SYS-CON

NOW INCORPORATING XML JOURNAL

WebServices JOURNAL

.NET J2EE XML

July 2004 Volume 4 Issue 7

Application Server Architecture and BPEL

Promises and Challenges...pg 22

Quality Management for Web Services

The requirement of interconnected business...pg 30

Web Services Integration Brokers and Enterprise Application Integration

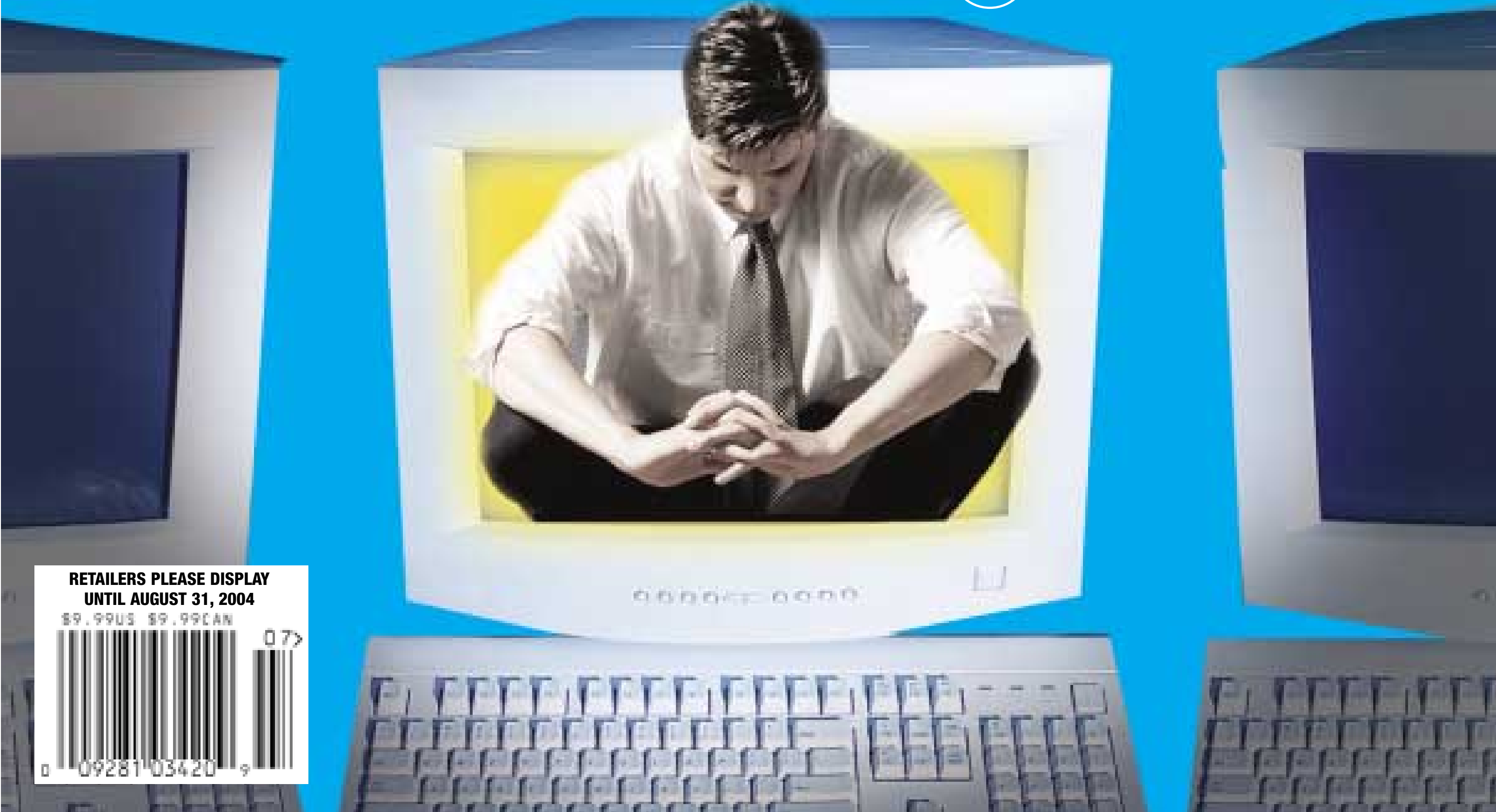
Preparing for the future...pg 42

ENTERPRISE ARCHITECTURE

Making One Out of Many

Build tomorrow's foundation today

pg 12



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

Introducing OpenLink Virtuoso

A CROSS-PLATFORM Universal Server that transparently integrates disparate application databases and disparate applications across your enterprise. It provides a Virtual Database, XML Database, SQL Database, Web Application Server, and Web Services Platform as part of a **single-server solution.**

Empower your organization today, download a FREE evaluation copy from www.openlinksw.com/virtuoso or call us on 1 800 495 6322

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

Programming Languages
Any .NET bound language, Java, C/C++ etc.

Protocol Support
SQL, ODBC, JDBC, XML, XSL-T, XQuery, XPath, HTTP, WebDAV, NNTP and POP3

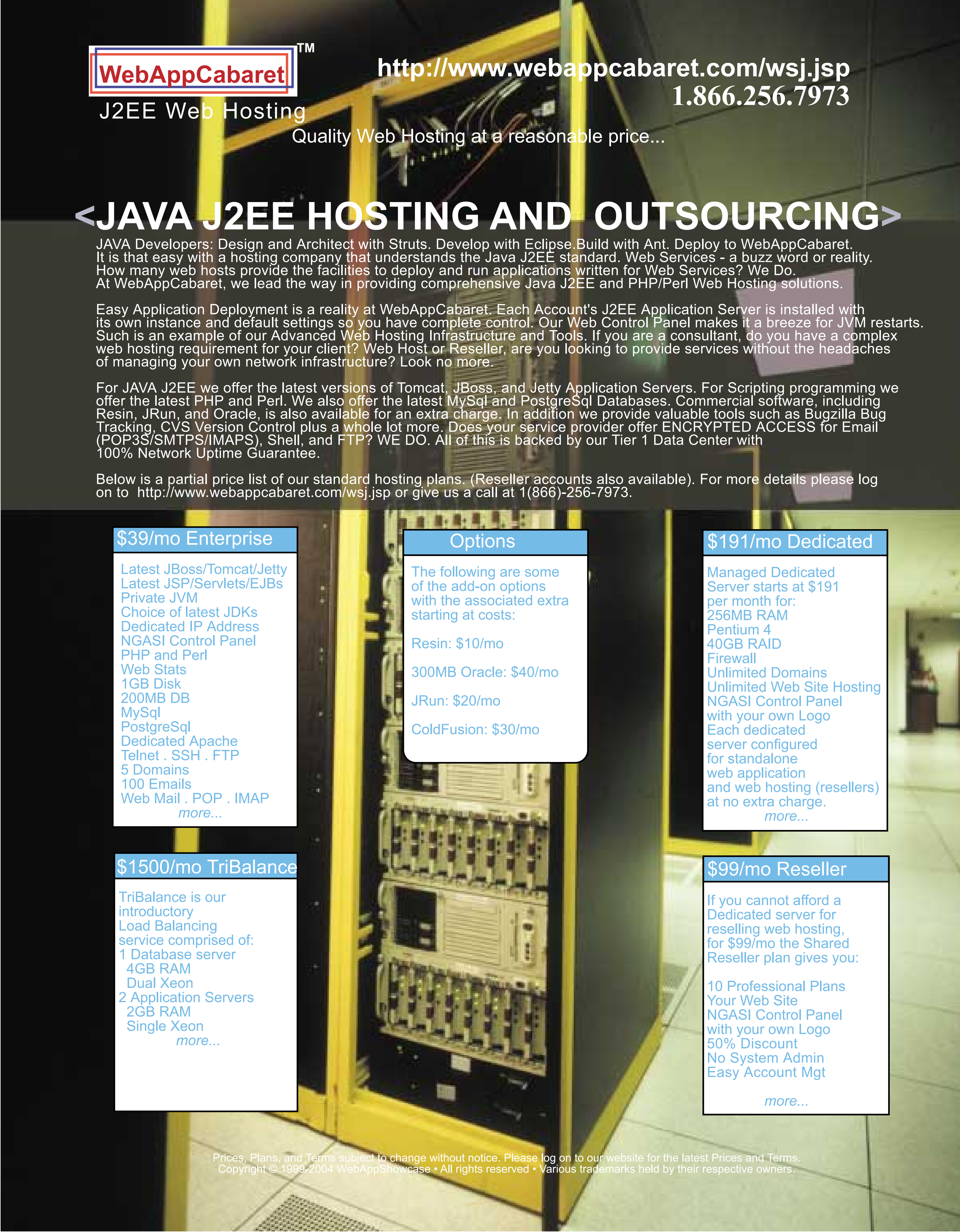
Web Application Hosting
ASP.NET, Java Server Pages, PHP, Perl, and Python

