	Save: $68 + FREE $198 CD
U.S. - One Year (12) Cover: $84	You Pay: $69.99 /	Save: $14
Can/Mex - Two Years (24) $192	You Pay: $129 /	Save: $63 + FREE $198 CD
Can/Mex - One Year (12) $96	You Pay: $89.99 /	Save: $6
Int'l - Two Years (24) $216	You Pay: $170 /	Save: $46 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

☐ **Information Storage + Security Journal**

U.S. - Two Years (24) Cover: $143	You Pay: $49.99 /	Save: $93 + FREE $198 CD
U.S. - One Year (12) Cover: $72	You Pay: $39.99 /	Save: $39
Can/Mex - Two Years (24) $168	You Pay: $79.99 /	Save: $88 + FREE $198 CD
Can/Mex - One Year (12) $84	You Pay: $49.99 /	Save: $34
Int'l - Two Years (24) $216	You Pay: $89.99 /	Save: $126 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $59.99 /	Save: $48

☐ **Wireless Business & Technology**

U.S. - Two Years (12) Cover: $120	You Pay: $49.00 /	Save: $71 + FREE $198 CD
U.S. - One Year (6) Cover: $60	You Pay: $29.99 /	Save: $30
Can/Mex - Two Years (12) $120	You Pay: $69.99 /	Save: $51 + FREE $198 CD
Can/Mex - One Year (6) $60	You Pay: $49.99 /	Save: $10
Int'l - Two Years (12) $120	You Pay: $99.99 /	Save: $20 + FREE $198 CD
Int'l - One Year (6) $72	You Pay: $69.99 /	Save: $2

☐ **MX Developer's Journal**

U.S. - Two Years (24) Cover: $143	You Pay: $49.99 /	Save: $93 + FREE $198 CD
U.S. - One Year (12) Cover: $72	You Pay: $39.99 /	Save: $32
Can/Mex - Two Years (24) $168	You Pay: $79.99 /	Save: $88 + FREE $198 CD
Can/Mex - One Year (12) $84	You Pay: $49.99 /	Save: $34
Int'l - Two Years (24) $216	You Pay: $89.99 /	Save: $126 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $59.99 /	Save: $48

☐ **ColdFusion Developer's Journal**

U.S. - Two Years (24) Cover: $216	You Pay: $129 /	Save: $87 + FREE $198 CD
U.S. - One Year (12) Cover: $108	You Pay: $89.99 /	Save: $18
Can/Mex - Two Years (24) $240	You Pay: $159.99 /	Save: $80 + FREE $198 CD
Can/Mex - One Year (12) $120	You Pay: $99.99 /	Save: $20
Int'l - Two Years (24) $264	You Pay: $189 /	Save: $75 + FREE $198 CD
Int'l - One Year (12) $132	You Pay: $129.99 /	Save: $2

☐ **WebSphere Journal**

U.S. - Two Years (24) Cover: $216	You Pay: $129.00 /	Save: $87 + FREE $198 CD
U.S. - One Year (12) Cover: $108	You Pay: $89.99 /	Save: $18
Can/Mex - Two Years (24) $240	You Pay: $159.99 /	Save: $80 + FREE $198 CD
Can/Mex - One Year (12) $120	You Pay: $99.99 /	Save: $20
Int'l - Two Years (24) $264	You Pay: $189.00 /	Save: $75
Int'l - One Year (12) $132	You Pay: $129.99 /	Save: $2

☐ **PowerBuilder Developer's Journal**

U.S. - Two Years (24) Cover: $360	You Pay: $169.99 /	Save: $190 + FREE $198 CD
U.S. - One Year (12) Cover: $180	You Pay: $149 /	Save: $31
Can/Mex - Two Years (24) $360	You Pay: $179.99 /	Save: $180 + FREE $198 CD
Can/Mex - One Year (12) $180	You Pay: $169 /	Save: $11
Int'l - Two Years (24) $360	You Pay: $189.99 /	Save: $170 + FREE $198 CD
Int'l - One Year (12) $180	You Pay: $179 /	Save: $1

☐ **WLDJ**

U.S. - Four Years (24) Cover: $240	You Pay: $99.99 /	Save: $140 + FREE $198 CD
U.S. - Two Year (12) Cover: $120	You Pay: $49.99 /	Save: $70
Can/Mex - Four Years (24) $240	You Pay: $99.99 /	Save: $140 + FREE $198 CD
Can/Mex - Two Year (12) $120	You Pay: $69.99 /	Save: $50
Int'l - Four Years (24) $240	You Pay: $120 /	Save: $120 + FREE $198 CD
Int'l - Two Year (12) $120	You Pay: $79.99 /	Save: $40

*WHILE SUPPILES LAST. OFFER SUBJECT TO CHANGE WITHOUT NOTICE

Subscribe Online Today www.sys-con.com/2001/sub.cfm

SYS-CON MEDIA

.NET: Platform of Choice

Prepared for tomorrow's service-oriented world

I recently wrote an editorial as editor-in-chief of *.NET Developer's Journal* in which I openly questioned the value of re-architecting existing systems to use the latest and greatest technologies. Specifically, I illustrated my argument with the case of a local ISV (independent software vendor) I know that spent several months re-architecting a successful COM-based application to use .NET while its competitor continued to add features to its existing system. The end result was that after a few months of development by both organizations, the company that had stayed on its legacy platform and invested in features had a much richer product than the folks who had spent their resources porting to .NET.

WRITTEN BY
DEREK FERGUSON

Unfortunately, this editorial has been misunderstood in some quarters as an attack on the virtues of upfront design versus most organizations' "just start coding" approach. Others among my critics have suggested that I bring the value of the .NET platform into question whenever I suggest that not every legacy system out there is an equally good candidate for migration to my favorite platform. To both of these groups, I have just one thing to say: get real!

As a Microsoft Most Valuable Professional for the .NET Compact Framework and principal consultant for Magenic Technologies, Inc. (http://www.magenic.com), a premier software development and consulting company focused on Microsoft technology, I am well acquainted with the value of up front design and education. Organizations that refuse to do proper planning and design before embarking on the construction of any modern, large-scale system will soon find themselves bleeding money like stuck pigs once they get midway in their development efforts.

This truth will become increasingly evident as the move towards interconnected, service-oriented architectures (SOA) continues to accelerate over the remainder of this decade. SOA has rendered forever obsolete the idea of the lone programmer acting in complete isolation as analyst, architect, and coder for any serious software project.

> "SOA has rendered forever obsolete the idea of the lone programmer acting in complete isolation"

We are fortunate as software engineers to have a platform as robust and extensible as .NET upon which to build the large-scale SOA systems of the near future. Virtualized execution on the CLR (Common Language Runtime) represents the same quantum leap forward in reliability and security as the move from DOS to Win32 before it. The ability to transparently access, store, and expose object-based functionality via industry-standard XML is – although not strictly required by SOA – so fundamental to 21st-century interoperability requirements as to make any attempt at modern software development in its absence nearly impossible!