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

www.openlinksw.com/virtuoso

OPENLINK®
SOFTWARE



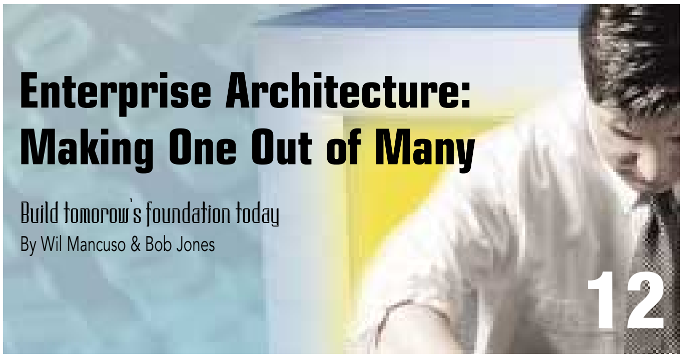
JULY 2004 Volume 4 Issue 7

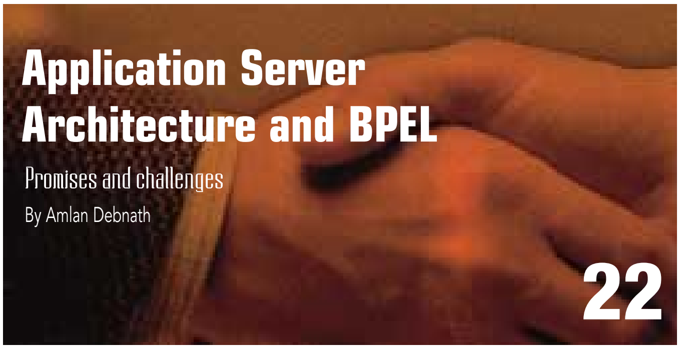
WebServices JOURNAL .NET J2EE XML

XML JOURNAL

Inside WSJ

WSJ: Features





From the Editor

Industry Commentary

WSJ: technology

WSJ: Success story

WSJ: Product Review

WSJ: Web Services@Work

WSJ: Supply Chain

The Problem with Most Team -Wide Practice Implementations

Most development teams recognize that if they want to improve software quality, their entire team should implement practices such as coding standards enforcement and unit testing. However, many teams don't recognize that implementing these practices requires more than simply purchasing the tools that facilitate the practices and then installing those tools on every developer's workstation.

Although it would be nice if tools alone could drive a team to implement a practice, it is not possible. Tools don't operate in a vacuum. Truly implementing a practice in a team requires more than just tools. It also requires the team culture, workflow, and supporting infrastructure required to embed the practice into the team's development process.

Teams that attempt to implement practices with tools alone typically do not achieve the expected quality improvement benefits. For instance, assume that a team tries to implement the coding standards enforcement practice by only purchasing a coding standards enforcement tool and asking each developer to use that tool. Over time, it's likely that most of the coding standard violations will remain in the code. Why? Without additional team-wide support for the coding standards enforcement practice, developers typically become overwhelmed by the number of problems reported and do not know how to approach them. The tool helps the team members recognize the faults in their code, but if the developers do not have the necessary support, the faults remain and the code quality does not significantly improve.

The Parasoft AEP Methodology details one strategy for embedding best practices into a team's development process. To learn how this methodology works, visit **http://www.parasoft.com.**

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

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

Introducing Parasoft® SOAPtest®

The simple fact is, no other Web service testing tool can do what SOAPtest can do. Or do it as fast and reliably. From ensuring functionality and interoperability to telling you how your Web service is performing, SOAPtest automates all critical testing processes across the entire lifecycle of your Web service.

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

For Downloads go to www.parasoft.com/soaptest
Call 888-305-0041 x3303 or email: buzz@parasoft.com

PARASOFT®
We make software work.™

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

Features

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

Benefits

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

Protocol Support

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

Platforms

Windows 2000/XP
Linux
Solaris

Contact Info:

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

www.parasoft.com

How to Make Money with Web Services

By Bob Brauer
bob.brauer@strikeiron.com

A new way to work with Web services - build them and they will come?

The benefits of Web services are well understood – lower integration costs, maximum reusability, faster deployment, more automation, easier to work with new partners, and so on. You can build them but that does not necessarily mean that people will use them or that you can make money on them. Why? And more importantly, how?

What drives Web services utilization and return on investment?

Here are some key points to consider:

- Accessibility and ease-of-use
- Billing and accounting
- Subscriptions and trials
- Availability, reliability, and security
- Promotions and pricing

What tools and services do you need to commercialize your Web services?

- Tools to improve understanding and use.
- Commerce capabilities to manage subscriptions, accounting, billing, payments, account management, etc.
- Ability to manage free trials and convert trial users to subscribers.
- Service levels that ensure availability, reliability, and security.
- Knowledge about acceptable pricing structures based on value of the data, process, accessibility, and performance.
- A way to deliver and promote them to the appropriate target audience.

Creating a new revenue channel

Your Web services provide value and you need to be reimbursed for that value. However, before that can happen, you need a distribution channel with the infrastructure to publish and sell your Web services. This channel must take care of issues such as delivery, account set up, billing and collections, marketing, and customer support, to free you of having to make that investment.

Search
Tools/Services
Access
Monitor
STRIKE IRON
Web Services
Business Network
Trusted/Reliable
Subscribe
Analyze
Understand

The answer is Premium Web Services

Premium Web Services provided by StrikeIron and available through the StrikeIron Web Services Business Network™ (WSBizNet™) are the answer.

Whether you are a Web service provider wanting to commercialize your Web services or a company wanting to find a way to manage your Web services without the infrastructure investment, StrikeIron will do the work for you, and you make money in the process.

StrikeIron Premium Web Services are subscription-based Web services that deliver "live" data and functionality over the Internet via XML using a dynamic SOAP interface. Premium Web Services are supported by the commercial services and easier-to-use tools of the StrikeIron WSBizNet to remove barriers to purchase.

Premium Web Services benefits

- An alternate sales channel to reach more customers for minimal cost.
- Web services hosting and delivery services in a secure environment.
- Pricing, billing, collections, and support.
- Reporting for easier maintenance and tracking.

The StrikeIron Web Services Business Network

The StrikeIron WSBizNet is a subscription-based online network of services and tools for easier and faster utilization of Web services by business users and developers.

The WSBizNet allows anyone, with minimal experience, to quickly find, understand, and utilize Web services for accessing information, integrating with existing applications, and assembling new applications.

StrikeIron WSBizNet benefits

- Service levels to meet your business requirements.
- Integrated online StrikeIron Web Services Analyzer for faster understanding of Web services.
- Integrated online Knowledge Base makes it easier to understand and work with your Web services.
- Prime positioning in the WSBizNet.
- Increased visibility in one of the largest Web services business directories.

Delivering Web Services to the World!

For Free Trials and more about how to make money with the StrikeIron Web Services Business Network and Premium Web Services, please visit our Web site at www.strikeiron.com.

StrikeIron, Inc.
+1.919.405.7010
sales@strikeiron.com
www.strikeiron.com

WebServices JOURNAL
.NET J2EE XML

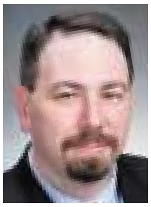
EDITORIAL
EDITOR-IN-CHIEF
Sean Rhody sean@sys-con.com
XML EDITOR
Hitesh Seth
INDUSTRY EDITOR
Norbert Mikula norbert@sys-con.com
PRODUCT REVIEW EDITOR
Brian Barbash bbarbash@sys-con.com
.NET EDITOR
Dave Rader davidr@fusiontech.com
SECURITY EDITOR
Michael Mosher wsjsecurity@sys-con.com
RESEARCH EDITOR
Bahadir Karuv, Ph.D Bahadir@sys-con.com
TECHNICAL EDITORS
Andrew Astor aastor@webmethods.com
David Chappell chappell@sonicsoftware.com
Anne Thomas Manes anne@manes.net
Mike Sick msick@sys-con.com
EXECUTIVE EDITOR
Gail Schultz gail@sys-con.com
MANAGING EDITOR
Jennifer Van Winckel jennifer@sys-con.com
EDITOR
Nancy Valentine nancy@sys-con.com
ASSOCIATE EDITOR
Jamie Matusow jamie@sys-con.com
ASSISTANT EDITOR
Torrey Gaver torrey@sys-con.com

PRODUCTION
PRODUCTION CONSULTANT
Jim Morgan jim@sys-con.com
LEAD DESIGNER
Richard Silverberg richards@sys-con.com
ART DIRECTOR
Alex Botero alex@sys-con.com
ASSOCIATE ART DIRECTORS
Louis F. Cuffari louis@sys-con.com
Tami Beatty tami @sys-con.com

CONTRIBUTORS TO THIS ISSUE
Scott Barstow, Brian Barbash, Scott Bushey, Amlan Debnath,
Fred Down, Tyson Hartman, Bob Jones, James Kobielus,
Mike Lehmann, David Linthicum, Wil Mancuso,
Peter Moskovitz, Sue Vickers, Vijay Tella, Jay Weiser,
Kristin Weller, Matthew Zager

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

Leveraging EAI



WRITTEN BY
SEAN RHODY

Roughly last year at this time, as we covered the convergence of EAI and Web services, many people were asking about the ultimate survival of EAI as a software solution. In one sense, this was understandable, as the technologies that make up Web services provide significant amounts of functionality similar to that provided by EAI products, while achieving greater interoperability.

Still, it was fairly ingenuous to suggest that EAI was about to go the way of the dodo bird. And a year later, we still see a strong presence in the market from EAI vendors. This is true for several reasons.

> “it was fairly ingenuous to suggest that EAI was about to go the way of the dodo bird”

First of all, besides technology, EAI vendors supplied knowledge. Consider a new programmer who looks at an API to accomplish some task. Sure all the services may be exposed as method invocations, but do you know which service to invoke to create the application that you want? Are some services meant only for aggregations, and should not be used directly? Do some services have defects that make them challenging to use? Is there a better way to accomplish the task, given several services that provide similar functionality? All of these questions require knowledge, not just information about the API. EAI vendors have concentrated on examining the most critical applications in use in many industries, worked with their APIs, and concentrated that special business knowledge in their products. This extra knowledge is necessary, regardless of whether or not a Web services approach is used to access the services provided by the underlying application.

Additionally, EAI vendors have adapted to the changing times and have embraced Web services as a key delivery mechanism for their knowledge. One of the biggest impediments to EAI had always been the proprietary nature of the products, which prevented one system from working with another and limited the synergy that could be created. Web services provides widespread, vendor-neutral connectivity that alleviates the need for proprietary solutions. Newer standards, some authored in part by representatives from EAI vendors, will also continue to standardize aspects of EAI. In particular, the specifications for transactions and coordination will become a key component in the further merger of Web services technology into EAI. And the EAI vendors will likely be leaders in the release of new products based on their previous proprietary transaction and coordination engines, which will implement these new standards. Much like what happened in the Java world, vendors will compete by implementing standards, then achieving success in the field by product differentiations, such as ease of use, speed, and reliability.

Undoubtedly, the rise of Web services has also had a tremendous impact on EAI. Interapplication communication, once an extremely difficult interaction that formed part of the basis of EAI in the first place, is now much easier to accomplish, and no longer requires the custom adapters that made up a typical EAI product. This is forcing EAI vendors to become Web services vendors, and move their products up the evolutionary ladder in order to survive the commoditization of a portion of their services. Change is never easy, but there is of course the upside – now more and more software can be governed by an EAI engine, not just those for which adapters have been written, but a broad spectrum of applications, without the need to create custom adapters. This allows the EAI vendors to focus their attentions on the task of making business sense out of the offered services, not just plumbing. And that's where their value add is still intact.

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

LINUXWORLD® CONFERENCE & EXPO

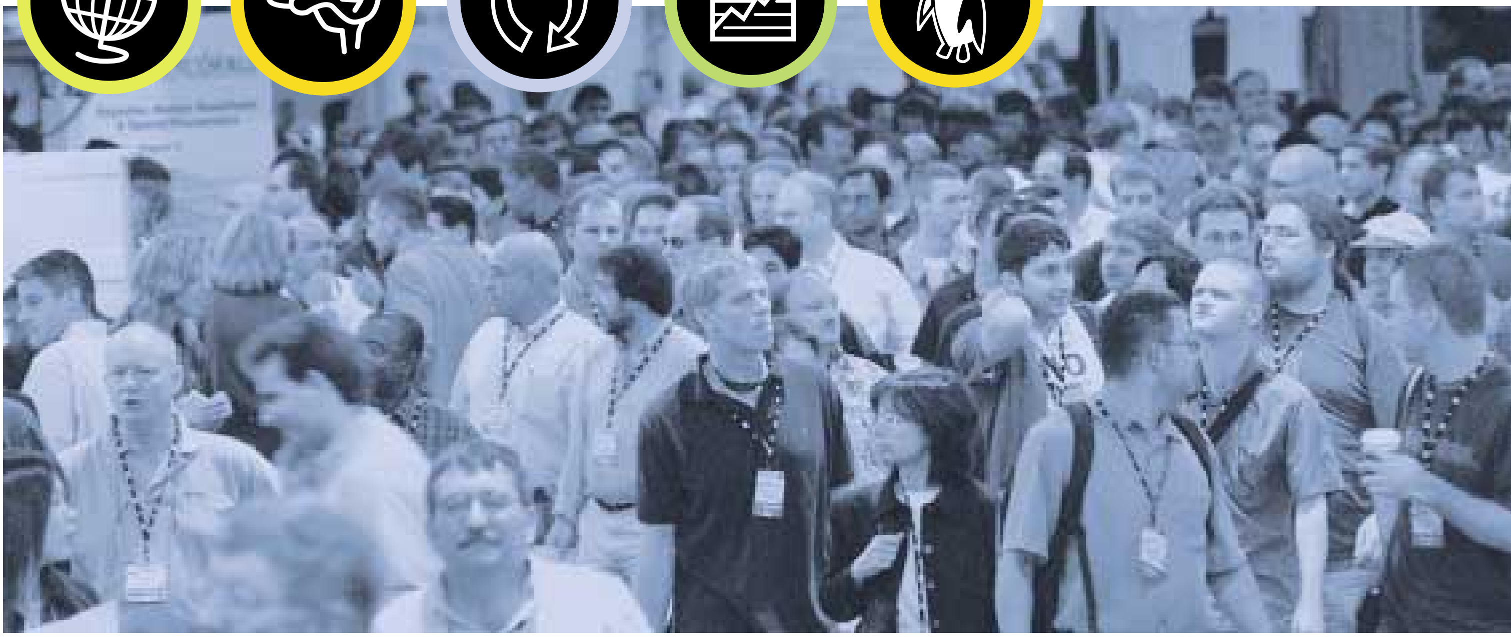
CONFERENCE: August 2 – 5, 2004 EXPO: August 3 – 5, 2004

MOSCONE CENTER • SAN FRANCISCO, CA

LinuxWorld Conference & Expo

is the #1 marketplace for companies that sell, market or promote open source based products, services, applications and solutions. Experiencing astounding growth, it's the largest gathering of open source professionals in the world.

Where
OPEN MINDS
Meet



CORNERSTONE SPONSOR

ORACLE®

PLATINUM SPONSORS

ca Computer Associates® DELL hp invent IBM intel. Novell. Sun microsystems

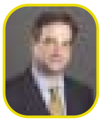
Keynote Speakers

Matthew Szulik,
*Red Hat Chairman,
Chief Executive Officer
and President*
**Tuesday, August 3
9:15 am – 10:00 am**

Nick Donofrio,
*Senior Vice President,
Technology and
Manufacturing, IBM Corp.*
**Wednesday, August 4
11:00 am – 11:45 am**

Martin Fink,
*Vice President, Linux
Business, Enterprise Storage
and Services Business Unit,
Hewlett-Packard Company*
**Tuesday, August 3
11:45 am – 12:30 pm**

Alfred Chuang,
*BEA Systems Founder,
Chairman and CEO*
**Wednesday, August 4
2:30 pm – 3:15 pm**



Michael Steven Rocha,
*Executive Vice President,
Global Support Services
& Platform Technologies,
Oracle Corporation*
**Tuesday, August 3
2:45 pm – 3:30 pm**

Attend LinuxWorld and...

- Evaluate the **latest Linux and open source products and solutions** showcased by the industry's top companies
- Discover **real business solutions** for real business problems
- Network with peers and gain **valuable insight** from open source industry leaders
- Participate in LinuxWorld's world-class educational conference and realize Linux as more than an operating system, but as **a world of applications**
- Learn how Linux can **save your company time and money**

Register Online With Priority Code: D2501

WWW.LINUXWORLDEXPO.COM

IDG
WORLD EXPO

WebServices JOURNAL
.NET J2EE XML

PRESIDENT AND CEO
Fuat Kircaali fuat@sys-con.com
VP, BUSINES S 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
CONFERENCE MANAGER
Lin Goetz lin@sys-con.com

CUSTOMER RELATIONS/JDJ STORE
CIRCULATION SERVICE COORDINATORS
Shelia Dickerson shelia@sys-con.com
Edna Earle Russell edna@sys-con.com
Linda Lipton linda@sys-con.com

SYS-CON.COM
VP, INFORMATION SYSTEMS
Robert Diamond robert@sys-con.com
WEB DESIGNERS
Stephen Kilmurray stephen@sys-con.com
Mathew Pollotta mathew@sys-con.com
ONLINE EDITOR
Lin Goetz lin@sys-con.com

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

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

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

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

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

SYS-CON MEDIA

Looking Beyond Vendor-Driven Web Services

WRITTEN BY
KRISTIN WELLER

Every emerging technology moves through predictable phases of adoption. Throughout these phases there are equally predictable signposts indicating whether the new technology will take root in the mainstream, or be relegated to the sidelines. No clearer sign of an innovation's merit can be found than when leading enterprise resource planning (ERP) vendors begin tapping into that technology's promised capabilities. The evolution of Web services and service-oriented architectures (SOAs) have risen from once being considered bleeding-edge technologies to forming the strategy cornerstones of today's leading ERP vendors.

But the reality is that the ERP vendors, too, are at an early stage of adoption. The initial strategy is to use Web services to move away from their traditional monolithic architecture by exposing component business processes as Web services. This conceptually allows these ERP processes to be used with other Web services within the enterprise, but such an approach relies heavily on an SOA infrastructure to provide the necessary interoperability between Web services from many differing sources. Companies must perform the due diligence to ensure that their ERP vendor is sufficiently motivated to provide an SOA infrastructure that can effectively leverage Web services from all sources – not just those from the ERP vendor – as the consequences of selecting a potentially limiting, vendor-specific approach to integration and SOA can be severe.

Enterprise integration vendors, on the other hand, have their history in application interoperability as their highest priority. The more progressive of these companies have taken the lead in developing and embracing SOA and Web services infrastructure technologies. Indeed, certain integration platforms have been built such that everything they touch – every application and every data source – can be implemented as a service, so that supporting Web service standards is simply a matter of a snap-in protocol. This capability effectively exposes all IT assets as Web services, even those legacy assets not designed with Web services in mind. The result is the best of both worlds – an SOA infrastructure that functions effectively as both a producer and a consumer of Web services.

Another important consideration is the evolution of standards. Web services standards are still fairly nascent, and while most vendors support widely accepted standards like XML and SOAP, there are still a number of gaps in the standards yet to be finalized. Progressive enterprise integration vendors have stepped in to fill these gaps within their SOA platforms to provide the enterprise-class service necessary to run mission-critical processes. These solutions are architected so that new standards can easily be plugged into the platform as these standards are released. A standards-based, vendor-neutral platform that readily addresses the evolving SOA capabilities prevents organizations from being locked into any specific ERP vendor's technology and ensures that emerging standards will be properly implemented.