The devil in all of these details, however, is the phenomenal difficulty that can arise in implementing any of these technologies without sufficient expertise. .NET reduces the knowledge base a developer needs in order to be successful with SOA to the absolute barest minimum. It is for this single reason that I expect .NET to become an increasing dominant platform as the role of SOA within organizations continues to expand in 2005 and beyond.

In preparing this editorial for publication in *WSJ*, I consulted a number of software engineering experts here in Illinois, where I live. Arguably the most famous person to respond was Ralph Johnson – one of the coauthors of Addison-Wesley's famous *Design Patterns* book and currently on the faculty at the University of Illinois at Champaign-Urbana. To read about what he had to say, as well as my response, visit my blog at http://derek.blog-city. com/.

About the Author

Derek Ferguson is a principal consultant for Magenic Technologies, Inc. (http://www.magenic.com), a premier software development and consulting company focused on Microsoft technology. He is also editor-in-chief of *.NET Developer's Journal* and a Microsoft Most Valuable Professional for the .NET Compact Framework.

derek@sys-con.com

XML JOURNAL SUPPLEMENT

This Month

XQuery: Reporting XML Data

BY SESHU KUMAR ADIRAJU & SRINIVAS THONSE

Reporting is an important functionality in software business applications and is now increasingly required for XML data. Applications typically generate reports by extracting relevant information from a database. Applications that don't adopt XSL/FO and proprietary techniques.

Application Integration Using XML: An Open-Architecture Approach Provides Unforeseen Benefits

BY MIKE MORGAN

Application integration comes in many forms, whether it be integration between components of a single software system or integration between systems. Making use of a common communications protocol between disparate systems allows for the creation of a system in which the benefits of the combined system exceed the benefits of each of the standalone systems.

Citrix Systems – A Global Company with a Mission

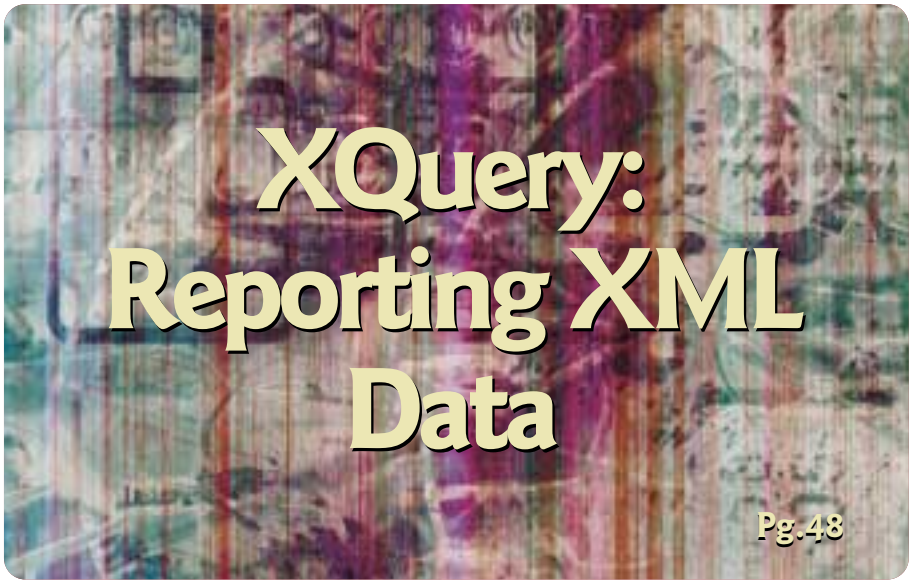
BY GIJSBERT IN 'T VELD

Citrix, a leader in access infrastructure solutions, launched a global systems integration project and turned to Microsoft and Covast for help. They chose the Covast EDI Accelerator to provide connectivity to their 120 distributors worldwide and to enable the exchange of EDI documents.

Managing Engineering Information

BY CHRIS RANDLES

It's virtually axiomatic: technology innovations first boost personal productivity then group productivity. Today's goal is to harness other personal productivity gains that technology has delivered and transfer them to entire organizations.



XML-Based Interop, Close Up

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

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

Content Management:
Organization, dissemination, and presentation of information

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

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

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

XQuery: Reporting XML Data

Using XQuery and templates

WRITTEN BY
SESHU KUMAR ADIRAJU & SRINIVAS THONSE

Reporting is an important functionality in software business applications and is now increasingly required for XML data. Applications typically generate reports by extracting relevant information from a database. Applications that don't use the underlying database for information storage adopt XSL/FO and proprietary techniques. Other solutions rely on migrating data to a database to enable reporting. These techniques are inflexible and expensive for reporting XML data.

A cost-effective and flexible reporting framework allows quick report creation and keeps the reporting independent of the existing code base and the XML data source. This article examines a reporting framework that uses Report Templates and XQuery to fit the bill. The report templates contain static content and formatting information for the report (in HTML) and embedded XQuery constructs to generate dynamic content.

The reporting framework provides an execution context and also some value-added services like converting the XML report to desired formats such as PDF, RTF, and HTML. XQuery is based on XML with powerful but easy to use syntax like SQL. It is system-independent and applicable across all sorts of XML data from files to databases. The use of Report templates can result in faster development cycles, improving turnaround time for report creation and offering greater flexibility.

Context

As an increasing amount of information is stored, exchanged, and presented using XML, the ability to intelligently query XML data sources and report this data becomes more important.

While using XML for data representation brings a lot of flexibility, extensibility, and openness, the lack of a high-level query language until now meant that reporting XML data was complex. The XQuery specification provides a powerful and convenient language designed for processing of XML data.

In this article, we propose an XML reporting framework using XQuery. The framework allows creation of reports that are not merely a transformation of XML data but involve querying, extracting and analysis as well. XQuery aims to provide a standard querying mechanism for all kinds of XML data sources. Therefore, a reporting framework using XQuery can be used in a variety of applications using XML data.

The reporting framework was developed for and used successfully in a UML-based modeling tool developed at Infosys.

Current Approaches

Most existing solutions use XSL files for reporting, but there are many disadvantages in this approach. For one, XSL is mainly a transformation, not a query, language. The syntax is difficult for nonprogrammers. And, this technique is file based and cannot work with other XML data sources. Other approaches are proprietary and come with an additional cost. Some solutions use a template-based approach for reporting, but use proprietary tags in the template.

This requires a standards-based solution that will allow querying from disparate data sources. We require a reporting/analysis framework that:

- Uses a query language that is easy to use (by domain experts)
- Uses a query language that supports arithmetic and Boolean operations, conditional logic, recursion, aggregation, sorting, filtering and access to environment information
- Supports disparate XML data sources including streamed XML data
- Allows easy formatting of the report, preferably through templates
- Facilitates easy customization of reports

Introduction to XQuery

Let's take a high-level look at XQuery and cover some aspects relevant for this article. (For more details on XQuery see References.)

XQuery introduces an easy-to-use, high-level query language syntax for XML. XQuery is to XML what SQL is to databases. It is designed to be a language in which queries are concise, easily understood, and human readable. It is also flexible enough to query a broad spectrum of XML information sources, including both databases and documents. Like SQL for relational databases, XQuery will become a system-independent standard, applicable across all sorts of XML data.

XQuery offers different types of expressions – called FLWR (pronounced "flower") expressions. FLWR is an acronym for the for-let-where-return XQuery keywords. Also, it supports path-

expressions and element/attribute constructers. Using these features we can use XQuery to Query, Transform or even construct XML documents.

XQuery can be used in different usage scenarios, such as,

- ***XML Transformation:*** XQuery provides a powerful (extensible) mechanism for nontrivial transformations of XML documents from one format into another .
- ***XML Data Extraction:*** XQuery can be used to extract XML data from different sources like XML documents, relational databases or Native XML Databases.
- ***HTML/XHTML Generation:*** HTML/XHTML pages can be generated dynamically using embedded XQuery, similar to other techniques like JSP, ASP, and PHP, etc.

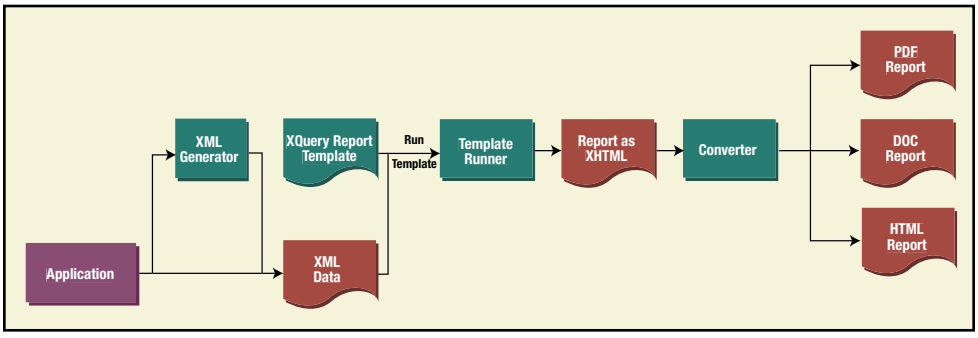


Figure 1 • XQuery reporting framework

Proposed Framework: Report Templates Using XQuery

In this section we present a new Reporting framework using Report templates and XQuery. This reporting framework can be used by any application that stores application data in XML.

Reporting Framework

The Application "XML Data" can be available in different XML formats. The report content and structure are defined in an "XQuery Report Template". The "Template Runner"' runs the Report Template against XML Data to generate a report in XHTML format.

Optionally this report can be converted into other formats like PDF or RTF. Also, for applications storing data in non XML formats, an additional component "XML Generator" can create XML representation of its data for reporting purposes (see Figure 1).

The Reporting Framework contains two main components, "XQuery Report Template" and "Template Runner," described below.

XQUERY REPORT TEMPLATE

The Report Templates are essentially HTML documents with embedded XQuery. This approach is very similar to dynamic scripting languages like JSP or ASP.

The HTML part of the template defines the overall report layout and formatting information. The embedded XQuery expressions define the dynamic content that should be queried from the project XML. In fact, HTML and XQuery can be combined in two ways. One, we can embed XQuery expressions in HTML tags. Also, we can embed HTML within XQuery. This provides a lot of flexibility in structuring the report template.

TEMPLATE RUNNER

The template runner provides the execution context for the XQuery execution. It binds the XML Data and the user input required for Report generation to the XQuery processor. It also handles the output from XQuery processor and optionally converts the report into formats like PDF, RTF, and HTML (see Figure 2).

We used the XQuery library Qexo to process the XQuery report.

Using a similar approach, we have also published the model data as a browse-able Web site. This has been found to be an effective way of disseminating model data among the project team.

This approach provides an efficient mechanism to query model data from XML and then use model data for analysis and reporting. The key benefits of this approach are:

- Easy-to-write XQuery language that is human readable and has high-level constructs similar to SQL.
- Easy-to-extend process as new models are introduced.
- Templates can be created independent of the tool without change in tool code. Hence, there are fewer maintenance problems and greater productivity.

An Example

Let's illustrate the XQuery template concept with the simple example of a book catalogue.

Step 1: XML Data

Consider an XML representation of a book catalogue with details of the name of the book and the author. Listing 1 shows an XML for the book catalogue.

The XML has one node for each book and subnodes containing the details like title, author, and price information for each book.

Step 2: Report Template

Suppose that we want to create a simple report showing the list of books. We need to create a Report template that queries the XML file for that data and present the data in a tabular column. Listing 2 shows a report template that does this.

This template is a simple HTML document that mainly contains a <table> tag with a header row and also an embedded XQuery segment. The XQuery segment (enclosed in flower braces '{'and '}') is:

- ***let clause:*** The document command used to load data from an XML document

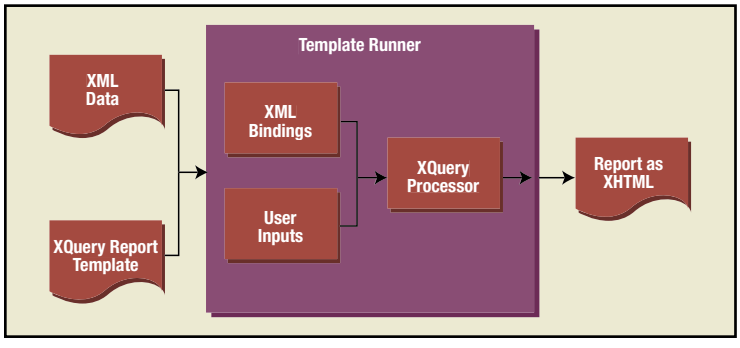


Figure 2 • Template runner

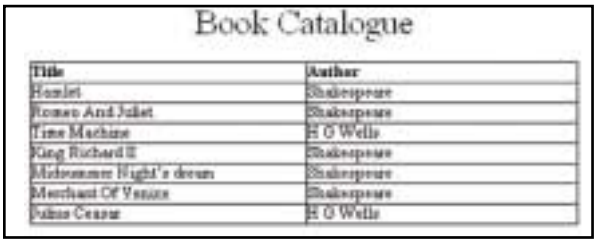


Figure 3 • Sample report

- ***for clause:*** For every "book" in the "catalogue"
- ***return clause:*** Book title and author

The enclosing HTML provides the required formatting.

Step 3: Template Execution

Next, the Template runner executes the Report template (Listing 2) against the XML data (Listing 1).

Template runner creates the execution context for the XQuery processor. The execution context provides access to the XML data and also executes the xquery against it. The context can also be used to bind any other variables that might be provided during execution of report. For example, the report might require user inputs like author name as a filtering criterion. Template runner performs this binding during the report execution.

The output from Template Runner is a XHTML document which can be converted to PDF/RTF/HTML formats. Figure 3 shows an RTF report generated from the Report template shown in Listing 2.