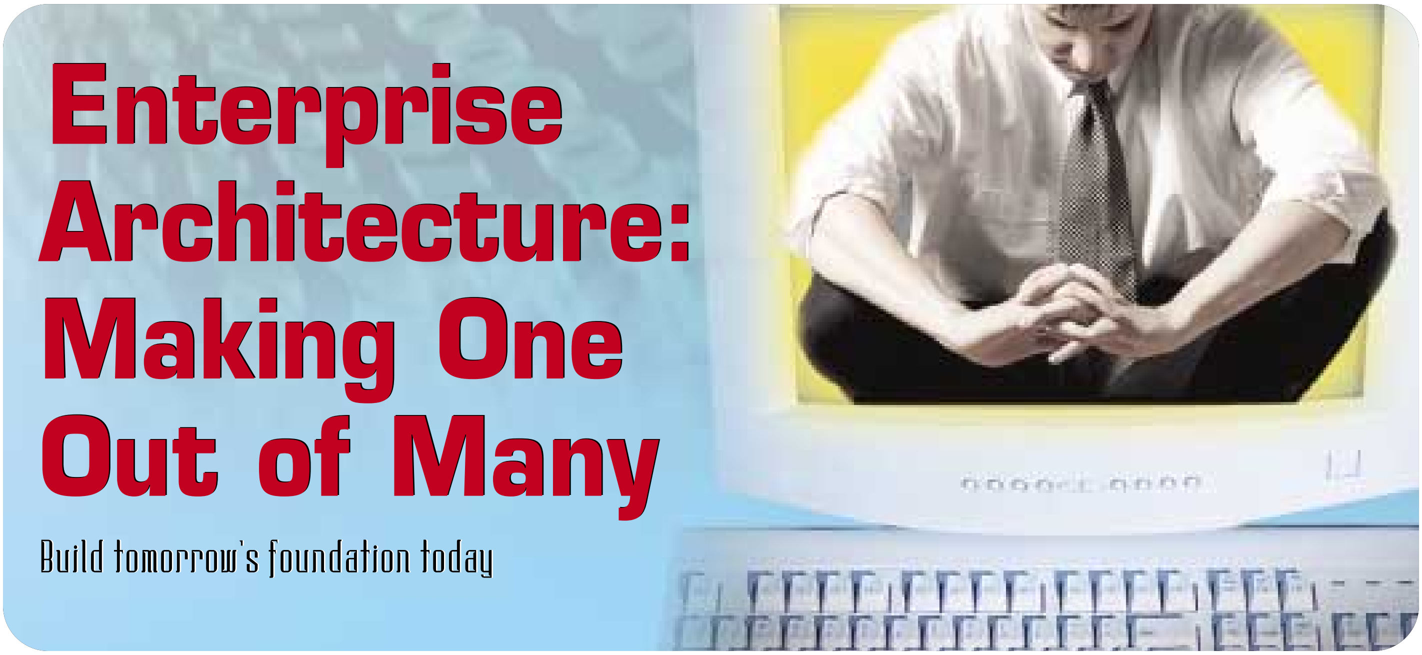
Without a proven SOA platform, companies may find themselves once again building point-to-point integrations between Web services creating a whole new generation of fragile legacy applications. Point-to-point connections – even if they are Web services–based – are slow and costly to develop, difficult to maintain and inflexible. Each time a new connection is added, the complexity of the application grows exponentially. A platform with mature SOA capabilities provides the tools to discover, orchestrate, and deploy Web services as coordinated business processes in a graphical, intuitive development environment. Such platforms address vital issues such as security, availability, and reliability, thereby eliminating the problems inherent with the point-to-point approach.

The decision by ERP vendors to embrace Web services and SOAs may deliver significant benefit. However, it is vital that companies consider how to best achieve the maximum return on their investment in Web services, secure the future of their IT assets, and maintain corporate agility. The answer to that challenge lies in developing an enterprise-wide, vendor-neutral strategy for Web services and SOAs. Through such a strategy, business processes can be streamlined, redundant processes eliminated, and valuable blocks of programming can be reused for many different purposes.

About the Author

Kristin Weller is the executive vice president of product development at webMethods, Inc.

■■■ kweller@webmethods.com



Enterprise Architecture: Making One Out of Many

Build tomorrow's foundation today

■ Service-oriented architectures are emerging quickly as the commercial world's answer to a flexible, standards-based infrastructure. In the world of government IT, enterprise architecture through enterprise integration is its equivalent.

WRITTEN BY
WIL MANCUSO &

BOB JONES

But, there are important considerations that can significantly improve implementation speed, ROI, and costs associated with creating a more dynamic, standards-based infrastructure. This article explores what it takes for organizations in the public sector to build an effective enterprise architecture and practical recommendations for a successful, and cost-effective implementation.

To meet today's demands and plan for those of tomorrow, IT organizations in both the commercial and public sectors must establish an architecture that is both flexible and standards-based. The ability to proactively manage the enterprise computing architecture is the chief goal of information technology departments within any government organization. In fact, an organization's ultimate success may hinge on its ability to rapidly respond to changing business strategies, regulations, and technologies.

Today, that translates into an architecture that provides the means to automate processes between existing legacy applications and newer enterprise applications, and one that can support the exchange of data between multiple applications in real, or close to real, time. It also means an architecture that provides the basis to easily add or plug in new applications, that supports widely accepted industry standards, and that provides the capability to monitor and measure performance.

The challenge, of course, for government organizations and Fortune 500 companies alike, is in getting there. It is a significant undertaking, fraught with potential roadblocks, but there are products and technologies available today to help organizations overcome these obstacles and achieve the transformation. And the payoffs are worth it.

With a modern, performance-based enterprise architecture, organizations will achieve significant outcomes from their investments. The first is an enterprise with a reengineered set of business processes that can be monitored, measured, and centrally tracked in real time, achieved using best practices like the Supply Chain Operational Reference model (SCOR). The second is a framework that transcends technology, allowing for data standards as they evolve. The third is an aggressive portfolio management plan that integrates legacy systems with modernization efforts, and in the process eliminates unnecessary and redundant systems. Finally, organizations will be able to put into place a set of metrics to measure, track, refine, and ultimately ensure the success of the enterprise.

Making the Leap To Service-Oriented Architecture

The first step in achieving a modern architecture involves creating a strategic vision – determining exactly where the enterprise needs to be in the long term. The vision should only consider the "Future Enterprise," and not be influenced or limited by the "As Is Enterprise." A strategic vision will serve as the basis for the road map that will chart and transform the current architecture to the desired state.

For many organizations, that strategic vision will mean deploying a service-ori-

ented architecture (SOA). At its core, an SOA consists of a collection of services on a network that communicate with one another. Rather than serve one application or function, those services perform a business process, which can be utilized by multiple applications. For example, when a credit card authorization application is developed as a service rather than built directly into a program, it can be used or consumed by multiple applications. This highlights obvious advantages for IT departments since applications can be reused and are not hard-coded.

The benefits of migrating to an SOA are many. The most significant, particularly with large government organizations that have grown up over the years with stovepipe applications, is the ability to economically integrate systems. Legacy systems need not be ripped out and replaced; they can be linked and modernized using a wide range of standard protocols, such as Simple Object Access Protocol (SOAP) and Web Services Description Language (WSDL). Existing business processes can then be leveraged as services by multiple applications – using a service only requires knowing its interface and name.

With an SOA, infrastructure development and deployment become more consistent across the enterprise. Newly developed components and applications purchased from vendors can be integrated easily and in turn deployed as services. New initiatives are able to leverage existing services and components kept in Web services libraries, greatly reducing the time-to-market and cost of development. Reusing tried and tested components also reduces the risk and maintenance factors involved in introducing new applications into the environment.

Today's SOA-based products make the once overwhelming task of integrating enterprise systems now seem rather tame. A key consideration for selecting an SOA architecture is a flexible deployment environment – due to the diverse, heterogeneous nature of most IT organizations, it is highly advantageous to have the option of peer-to-peer deployment of SOA solutions.

Defining the Future – Beyond the "As-Is" State

The next step involves choosing the architectural framework. This is a critical phase; the goal is to achieve application and data interoperability across the enterprise that extends interoperability to include partners and suppliers through common data standards and processes. The Zackman Framework is the forefather of many of the current models of the architectural framework now deployed in the federal government and the Department of Defense (DoD). Within the DoD, the models in use are the DoDAF and an extended version called the DoDAF Plus. The DoDAF consists of an operational architecture, (OA), system architecture (SA), and technical architecture (TA). The DoDAF Plus extends the model to include a data architecture (DA), which focuses on ensuring data interoperability (see Figure 1).

Once the architectural framework has been selected to create the enterprise architecture, the next step is to capture the business processes currently in place so that they can be mapped and transformed to the strategic vision. This is the area where many enterprises fall short. If the right tools are not chosen to map and transform the processes to the desired enterprise architecture, the organization loses the benefits of creating the enterprise architecture in the first place and cannot achieve the expected return on investment.