Step 4: Template Modification

One of the main advantages of the template-based approach is that the report templates can be modified easily, without effecting the rest of the application. For example, assume that we need to modify the report created above to show the book price in addition to the title and author. Also, let's say that we require this information only for the books available in paperback format. The modified template would be as shown in Listing 3.

The template introduced an extra column in the HTML table tag for displaying the price information. A "where clause" is introduced to query for only those books available in "paperback" format. The third change is the expression to query and display the price. The generated report is shown in Figure 4.

This simple example shows how easy it is to change both the content and the formatting of the report with the XQuery template approach.

LISTING 1

```
<?xml version="1.0"?>
<catalogue>
   <book>
   <title>Hamlet</title>
      <author>Shakespeare</author>
   <price format="Paperback">4.99</price>
   </book>
   <book>
   <title>Romeo And Juliet</title>
      <author>Shakespeare</author>
   <price format="Hardcover">11.89</price>
   </book>
   <book>
   <title>Time Machine</title>
      <author>H G Wells</author>
   <price format="Paperback">3.99</price>
   </book>
   <book>
   <title>King Richard II</title>
      <author>Shakespeare</author>
   <price format="Paperback">13.99</price>
   </book>
   <book>
   <title>Midsummer Night's dream</title>
      <author>Shakespeare</author>
   <price format="Hardcover">12.59</price>
   </book>
   <book>
   <title>Merchant Of Venice</title>
      <author>Shakespeare</author>
   <price format="Hardcover">12.60</price>
   </book>
   <book>
   <title>Julius Ceaser</title>
      <author>Shakespeare</author>
   <price format="Paperback">4.99</price>
   </book>
</catalogue>
```

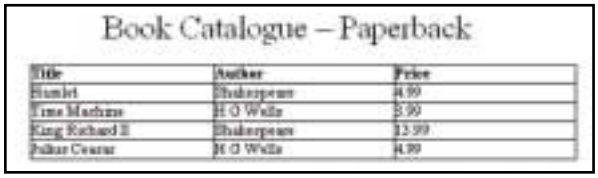
LISTING 2

```
<html>
<body>
<h1 align="center">Book Catalogue</h1>
<table   width="80%"  border="2" columns="2">
 <tr>
  <th> <b>Title</b> </th>
  <th> <b>Author</b> </th>
 </tr>
 {
  let $catalogue := document("catalogue.xml")
  for $book in $catalogue/book
  return
  <tr>
   <td> {string($book/@title) } </td>
   <td> { string($book/@author) } </td>
  </tr>
 }
</table>
</body>
</html>
```

LISTING 3

```
<html>
<body>
<h1 align="center">Book Catalogue - Paperback</h1>
<table   width="80%"  border="2" columns="3">
 <tr>
  <th> <b>Title</b> </th>
  <th> <b>Author</b> </th>
  <th> <b>Price</b> </th>
 </tr>
 {
  let $catalogue := document("catalogue.xml")
  for $book in $catalogue/book
  where $book/price/@format = "Paperback"
  return
  <tr>
   <td> { string($book/@title)  } </td>
   <td> { string($book/@author) } </td>
   <td> { string($book/price)   } </td>
  </tr>
 }
</table>
</body>
</html>
```

Download the Code
www.sys-con.com/xml



Book Catalogue – Paperback

Title	Author	Price
Hamlet	Shakespeare	8.99
Time Machine	H G Wells	5.99
King Richard II	Shakespeare	13.99
Julius Ceasar	H G Wells	8.99

Figure 4 • Report for modified template

Conclusion

XQuery is based on XML and has powerful but easy-to-use syntax like SQL. This makes it easy for authoring templates by domain experts. Using Report templates can result in faster development cycles and keep application and reports independent of each other. This improves the overall turn around time for report creation and also offers greater flexibility.

XQuery is a system-independent standard, applicable across all sorts of XML data. The majority of XML database vendors, tool vendors and even some relational database vendors, have developed or announced plans for XQuery support. Therefore, XQuery Template based reporting is a promising approach.

References

- *XQuery 1.0 W3C Working Draft, Nov. 2002*: www.w3.org/TR/xquery
- *XQuery Requirements, May 2003:* www.w3.org/TR/ xquery-requirements/
- *XQuery-Intro, Per Bothner, 2002:* www.gnu.org/software/qexo/XQuery-Intro.html
- *InFlux Methodology,Infosys Technologies Limited:* www.infosys.com/ technology/influx.asp
- Cheng, Alex. "XQuery: A Flexible Query Language for XML": www.ddj.com/documents/s=7067/ddj0250a/cheng.htm.
- Bothner, Per. (Dec 2003) "Using XQuery to generate HTML." www.xml.com/pub/a/2002/12/23/xquery.html.
- *Qexo: The GNU Kawa XQuery implementation:* www.gnu.org/software/qexo.

Author Bio

Seshu Kumar Adiraju is a technical architect with SETLabs, an applied research unit of Infosys Technologies Limited (www.infosys.com). His areas of interest include J2EE, EAI, and methodologies. He is currently part of the development team of InFlux Workbench, a business process modeling application developed by Infosys. Seshu Kumar has worked on various J2EE software development and Enterprise Application Integration projects. He holds a master of technology degree in chemical engineering from the Indian Institute of Technology, Bombay.

Srinivas Thonse: is a principal architect with SETLabs, the applied research unit of Infosys Technologies Limited (www.infosys.com). His areas of interest include object technology, methodologies, business process management, and software architecture. Srinivas has directed the architecture and design of many enterprise projects. He regularly mentors and trains architects and designers in software design and object methodologies. Srinivas holds a bachelor of engineering degree in computer science from the Bangalore University.

SESHUKUMAR_ADIRAJU@INFOSYS.COM

SRINIT@INFOSYS.COM

WRITTEN BY **MIKE MORGAN**

Application Integration Using XML: An Open-Architecture Approach Provides Unforeseen Benefits

The need to integrate more systems is crucial

Application integration comes in many forms, whether it be integration between components of a single software system or integration between systems. Making use of a common communications protocol between disparate systems allows for the creation of an integrated suite of applications whereby the combined benefits exceed those of each of the stand-alone applications. Using a common, standards-based protocol such as XML as a means of communication between the components in an application provides for integration opportunities and expansion that can extend the lifespan and benefits provided by a system far beyond its original design. This, in turn, can reduce the cost of creating new systems as existing components have already been written and tested. Also, the abilities of other external applications can be integrated into an existing application to enhance the functionality with minimal additional cost. Finally, because XML is a platform-independent protocol, the integration of external applications can be done without regard for the operating system in which they currently execute; hence minimizing the dependence on any one software platform.

This article focuses on:

- In-system communication benefits
- Integration with external systems
- Cost benefits and component reusability

AUTHOR BIO

Mike Morgan is senior software consultant and architect for Edgewater Technology, a consulting and systems integration firm based in Wakefield, MA. Mike specializes in the architecture and development of client/server–based applications on Windows and Internet-based platforms utilizing C/C++/C#, VB, HTML, ASP languages on Windows 2000/NT/XP and UNIX operating systems. His projects include a call center software design study; call center development platform; transportation operations management system; loan processing systems; and the development of a manufacturing application.