> "With a modern, performance-based enterprise architecture, organizations will achieve significant outcomes from their investments"

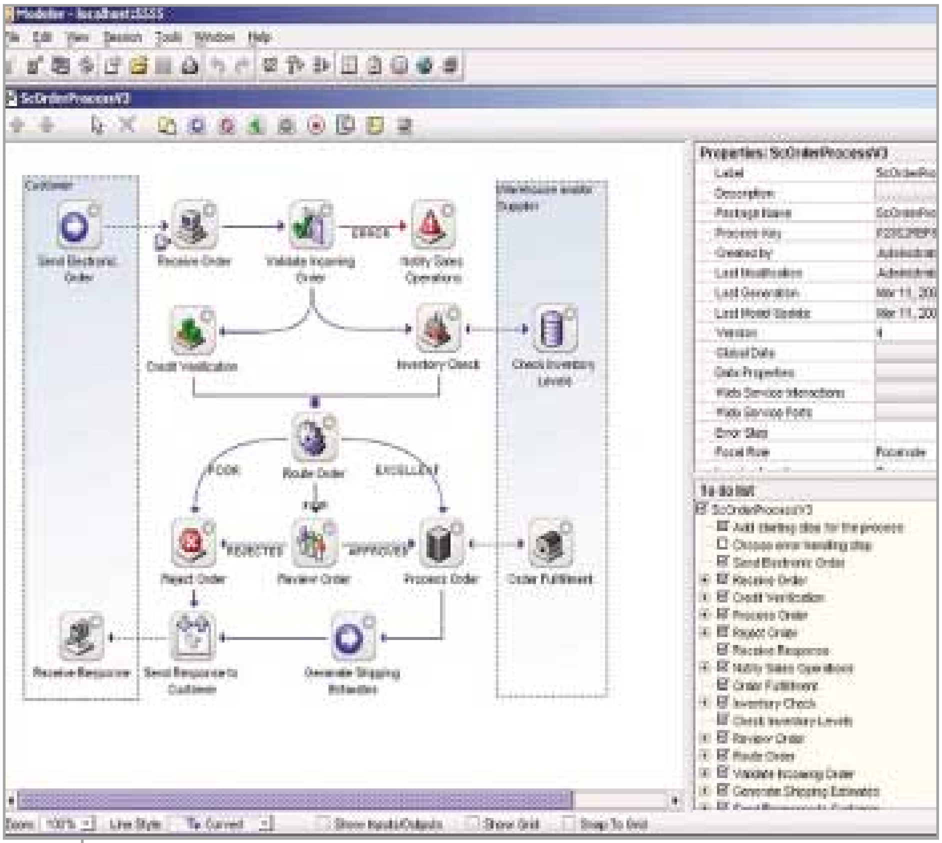


FIGURE 1 End-to-end graphical business modeling with webMethods BPM product

The work begins by capturing the existing or "As Is" business processes. The selected solution should provide more than just the graphical representation of the flow of information among systems, organizations, and people; it should have the ability to execute the underlying code and deploy the business process within the same tool. It is essential that it include the following features:

- ***Ease of use:*** Nontechnical process owners, such as business users and operations staff, must be able to easily define and understand process flows.
- ***Complete process integration:*** Must be able to include automated tasks that span intra- and extra-enterprise operations, Web services, as well as steps performed by people.
- ***Agility:*** Offer the ability to dynamically modify and refine business processes in response to changing business requirements.

A primary goal of business process automation is, of course, to reduce the level of human involvement in a process. However, many processes require exceptions handling at some level. Those human workflow components need to be captured and incorporated into the overall business processes. There are solutions available today that include human workflow capabilities and ensure that processes can be graphically mapped, represented, and executed.

With a clear end-to-end picture of the system and human-related components involved in business processes, organizations can then easily and dynamically look for ways to drive operational efficiencies. Information can be tracked across people and departments, with an eye towards standardizing work methods and audit trails. In effect, process owners can point-and-click their way towards better, more cost-efficient operations and more easily duplicate best practices.

Monitor and Measure

It's no longer enough to cobble together several systems or let one or two applications reach into another's data. Today's highest value initiatives require end-to-end integration of complex enterprise processes, coupled with technologies that enable real-time insights into what's really happening in those processes.

To achieve true effectiveness in optimizing your operations, you need to be able to monitor and manage processes as they are being executed. Business Activity Monitoring (BAM) allows organizations to define key performance indicators (KPIs) that are critical to the enterprise, along with the key metrics associated with each KPI. In conjunction with this, BAM allows you to monitor transactions in real time and to compare and analyze transactions against KPIs. These systems also have built-in intelligence to learn from past events and to predict and recommend solutions to future problems.

The Big Picture

There is a growing movement among government organizations to develop enterprise architectures that not only serve the needs of today, but are also positioned to respond to the rapidly changing needs of tomorrow. The thought of getting there seems overwhelming and in the past has been an extremely expensive and difficult undertaking. However, with SOA-based integration products, almost every aspect of the modernization process has been addressed by technologies and services that make the task much more realistic and economical.

> "With an SOA, infrastructure development and deployment become more consistent across the enterprise"

- Business process management solutions simplify the process of mapping the "As Is" enterprise architecture against the desired or "To Be" architecture. Processes can be graphically depicted, making it easier for business owners to streamline operations and find efficiencies.
- Sophisticated enterprise integration platforms facilitate the implementation of service-oriented architectures, providing the means to integrate legacy systems with newer applications, and to implement businesses processes as services, which may be accessed and utilized by multiple applications.
- Off-the-shelf adapters and brokers, combined with widely accepted standards, simplify the process of exchanging data between disparate systems.
- Business activity monitoring solutions, which include balanced scorecards and dashboards, complete the picture, providing managers with the means to monitor and measure processes, and to ultimately predict and resolve problems before they have a chance to negatively impact the bottom line.

By taking a holistic approach towards achieving a modernized enterprise architecture and utilizing the technologies and services available today, government organizations and businesses alike can speed the time to achieving tangible results. They will be able to extend data interoperability across the enterprise, eliminate redundant systems, improve data quality, and measure processes against business-oriented goals.

Most importantly, a foundation will be in place to automate and incorporate a wide range of processes to accommodate the needs of today and tomorrow.

About the Authors

Wil Mancuso is consulting as the chief enterprise architect for the Department of Defense for Logistics and is a leading authority for enterprise architecture and enterprise data Integration. Wil is the author of the DoD's Business Enterprise Architecture LOG and the Enterprise Integrated Data Environment.

■■■ William@Mancuso.us

Bob Jones is the director of the government business unit at webMethods, Inc.

■■■ Bjones@webMethods.com

Sleeping with the Enemy

.NET and Java living together

People often assume that .NET and J2EE are locked in some kind of life and death struggle. In fact, they can co-exist very well as I proved on a recent project. Both .NET and J2EE are based on open standards; these are Web Services Definition Language (WSDL) and Simple Object Access Protocol (SOAP), both of which are built on Extensible Markup Language (XML). This article describes how a .NET Web Services application was cloned using Java to run on a variety of Unix platforms.

WRITTEN BY
FRED DOWN

History

Nova Marketing Group has developed a set of shared libraries and an address database that allows their clients to include powerful address validation, correction, and transformation functionality within their business applications. The product is called nCode. Banks, call centers, and government departments are typical users of nCode. It uses a unique, extremely high performance engine written in C++ with a lot of attention to portability. The same source code runs on a wide range of hardware, from IBM AS/400 and Z Series mainframes, to low end Windows machines. In between, it runs on practically every flavor of UNIX. This portability is achieved with no compromise to performance. Over time, various enhancements including Java Native Interface and DCOM client/server wrappers have been developed. My initial connection with Nova Marketing Group was as the maintainer of the Unix ports.

Nova Marketing Group decided to produce a Web services version of nCode. The first implementation of nCode Web Services Edition was built on the Windows .NET architecture. My project consisted of building a version that would run on UNIX, specifically AIX and WebSphere.

Free Tools: Not Just Because I'm Cheap

I can't deny that cost was a factor in my choice of tools. It was not, however, the main factor. Since Nova develops on eight flavors of UNIX as well as OpenVMS, it is very helpful to have the same set of tools. For this reason development was achieved using g++ and Xemacs using CVS as the version control system. Using the same compiler on each platform enhances portability. I only have to deal with the quirks of one compiler. Make files are substantially the same for each platform.

When building open source tools, things go better if one builds them with gcc. They also seem to work better on open source platforms. This will become very relevant later.

Implementation

Web services work by implementing remote procedure calls using SOAP as an architecture independent transport mechanism. Mechanisms exist that allow the code that implements the procedure calls to create a WSDL file that defines the interfaces. Mechanisms also exist to automatically produce code that implements Web services interfaces given a WSDL file.

At the start of the nCode Web Services Edition project we developed a proof of concept based on gSoap, gSoap is a C++-based set of utilities for writing SOAP clients and servers. It has a utility that generates C++ code for clients and servers given a WSDL file. This has the advantages of producing low footprint, high performance clients and servers. These SOAP application servers work with Apache 1.x Web servers. In many ways, this seemed to be an excellent product, and for many applications it is highly recommended.

Unfortunately for us, this approach had two fatal drawbacks:

- The SOAP produced was not compatible with the SOAP expected by Microsoft's SOAP handler. The naming conventions did not quite match.
- gSoap is tied to Apache. This was no longer acceptable since our potential customers needed to run nCode Web services on the application server of their choice.

Having abandoned gSoap as the solution, I looked at various other approaches. Java was initially ruled out for performance reasons. The four that got some attention were:

- Mono
- PERSIA
- Rolling our own SOAP
- Unix-based DCOM

Mono is an effort to create an open source implementation of the .NET Development Framework. Mono is produced by Ximian, which is now part of Novell. Mono has the potential to allow considerable interchange between the Microsoft and open source platforms. Microsoft has been somewhat ambivalent towards this effort. Mono was not, in my opinion, mature enough at the time I was doing the evaluation (November 2002). One problem I found was the lack of documentation. Someone needs to write "Mono for Dummies". Bear in mind that the platform I was targeting was AIX. Functioning AIX

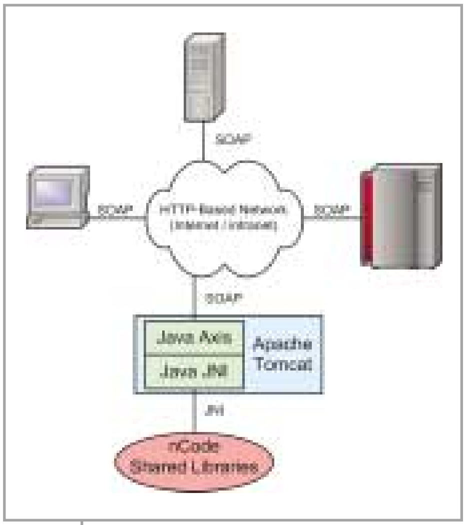


FIGURE 1 Unix-based proof-of-concept architecture

versions of open source projects tend to become stable sometime after the equivalent Linux and BSD releases.

PERSIA, which is now called LIEF3, is a proprietary Web services product from Roguewave. This looked promising, as at the time we were very into doing the project in C++. However an AIX version was not available, and there were requirements concerning the C++ compiler. The folks at Roguewave were extremely helpful with technical information.

Rolling our own SOAP was considered, but rejected after my experiences trying to get gSoap to work with IIS SOAP. This is seriously complex code that would overstretch us in both implementation and testing.

Unix-based DCOM was explored as part of a solution. It seemed to have the option of allowing us to reuse the DCOM-based nCode server that had been produced for Windows. This ultimately turned out not to be true. The Windows-based nCode DCOM server made heavy use of Windows threads, which are not portable to a POSIX based environment. Another problem was that the data transformations from DCOM format to nCode format were coded in Visual Basic. DCOM stores variable length arrays in a particular format called SAFEARRAY. While easy to access from .NET languages like C#.NET and VB.NET, they are not at all easy when using MS Visual C++. It would be even harder when using a generic C++ compiler such as g++.

The Java Revelation

The decision to use Java was made reluctantly because I was blinded by performance concerns. In fact these concerns were misplaced, since Java offers considerable advantages which more than compensate for any loss in performance.

Java is a managed environment. It is much easier to write crash-proof programs using Java than using other languages like C or C++. Writing C++ that does not contain memory leaks or memory access errors is much easier said than done. Server software must run for months without restarting, so even small memory leaks are problematic.

J2EE provides a standard cross-platform and cross-vendor environment. “Write once, run anywhere” is considerably more than just a slogan.

Java provides a standard way to process WSDL to implement Web services. It is also very well documented. Using the tool wscompile and a small amount of Swing Java GUI code, it was quite easy to produce a Java-based client that could access a subset of the .NET-based IIS server’s Web services. The utility wscompile or an equivalent is contained in any Java Web services toolkit. This utility takes as input a WSDL file and produces the classes required to create either a SOAP client or a SOAP server. The WSDL file I used was produced by the nCode Web Services Edition .NET development team. In this case I used wscompile to produce the classes for a client.

Producing the client was a significant milestone since it verified that .NET SOAP and Java SOAP were actually compatible. This was not the case with the prototype built using gSoap. The data structures used were not trivial, variable length arrays of nested structures were amongst some of the challenges presented.

Next I had to produce a proof-of-concept server. The WSDL file was processed once again with the wscompile utility to create the Java classes needed for the server. The server’s Java code accessed the nCode libraries by JNI. JNI is Java’s mechanism for invoking routines in external libraries. Nova Marketing Group already had the necessary JNI wrapper classes.

The proof-of-concept server was produced as a servlet running on the Tomcat application server platform. According to Sun:

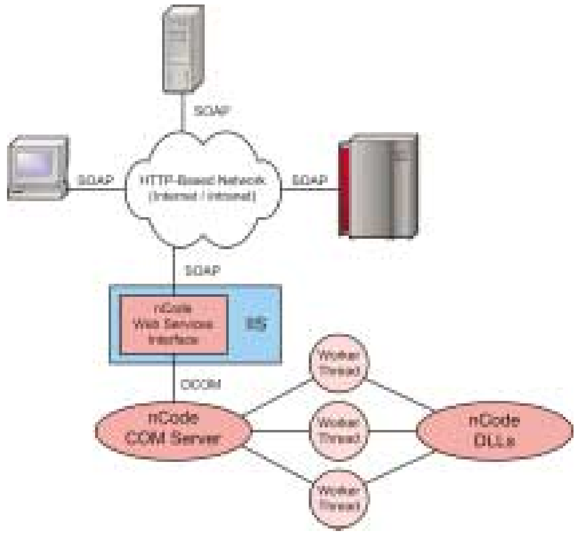


FIGURE 2 **Windows implementation of nCode Web Services**

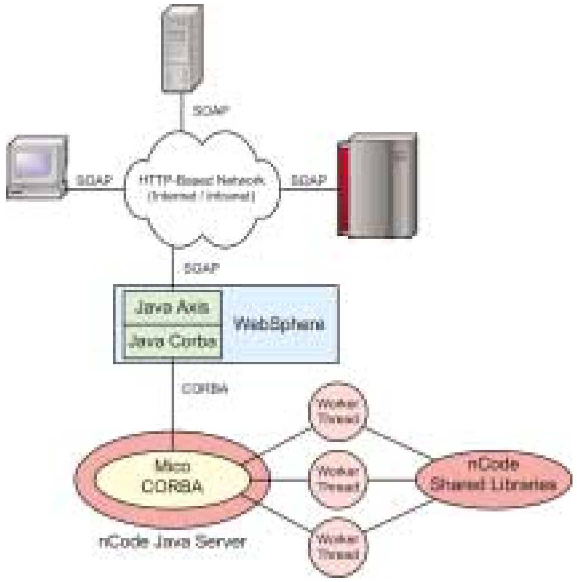


FIGURE 3 **Unix implementation of nCode Web Services**

"Java Servlet technology provides Web developers with a simple, consistent mechanism for extending the functionality of a Web server and for accessing existing business systems. A servlet can almost be thought of as an applet that runs on the server side – without a face. Java servlets make many Web applications possible."

Apache Axis was selected to handle the SOAP on the application server end. According to the Axis Web site: "Axis is essentially a SOAP engine – a framework for constructing SOAP processors such as clients, servers, gateways, etc. The current version of Axis is written in Java."

Axis provided the tools to convert the WSDL produced by the .NET implementation into a set of Java classes that, in turn, implemented an applet that could be run on any J2EE-compliant application server. Axis also provided stub routines for implementing the nCode interface logic. Tomcat was chosen as the application server since it is free, small, and cross-platform. It is also well documented and easy to configure. This allowed the proof of concept to be demonstrated with Tomcat running on a Pentium 200 running Linux. Figure 1 illustrates the UNIX-based proof-of-concept server architecture.

I proved to myself that Java SOAP and .NET SOAP were interoperable. The proof of concept used moderately complex data such as nested structures and variable length arrays and could exchange data in both directions with its .NET counterpart.

Next I had to design the real thing. Various approaches were considered. For a while I favored scaling up the proof of concept and calling nCode directly using JNI. I was persuaded from this by Nova's technical team who had produced a separate nCode server process to handle nCode transactions.

The Windows approach is shown in Figure 2. Requests are processed by IIS, the Windows Web/application server. These are then passed by DCOM to an nCode transaction server, which passes the request to an available worker thread or queues the request until a worker thread becomes available. The number of worker threads is defined at start time by a configuration file. Each worker thread contains its own connection to the nCode API, which resides in a set of Windows DLLs. These connections are created only once, at start-up time. Connecting to the API is a comparatively expensive operation, so connecting only once saves resources. If no transactions are received for a certain period (about 20 minutes) the transaction server closes down. This flushes any data requests as well as freeing resources.

A multithreaded approach was mandated for better scalability. Since some nCode transactions could be too time consuming for high transaction volume systems, it would be impractical for all other Web requests to be stalled pending the completion of a complex query.

Why a JNI Approach Would Have Been a Bad Idea

Before looking at how I finally architected my solution, it is worth explaining why JNI would have been a bad idea. JNI (Java Native Interface) is a mechanism that allows Java to call functions written in another language. This gives two main advantages:

- The code will run faster than if it was written in native Java.
- The code can do things that are impossible in Java, such as interfacing directly with the operating system.

This sounds wonderful. I can call nCode directly and run at C++ speeds. So what's the catch? Java and C# are called *managed environments*. Code is only allowed to do valid things. Bounds violations and memory allocation problems are handled by the virtual machine and trapped. This, however, does not extend to code called by the JNI mechanism. Errors here can and do crash the Java Virtual Machine. When running in an application server as one thread in the server's Java Virtual Machine, a memory error can bring down all of the services on the application server.

Evolution of the Solution

The Axis servlet approach worked well in the proof of concept and was used in the end product. The decision to have a separate nCode server process required some interprocess communication.

Various approaches were considered. CORBA was initially ruled out since I had bad experiences with it on an earlier project. However, it eventually became apparent that creating a language-neutral, architecture-neutral, bug-free interprocess communication package from scratch was a tall order. CORBA suddenly became more attractive. What made it more attractive was that Java supported CORBA as an inter-process communication protocol. The CORBA IDL (Interface Definition Language), which defines the interfaces, was obtained by converting the DCOM IDL used on the Windows implementation. There was a need to provide a Web services to nCode Server layer. The nCode Server was a port of the existing Windows nCode Server, which implemented an interface that was much closer to the API provided by the nCode shared libraries. The Web services API was much higher level. This mapping was done on the Windows version using Visual Basic, which had to be hand translated in Java for my implementation.