In-System Communication Benefits

A large software system is often composed of various specialized pieces of functionality for processing one or more items of data. For example, a manufacturing organization may have an order entry system that has multiple components to perform specific tasks. One component might handle the order fulfillment, another might take care of the shipping and delivery, and a third might handle calculating and generating the billing. By utilizing a simple communications protocol, such as XML, interconnection between these different components can be achieved while still allowing for integration with other off-the-shelf components to increase the overall functionality of the system.

In contrast, making the choice to use a proprietary protocol results in extra development time writing and debugging that protocol, potential firewall-related issues if communication needs to exist across different physical locations (as is almost guaranteed in client/server environments), and difficulties that will be encountered when the users want added functionality and integration with other systems.

Most database vendors provide an interface to their engines utilizing XML, which allows for minimal data manipulation and faster delivery time. Attaching a Web interface to various components of the application can be simplified based on the number of available XML integration packages that are available (e.g., Castor or XMLBeans).

In the previous example, monitoring applications might be required. The use of XML, as a data encapsulation method, would allow for interception or streaming of work units throughout the system for monitoring and processing. In the future, an application might be required that would allow for the display of items queued for delivery. Utilizing XML to manage the items as they pass through the system would allow for easy integration to a user interface to display/manage a delivery queue.

Integration with External Systems

External system integration allows for

expansion of the functionality of the system in countless ways. For example, a system that currently provides order processing, fulfillment, and shipping could easily be integrated with an inventory control system if XML is used as a common communications method between the two systems. This integration can be done using an XML derivative, such as SOAP or any one of a number of standard XML extensions that have been developed for different industries.

"With the number of XML-based standards being developed, the opportunities for component reusability have become greatly enhanced"

With the number of queuing technologies available to process text-based requests, XML is a natural choice as a format for managing. Two systems can be integrated together using messaging queuing technologies (available through IBM, Microsoft, and other vendors). This connectivity would create what appears to be a fully integrated solution. In this manner, the two systems can be developed in parallel or one of the systems can be developed separately and integration can be completed once each system is fully functional. In this manner, the benefits from the order entry system improve the corporate bottom line, as do the benefits of the inventory control system. But when the two systems are combined together, such as orders are processed, the inventory is updated and when inventory is received, the availability of products is now updated; the combined benefits incorporate the best of both worlds and more.

Cost Benefits and Component Reusability

With the number of XML-based standards being developed, the opportunities for component reusability have become greatly enhanced. There are hundreds of published XML formats for dozens of various industries that provide a basic format for encoding data. These formats allow for development of cross system code. Once the components are created that process format-specific XML, they can then be reused elsewhere in the organization (or used by integrating organizations) to process data of any given type. By reusing these components, the time required to develop, test, and bring a system into production is dramatically decreased resulting in lower overall costs.

Most database servers today support accessing data by utilizing XML. Since XML translates well as a data formatting method for passing data over the Internet, this provides extensive benefits for those who develop components that can be used in systems that are being developed now and in the future. As we look to the future, it is apparent that the need to integrate more systems is becoming more crucial than ever, and XML will clearly provide that ability easily and efficiently.

MMORGAN@EDGEWATER.COM

WRITTEN BY **GIJSBERT IN 'T VELD**

Citrix Systems – A Global Company with a Mission

Poised for strategic and sustainable growth

Citrix Systems, Inc., the global leader in access infrastructure solutions, is one of the top 15 software companies in the world. Headquartered in Fort Lauderdale, Florida, Citrix has offices in 22 countries around the world and conducts over half of its business outside the United States.

With 2003 revenues of $588 million, Citrix has set an aggressive goal of growing to $1 billion in revenues and positioning itself for sustainable growth. Citrix aspires to own the access infrastructure software market, whose potential is estimated at 8–10 billion dollars, and the company is building out its MetaFrame Access Suite of software to do just that. The company also acquired Expertcity, the market leader in Web-based access services. Suffice it to say, whether information is located on the server or the desktop, Citrix solutions provide secure, easy, and instant access to it.

Using Technology to Meet Growth Objectives

An important part of Citrix's strategic growth plan is to use systems to scale the business, rather than merely increasing the workforce. The company wants to reduce costs within its core business operations and redirect those resources to growth areas. "We have been engaged in finding out how we can make the engine at the core of our business run better, so we can invest resources in the emerging areas of our business," said John Gardiner, senior director of Finance at Citrix Systems.

One way to accomplish this is to put automated business processes in place to reduce workload, accelerate cycles, and provide enhanced support to customers, distributors, resellers, business partners, and other groups with which the company interacts. According to Gardiner, "We want to put our people into roles where they can think creatively and add value instead of performing routine functions. By automating these functions, we can ensure maximum consistency and efficiency, while freeing up staff for strategic projects."

The Global Systems Integration Project

Citrix sells its software through an indirect channel of distributors, value-added resellers, and integrators. At the outset of 2004, Citrix had 120 distributors worldwide, all of whom send purchase orders to the company by either fax or email. Citrix employs people to process those orders. These employees must manually check orders for accuracy, distribute them to obtain various signature approvals, and then input them into the company's SAP system. "This is a highly inefficient process that is labor-intensive and time-consuming," notes Gardiner. "At the same time, our distributors are asking where their orders are." Typically, an order works its way upstream from the purchasing agent, to the distributor manager, and then to the order entry team at Citrix corporate headquarters. Once an answer is obtained, the process is reversed, and the order slowly works its way back downstream.

"We want to implement a clearly visible process that people can access on demand, and see what is going on at every stage of the transaction. Trading partners want to see whether Citrix has the order, whether the company has accepted the order, why the order was rejected – if indeed it was – whether the order has been sent to manufacturing, if the order has shipped, what the tracking number is, etc.," explained Gardiner.

Citrix launched a global systems integration project and turned to Microsoft and Covast for assistance. Citrix selected the Covast EDI Accelerator to provide connectivity to its 120 distributors worldwide and to enable the exchange of EDI documents. The Covast EDI Accelerator, in conjunction with Microsoft BizTalk Server, provides the technical foundation for an all-digital infrastructure, offering both EDI functionality and support for XML transactions. The EDI Accelerator sits on top of Citrix's existing solution for BizTalk Server, which interfaces with its SAP enterprise

AUTHOR BIO

Gijsbert in 't Veld is the CTO of Covast, where he oversees the company's overall technical growth and product strategy. He is the principal architect of Covast's product line. Prior to Covast, in 't Veld managed the EDI product development group for the French company GSI (now ADP) s, and as a project manager was responsible for several large B2B integration implementations worldwide. For the last three years, he has worked closely with the Microsoft BizTalk product group designing new product versions.

resource planning solution.