Building the Unix nCode Server – First Find Your ORB

The UNIX nCode Server could only be loosely based on the Windows implementation since the Windows version used both DCOM and the Windows thread model. The first thing I required was an ORB – a library and set of utilities that allow one CORBA process to talk to another. My ORB had to be:

- Compatible with the CORBA stream emitted by my Java applet
- Cross platform; available for at least Solaris, AIX, and Linux
- Reasonably inexpensive; the cost of using the ORB should not impact the price of the nCode product

I tried several ORBs with varied levels of success. Then I encountered Mico, which worked, was well documented, and ran on virtually every platform.

Using the IDL-to-C++ generator, I was able to make a skeleton server. All I had to implement was the functionality that linked the IDL methods. Simple really; well, not quite. The first problem: How does the Java code find the C++ server? There are various ways a server based on CORBA can advertise its presence. I chose the simplest. When the server starts, it writes its ior (how to communicate) data to a file in/tmp. The Java side just reads this and links to the server accordingly.

Since Mico is multithreaded, there was no need to implement threading manually. It came for free with Mico. Mico threads are transitory. The nCode connections were implemented in a pool, since the setting up of connections to nCode is a comparatively expensive operation. When a thread starts, it looks for an existing free connection in the pool. If there are no free connections it creates a new one and adds it to the pool. If there is no server activity for twenty minutes all connection are closed to flush any data. The implemented solution is depicted in Figure 3.

Implementation Surprises

After the usual round of debugging, fixing typos, and other errors, it all seemed to work. Unfortunately, I wasn't out of the woods yet.

Software Layer	Implemented As	Implementation Language	Compiler	Comments
nCode API	Set of DLLs	C++	MS VC++	The implementation of the nCode libraries is C++-based for performance and portability reasons.
nCode COM Server	COM EXE	C++	MS VC++	A natural way to implement a custom COM EXE server that interacts with native C++ libraries is C++. Separating COM process from IIS means that fault tolerance is built in.
nCode Web Services Interface	Web Service	VB.NET	MS VS.NETCOM	Interop provided by .NET makes it very easy to call the COM EXE. At the same time, VB.NET makes working with SOAP and WSDL as transparent to the developer as possible.

TABLE 1 **Windows Implementation Summary**

Software Layer	Implemented As	Implementation Language	Compiler	Comments
nCode API	Shared libraries	C++	xLC for AIX Usually g++	The implementation of the nCode libraries is C++-based for per formance and portability reasons.
nCode Java Server	Daemon process stated at boot up	C++	g++	CORBA replaces the COM in the Windows version. Keeping the server process separate from the application server means that fault tolerance is built in. The IDL compiler provided by Mico converts IDL to C++ classes.
nCode Java WAR	Web Service	Java	Usually Sun's JDK On AIX IBM's Java 1.3	The Java Axis package from Apache makes the WSDL and SOAP transparent. The CORBA package provided with Java handles communication with the nCode Java Server. RMIC in Java converts the IDL into Java classes.

TABLE 2 **UNIX Implementation Summary**

The AIX nightmare

After a reasonable amount of testing on Linux, I decided to move on to AIX.

The AIX version of the nCode libraries is implemented using xLC, the IBM C++ compiler, since that is what Nova's customers use. Mixing different C++ compiled objects requires considerable patience, luck, and in-depth knowledge. Mico wouldn't build using xLC on my hardware. I eventually resolved this problem by building the Mico library using GNU g++ and using my own patience, luck, and in-depth knowledge. After combining the GNU g++ compiled Mico library with the xLC-generated nCode shared libraries, things went more smoothly.

Other Bad Dreams

Simple demos ran fine but under load the whole system started to slow down and fail. The problem was that Mico fires off a new thread for each transaction; these threads would not die until told to by the Java end terminating the transaction. This was done as part of the garbage collection routine of the Java Virtual Machine. When the Java end is busy, garbage collection is deferred and threads accumulate. I resolved this by directly invoking the garbage collector in the Java code.

The next problem was a memory leak in Mico when CORBA used output-only parameters in our interfaces. This went away when the parameters were made input/output.

Good Dreams

Testing was the good news. I was able to use the testing suite that Nova had developed when implementing the Windows implementation of nCode Web Services Edition. This included both a set of functional tests and a stress testing harness.

Functional testing was performed using Web pages hosted on an IIS Web server that made Web services requests to another application server to perform various nCode functions. The server URL for the application server requests was defined in a parameter file. Changing the URL value allowed me to test the Unix implementation of nCode Web Services Edition. Stress testing was performed by a suite of VB.NET utilities that made Web services requests to a server defined in the configuration files. This suite, which was highly multithreaded, was able to simulate a wide range of scenarios by defining complex test scripts.

Testing was performed initially using Tomcat, and finally using WebSphere. The implementation in WebSphere was simple, a matter of creating a WAR file containing all the Java class files and the JAR files to run them. A WAR file is the same as a Java JAR file but the suffix is changed to WAR. WebSphere's configuration utility just read the WAR file and performed all the necessary setup.

Comparison Between Windows nd Unix Implementations

Tables 1 and 2 summarize the two implementations.

Acknowledgements

I would very much like to thank my client Nova Marketing Group (www.novamg.com) for their patience and technical help. Special thanks go to Josip Mihaljevic, who patiently explained how the Windows implementation worked and reviewed my designs. Special thanks also go to Zelko Odorcic, who managed the project and also reviewed my designs.

Resources

- *gSoap:* www.cs.fsu.edu/~engelen/soap.html
- *Mono:* www.go-mono.org
- *PERSIA / LIEF:* www.roguewave.com/products/leif
- *Java service toolkits:* http://java.sun.com/webservices/webservicespack.html or www.alphaworks.ibm.com/tech/webservicestoolkit
- *For more information on Tomcat*: http://jakarta.apache.org/tomcat.
- *Apache Axis:* http://ws.apache.org
- *Mico:* www.mico.org.

About the Author

Fred Down is an independent consultant specializing in Java and cross-platform development. He wrote his first program in 1969, and has used Unix since 1983 and Linux since 1993. He has worked for IBM, NCR, and Citibank at various times on a wide range of projects. Fred loves his family, his cats, his old motorcycles, and his old computers in that order.

■■■ fdown@fdown.com

Application Server Architecture and BPEL

Promises and challenges

In recent years the application server has greatly evolved, expanding the set of core services provided by the infrastructure. The current Java platform supports XML data handling, scalability, load balancing, and other capabilities that allow application-level services to be developed more easily and deployed more reliably. This progression must now address developers' latest concerns regarding security, distributed transactions, and reliable messaging because applications no longer stand alone – they're deployed into a technology ecosystem that can span departmental and organizational boundaries.

WRITTEN BY
AMLAN DEBNATH

In this environment, a well-behaved application not only needs to interact with external systems and consume services from them, but also needs to be a service provider. This is driven by a need for reuse and adaptability and fuels the current push toward services-oriented architectures (SOA).

However, this leads to the question: How do we get all these services to work together in a heterogeneous, networked environment? The answer is BPEL, the Business Process Execution Language for Web Services. BPEL provides a standard, portable language for orchestrating services into end-to-end business processes and builds upon a decade of progress in the areas of business process management, workflow, and integration technologies. It's built from the ground up around XML and Web services and is supported on both the Microsoft .NET and Java platforms.

What does BPEL add to the existing Web services standards and Java platform? It's clear that the industry must have been hungry for BPEL, given the support it has received from nearly every major technology vendor in the past year – but why? The first driving force is the new class of connected applications, which makes implementing business processes a mainstream problem that most developers must tackle. Surely a second factor was the alphabet soup of earlier proprietary workflow languages, which slowed their adoption and created a standards vacuum that BPEL fit perfectly. And finally, Web services have accelerated the process by providing a standard interface for publishing services and requiring a shift in the way service composition is done. BPEL, then, is just what the doctor ordered. The emergence of BPEL as a standard for describing business processes is a step in the evolution of the application platform (see Figure 1).

Aligning with this shift, many of the major technology vendors in both the Java and .NET camps have announced that they will ship BPEL engines in the future, so why are so few commercially available today? It turns out that the maturity of BPEL as a process language makes it feature rich but complicates the process of building a scalable, reliable BPEL engine. For example, BPEL is designed with asynchronous services at its core, but this means that servers must deal effectively with persisting state for long-running flows, correlating asynchronous messages, and reliably handling the case where an outbound message has been sent but the server crashes before the response is received. The rest of this article examines how these requirements, and the new standards to support them, naturally extend both the Web services standards and the current Java platform.

To make this discussion more interesting – and more comprehensible – let's consider a real-world example: an order management process at a large hardware manufacturer. This manufacturer accepts wholesale orders from many different sources and responds immediately with an order tracking number, but has a long-running flow in the back end to process and track the order and call the client back when an invoice is ready.

As shown in Figure 2, this flow needs to invoke synchronous services, such as looking up payment terms in an Oracle Financials package, as

Assande EAI Messaging Suite features

- XML Schema Repository
- Data Abstraction, Transformation, and Mapping
- Message Assembly and Generation from Components
- Messaging Integration Service Deployment
- Automated Naming Standards, Version Control, and DIFF Functions
- Infrastructure Utilization and Pipeline Reporting
- Conversation Deployment and Message Release Management
- State Management and Integrated Workflow
- Advanced Search Capabilities
- Integration Documentation (MS Word, Excel) Repository

ASSANDÉ™
INTEGRATION MANAGEMENT COMPANY

http://www.assande.com

well as asynchronous services, such as submitting the order to a mainframe system, which will compute the invoice as part of a batch process. XML data is exchanged between all the systems and the manufacturer must process millions of these transactions a day at peak loads—tracking them, reporting on them, and handling exceptions, notifications, and manual processing steps as needed.