Citrix is currently tackling the purchase order process with a worldwide rollout underway throughout 2004. The system will ultimately process 80% of orders placed by distributors. Next year the company will use that infrastructure to automate even more processes, such as invoice management, partner list authorization, and price changes. For smaller partners who may not have the IT staff or infrastructure to support trading with Citrix through either EDI (electronic data interchange; as supported by Covast) or XML (as supported by Microsoft), Citrix intends to use Internet commerce and enable them to order online.

Citrix recognized early in the evaluation process that this project required a significant effort in terms of EDI expertise. With so many EDI formats available, Citrix wanted a provider that had built its business on developing EDI solutions. During the initial discovery phase of the project, a Microsoft account manager recommended Covast for a full-featured EDI solution that could support purchase orders, advance shipment notices, invoicing, and point-of sale inventory.

"If you look at BizTalk, it is a really good messaging software, and BizTalk Server 2004 is particularly impressive," said Dan Nicolaison, senior manager for Information Technology at Citrix Systems. "The logic to handle all of the hundreds, if not thousands, of EDI documents, between the standard X12 and EDIFACT, is enabled in our system through Covast's solution. The BizTalk-Covast partnership is a great marriage for us," continued Nicolaison.

In the all-digital infrastructure for purchase order processing that Citrix is implementing, the company will get the information it needs in electronic format from its distributors via EDI, XML or the Web. Those electronic documents will get processed through Covast EDI Accelerator for BizTalk Server and sent to SAP. "Through our all-digital infrastructure, we will be able process an order twenty-four by seven, 365 days a year, and every clean order we receive will be processed without anyone having to touch it," said Gardiner. "That is what we are driving to."

"By enabling us to put a better set of headlights on our business, this initiative will enable us to manage our business in a much more proactive way"

But how will Gardiner's team deal with human error, such as entering the wrong product number or forgetting the contract number? The real benefit to Citrix is that employees who are currently focusing on processing routine orders will have the time to help distributors get those orders right. The order processors of today will deliver more value-added service, as well as better customer service, because the Covast-BizTalk system has automated a manual function. "This will fundamentally help drive the business to the 1 billion dollar goal," noted Gardiner.

Citrix did put some numeric stakes in the ground as to precisely how much value the global systems integration project would deliver for the company, both in terms of cost savings and productivity increases. Savings stem from lower turnaround time on orders, whereas increased productivity will grow top-line revenue across the company. "The tangible benefits we tied to deploying this system are in the millions of dollars," explained Gardiner.

Intangible benefits of the system are numerous for Citrix's partners. With automated ordering, if there's a problem with an order, the partner knows of it immediately. In addition, the product ships to partners the same day. "Partners get a lot more visibility through functionality like order confirmation, advanced ship notices, and shipment confirmations," said Nicolaison.

Beyond the purchase order benefits for Citrix, the all-digital infrastructure will enable point-of-sale reporting and inventory reporting on distributors. As Citrix brings partners online with EDI and XML, Gardiner and his team will have distributor data available on a daily basis and the Citrix team can see exactly what is happening in real time. At the close of each business day, the team will know whether demand is going up or down, and how to adjust and react to trends and changes. The system will show Gardiner's team, and all of Citrix management, whether marketing initiatives are working and whether new products are taking off. "By enabling us to put a better set of headlights on our business, this initiative will enable us to manage our business in a much more proactive way," concluded Gardiner.

Architecture: At a Glance

Citrix Systems runs on a complete Microsoft environment. A Microsoft operating system is running on BizTalk Server 2003. Covast EDI Accelerator runs on BizTalk Server. BizTalk Server interfaces with Citrix's SAP enterprise resource planning system.

GIJSBERT.INTVELD@COVAST.COM

WRITTEN BY **CHRIS RANDLES**

Managing Engineering Information

Extend the productivity of the enterprise

It's virtually axiomatic: technology innovations first boost personal productivity then group productivity. The PC, for example, first helped individuals automate writing, accounting and personal organization functions, then spawned group productivity through networks, the Internet and e-commerce. Today's goal is to harness other personal productivity gains that technology has delivered and transfer them to entire organizations.

We in the design engineering industry are no exception. We are particularly intent on stretching personal productivity to the workgroup and the enterprise. This is especially valuable as engineering organizations decentralize product development and manufacturing, often offshore, in search of lower costs and around-the-clock productivity. To support this decentralized collaboration, information must flow easily, coherently and instantly among different kinds of systems.

This is why XML is a powerful force promoting robust information exchange. We see XML making an especially dramatic impact on productivity and collaboration in the area of innovation, specifically in its ability to enhance the management of engineering information.

XML's premise is that every piece of information, even the most granular piece of data, can for the first time include metadata that explains its purpose and significance. As a result, virtually any data item can be captured, managed, and leveraged. Web pages – essentially vehicles for representing real data – are giving way to documents that contain and reveal and transmit meaningful, structured content.

XML's simple text-based approach, combined with the power of specialized schemas eliciting meaning from otherwise unstructured data, is enabling engineers, R&D teams, and entire product development groups to exchange information more quickly. As a result, they can participate more deeply in one another's business processes and automate these processes without overhauling applications to accept every partner's unique data format. Fueling this evolution is the arrival of new XML vocabularies. The design engineering market – a market driven in part by mathematical calculations – is again no exception to this trend.

The Lifeblood of Product Development: Engineering Methods and Values

Calculations and their results, the lifeblood of engineering organizations, are to an engineer as words are to an author, paintings to a museum, or money to a banker. Calculations power innovation, whether a civil engineer at Bechtel is assessing the structural integrity of a bridge, a mechanical engineer at Motorola is developing a durable mobile phone, or a chemical engineer at Procter & Gamble is determining the effect of a new detergent on cloth.

Too often organizations find a way to preserve the result of a critical calculation but lose track of the assumptions and data that contributed to it. And sometimes, they even fail to preserve the result. XML promises to help capture this elusive information and provide what could be the biggest step forward to date in engineering collaboration. It will give every product design – and every engineering method and value underlying it – meaning and traceability, enabling on-time product development, product quality, regulatory compliance, and auditability. This is calculation management.

Redefining Knowledge Management in the Engineering Enterprise

Engineering enterprises have traditionally lacked powerful tools, XML-based or otherwise, to manage the methods and values underlying their designs. Calculations in the engineering-based enterprise are frequently performed in spreadsheets, or using programming languages. In most organizations, this work is scattered across desks, personal hard drives, document management systems, and file cabinets. Although these methods and data are

AUTHOR BIO

Chris Randles is chairman, president and CEO of Mathsoft, whose software is now used by 90%t of Fortune 1000 companies, more than 500 government agencies, more than 2,000 colleges and universities, and 1.5 million engineers in more than 50 countries. Prior to joining Mathsoft, Randles was vice president of marketing at Software Publishing Corp., where he was in charge of strategy, marketing and promotion for Software Publishing's products worldwide. Earlier, he served as director of marketing — and a founder manager — of the company's international division.

valuable corporate assets, they are rarely treated as such. The focus is often on the result of a calculation and not the method of reaching it. The harm in failing to properly manage calculations is needless redesign, sometimes disastrous errors, and always lost productivity and lost revenue. For that reason, organizations are waking up to the need to more strongly manage this aspect of the product development process.

Unfortunately, when equations are represented most accessibly in printed materials and handwritten notes, they are static pictures lacking meaning in the electronic world. In spreadsheets, formulas are invisible (i.e., embedded in the cells), as are functions coded in a programming language. The beauty of XML is that it can make calculations usable and visible without embracing a proprietary file format. All the elements of a calculation are tagged in one language, XML, and ready for other applications to consume or publish directly into any selected document format. You are embracing a method for encapsulating meaning in a way that can be interchanged with any individual, group or machine.

XML Vocabularies Describe Calculations

Recognizing XML's potential for calculation management, the engineering industry is developing several schemas to capture engineering information for better product development. UnitsML, for example, is a standard describing units of length, area, volume, mass, and other measures that my company is developing in cooperation with the National Institute of Standards and Technology (NIST). No longer is a number in an electronic document at best a known quantity of mysterious unknown units… Seven what? Meters? Liters? Rads? Mph? Units can now be specified in XML and searched, indexed and retrieved across the Internet as such.

My company has developed and published the first comprehensive XML-based schema for engineering, the Mathcad XML Information Architecture (XML-IA). Although XML schemes for math already exist, this for the first time accounts for the peculiarities of applied math and engineering information, including parameters, units, and results. It is an open, nonproprietary data model that incorporates UnitsML and is readable by human engineers and their software, whether it's Mathcad calculation software or any other vendor's design, document management, database, or product lifecycle management (PLM) application. This is not just for my company's software; it's for the benefit of product development wherever it takes place, across multiple companies and systems, whichever they choose to use.

"…organizations are waking up to the need to more strongly manage this aspect of the product development process"

Mathcad XML will further support calculation management by making engineering information traceable through the enterprise via a central XML repository. XSL Transformations (XSLT) will translate documents for databases, publication-quality reports, code developers, and search engines. Companies can easily repurpose this engineering information for calculation dependency analyses and design-change appraisals.

These and other XML-related developments will help engineering organizations better search for and retrieve numbers, calculations and results from within their organizations or partner networks. They will be able to publish their critical values in a wide variety of formats with greater control over the look and feel. They will be able to track calculations across any number of projects yet hide or lock them when necessary. They will more easily integrate calculations and results with organization-wide business processes and, through Web services, automate their interaction with upstream and downstream applications.

Specific Scenarios

With XML, engineers will be able to integrate calculations into automated business processes across teams and among discipline "silos." An engineer, for instance, could import procurement data into a Mathcad worksheet and let calculation results produce SmartSketch CAD drawings, Excel spreadsheets for financial analysis, and Adobe PDF documents (or any text-based format because XML is highly flexible and structured) for reporting, all with the push of a button.

Engineers will be able to identify at a glance the origin of a particular value or calculation regardless of how many times it is cut and pasted to other documents. A design engineer, for example, could publish a worksheet of constants to be used as baseline data for all projects involving a particular vendor, creating an auditable trail of numbers throughout the project.

Organizations will be able to get concrete answers to commonly asked questions like:

- Did we use the current requirement for cargo weight or the old requirement in this analysis?
- Can we give our vendors access to that calculation?
- Does the delivered document match the analysis that was actually performed?
- Which design points depend on the yield strength of ASTM316l steel, and where did we get that number?
- I'm sure someone must have done this calculation before.
- I hope we got the units right!

The ability to answer questions like these separates great engineering organizations from average ones and makes the difference between efficient product development and failing to cultivate the potential market for your products. Engineering organizations that harness the full power of XML for calculation management can nail these questions every time and enjoy on-time product development, higher product quality, painless regulatory compliance and easy integration with enterprise applications.

The personal productivity that computing delivered in the last decade will be extended to the enterprise, and results will be impressive.

CRANDLES@MATHSOFT.COM

Neon Systems Launches Shadow z/Services

(Sugar Land, TX) – NEON Systems, Inc., a provider of enterprise-class mainframe integration, has released Shadow z/Services, a SOAP-based mainframe integration solution that allows organizations to rapidly transform CICS, IMS, and IDMS applications into Web services.

Shadow z/Services' easy-to-use, single-step configuration can accelerate the development of mainframe Web services by eliminating the additional programming typically required with other mainframe Web services products.

Shadow z/Services allows developers to easily expose mainframe application logic and application presentation environments as Web services. Utilizing the product's Introspection technology, developers are able to dynamically parse the application program and screen definitions and generate the WSDL.

www.neonsys.com

Altova Announces End-to-End Suppoft of XSLT 2.0, XPath 2.0, and XQuery

(Beverly, MA) – Altova, producer of XMLSpy and other leading software development tools, has introduced version 2005 of its product line to enable accelerated application development and data integration. Version 2005 includes the industry's first native, production-ready implementation of the XSLT 2.0 and XPath 2.0 specifications across the entire product line, as well as native support for XQuery 1.0. In addition, version 2005 adds powerful new features to all of Altova's products, including graphical schema management, automated function building, Eclipse integration, and relational database content editing.

www.altova.com

Stylus Studio 6 Enables XML Data Integration with Powerful New Tools

(Bedford, MA) – Stylus Studio , a developer of XML development and productivity tools, has shipped Stylus Studio 6 XML Professional Edition, the latest major release of its XML IDE. This release adds new productivity tools and enhancements that simplify the development of standards-based XML data integration applications. Included is enhanced support for XSLT 2.0, XQuery, XML Schema, and XML Mapping. New in Stylus Studio 6 is Convert to XML, a flexible utility for converting legacy or nonrelational data into any XML format that can be used anywhere in Stylus Studio you would use XML.

www.stylusstudio.com

Forum Systems Announces Integration with AmberPoint for SOA Life Cycle Management

(Salt Lake City, UT) – Forum Systems, a provider of trust management and threat protection Web services security solutions, has announced a technology integration with AmberPoint to advance the life cycle management of Web services and service-oriented architectures (SOA). By activating an AmberPoint software agent on Forum Web services security products, enterprises can ensure real-time connectivity and interaction with AmberPoint's Web services management solutions, including Service Level Manager, Exception Manager, and AmberPoint Management Foundation. The joint security and management solution has already been selected by technology leader Motorola to speed time to market.

www.forumsys.com, www.amberpoint.com

Enterprise Software Companies Select Motive Profile to Build Self-Management Capabilities Applications

(Savannah, GA) – Motive, Inc., a provider of management software, has announced that three new companies are using its application configuration management product, Motive Profile: Callidus Software, Lawson Software, and Primus Knowledge Solutions. Motive Profile will allow these companies' customers to intelligently discover, capture, and compare changes in product configurations, helping their application environments to become more self-managing.

In addition to these customers, Motive recently announced that BEA Systems and PeopleSoft have licensed core components of Motive Profile to provide customers with enhanced visibility and control over their respective enterprise applications.

By automating traditionally manual, labor-intensive configuration management processes, Motive Profile allows application administrators to pinpoint changes in an application environment so that they can more effectively plan product changes, optimize product performance, and minimize downtime.

www.motive.com

GT Software Announces Ivory Web Services Solution

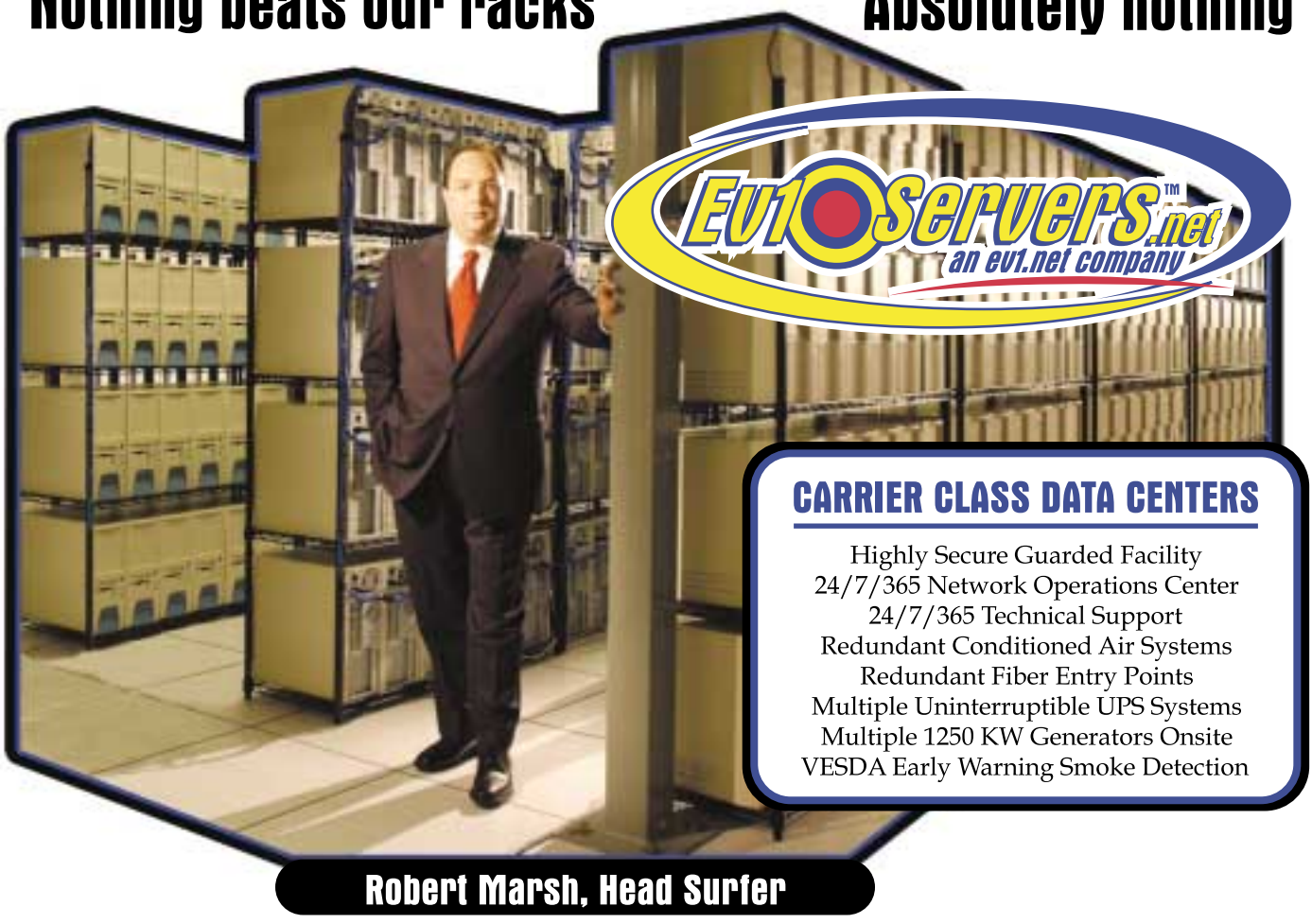
(Atlanta) – GT Software, Inc., a provider of enterprise mainframe integration solutions, has released Ivory Web Services. Ivory is a new product designed and written from the ground up as a Web services solution that complies with WS-I standards. It provides the power of a "programmatic solution" without the programming, for rapid, cost-effective development and deployment of Web service applications. With Ivory, developers can include legacy applications into their service-oriented architecture (SOA). It allows companies to graphically generate Web services for mainframe applications and data, and consume other Web services.

Ivory Studio is a PC-based development application that lets organizations graphically create and publish Web services from existing mainframe assets. Ivory Server is a high-performance SOAP server for seamless Web service deployment. The solution enables enterprises to generate Web services and WSDL without the long development time and high costs associated with writing new programs.

www.gtsoftware.com

(L to R) Joe Ganem, president & CEO, GT Software; Gail Schultz, executive editor, Web Services Journal; Miles Silverman, vice president of advertising sales, SYS-CON Media; and Sal Apollo, vice president of sales, GT Software. GT Software visited SYS-CON to discuss their new initiatives in the Web services field.

Nothing beats our racks Absolutely nothing



Robert Marsh, Head Surfer

CARRIER CLASS DATA CENTERS

Highly Secure Guarded Facility
24/7/365 Network Operations Center
24/7/365 Technical Support
Redundant Conditioned Air Systems
Redundant Fiber Entry Points
Multiple Uninterruptible UPS Systems
Multiple 1250 KW Generators Onsite
VESDA Early Warning Smoke Detection

START YOUR OWN WEB HOSTING BUSINESS TODAY!

from **$299*** Instant Activation!

Dedicated Server

Dual Xeon 2.4 GHz
2 GB RAM • 2 x 73 GB SCSI HD
Remote Console • Remote Reboot
1700 GB Monthly Transfer Included

Over 20,000 Servers!

1-800-504-SURF | ev1servers.net

PLESK7 RELOADED
Preferred Control Panel

IP Compliant. Price subject to change. Quantities Limited.
*Per month. Set-Up fees apply. See web site for complete details.

DB2 Key

1. **Player attempts 30-foot chip.**
2. **Operator measures distance.**
3. **Stats entered into PDA.**
4. **SHOTLink truck transmits data.**
5. **Broadcaster broadcasts a "birdie!"**

MIDDLEWARE IS IBM SOFTWARE. Fans, broadcasters, even players, are accessing every shot at PGA TOUR® events online – in real time. The scalable multiplatform technology of IBM DB2 integrates and manages information, allowing SHOTLink, the PGA TOUR's ball-following technology, to uplink and downlink every shot, run all the numbers and tell the entire story – hole by hole.

Middleware for the on demand world. Learn more at **ibm.com**/middleware/information ON DEMAND BUSINESS™