This process is a typical example of the new class of requirements that developers must address when developing SOA-based applications, including:

- Bindings to heterogeneous back-end systems
- Asynchronous interactions
- XML data transformation
- Flow coordination

These requirements are transforming the application server from a container for presentation and tightly coupled business logic to an infrastructure that equally supports asynchronous messages and flow coordination. This transformation is enabled and accelerated by emerging standards that will extend the boundary of the Java platform as we know it (see Figure 3).

Some of the key standards that are being implemented in J2EE application servers around this area include the following.

Extensible WSDL Binding Framework (JSR-208)

Web services are clearly demonstrating their value as an integration standard; however, not all back-end systems are SOAP or Web service enabled. The JSR-208 working group and existing frameworks like WSIF (Web Services Invocation Framework) from Apache focused on helping the Java platform support Web services messaging without requiring every system to be wrapped with a Web service. In this way, hardware manufacturers can use BPEL to orchestrate JMS messages sent to and received from the mainframe.

Process Flow Coordination (BPEL)

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. While the order management flow looks simple, the long-running and asynchronous message exchanges alone would make it complex to implement in Java today. Add parallelism and compensation logic and things get downright ugly….

Reliable Web Messaging (WS-Reliability)

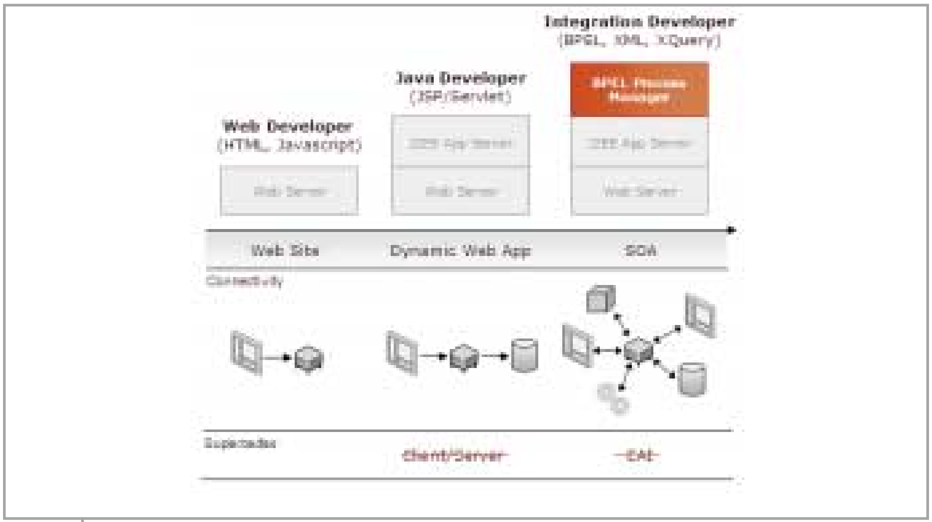


FIGURE 1 **Evolution of the application platform**

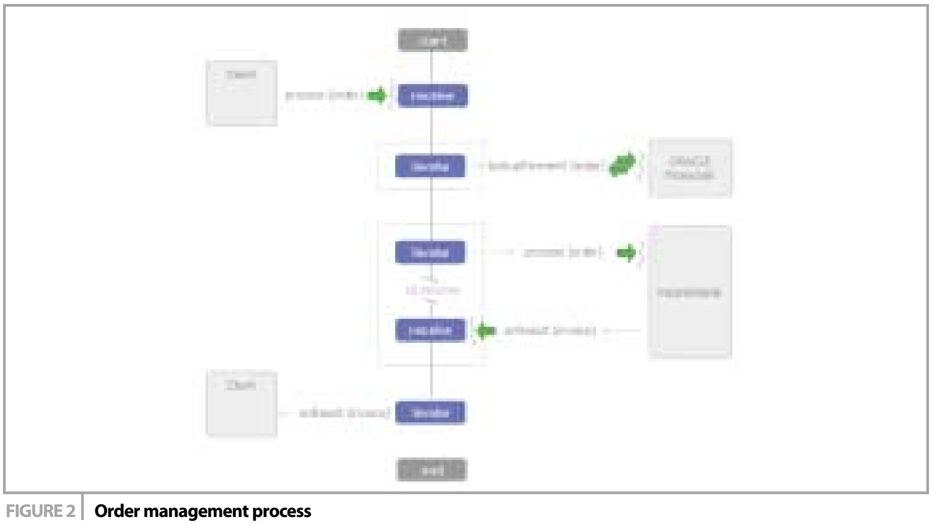


FIGURE 2 **Order management process**

One of the challenges of SOA-based applications is that you can't assume that all the endpoints are available at the same time, all the time. The WS-Reliability Web messaging standard lets the infrastructure guarantee the order and delivery of messages across service endpoints.

Security (WS-Security)

In addition, security requirements are obvious when exchanging text-based messages over (and across) unsecure networks. By addressing this problem with infrastructure-level standards, security is provided without sacrificing interoperability.

XML Data and Transformation (JAXB, XQuery)

SOA-based applications need to access and manipulate XML documents flowing into and out of each service. New Java facilities like JAXB and languages like XQuery simplify these tasks.

User Interactivity (WSRP)

Most business processes incorporate user interactions at many levels, such as portals to initiate and inspect the state of processes, manual approval tasks, and exception handling. The Web Services for Remote Portlets (WSRP) standard enables the next generation of application servers to support user interactions in composite processes as robustly as they're supported for Web applications today.

Choreography and Contracts (WS-CDL)

As the Web services standards reduce the barrier for trading partners to interact, a formalism is required to describe the contracts involved in richer business collaborations.

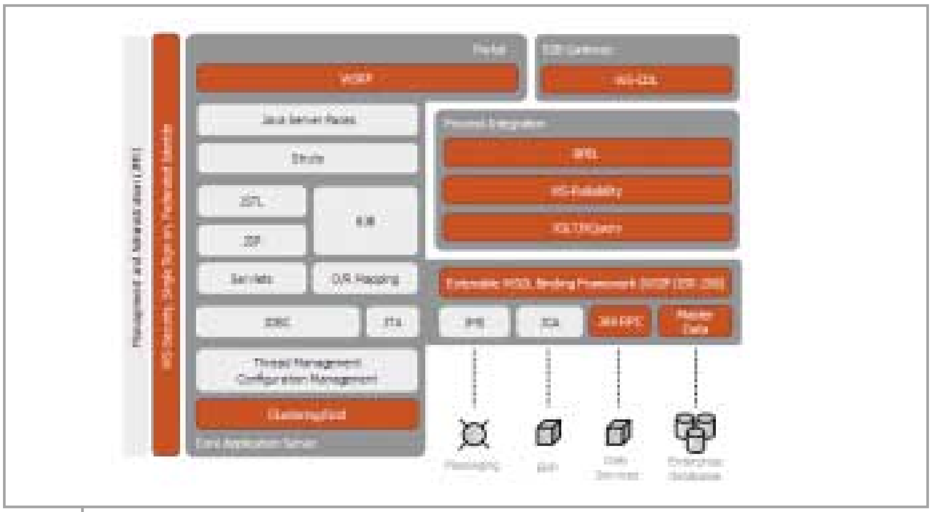


FIGURE 3 | **The evolving application server platform**

Summary

The trend towards building SOA-based applications marks a fundamental shift in the way that applications are built. Applications today are triggered by events, orchestrating services from both existing and new applications, and integration must be asynchronous and loosely coupled to be reliable. Not coincidentally, a new set of standards has emerged to address these requirements, and vendor adoption of these open standards is increasing confidence and accelerating adoption in the IT community. This is clearly promising and will offer enterprises more seamless interoperability between heterogeneous systems and services than was previously possible. Of course, all the applications deployed on this standards-based infrastructure benefit from the inherent capabilities of the underlying platform.Of course, some challenges remain. Several standards in areas such as reliability and connectivity are less mature. Also, because this new architecture doesn't fully address information quality, vendors must provide data-quality services that offer profiling and cleansing features. Still, it's clear that these developments are going to dramatically change how we build applications – and that the application server and BPEL are at the core of this new wave.

About the Author

Oracle Integration Guru and vice president of Server Technology, Amlan Debnath joined Oracle from TIBCO Software Inc., where he was vice president of Integration Products. At Oracle Amlan is moving the integration products forward to a standards-based offering.

■■■ amlan.debnath@oracle.com

IT solutions GUIDE
TECHNOLOGIES YOU NEED NOW!
How to Manage Your Ideas Using Today's i-Technologies
Top IT Solution Providers!

Cleansing Data at the Source

Relieve stress with real-time data validation

■ Bandwidth.com provides options for securing the best deals on Internet connectivity. It has sold Internet connectivity of all types and speeds from a wide selection of national service providers to thousands of homes and businesses since 1999. The company's quote service provides up-to-date prices from top providers nationwide.

WRITTEN BY
SCOTT BARSTOW

It was logical that the company would also look to the Internet for the best way to boost the performance of its growing customer service operation. Bandwidth built an application to verify customer address data before a quote is provided. The result: accurate quotes provided in seconds for customers.

The Problem with Customer-Input Data

After introducing an Internet-based service procurement application in January 2003, we soon realized we needed a swifter and more efficient way to cleanse and validate the information that customers were entering in our quote request form. Customers that were entering inaccurate or invalid name or address data in the form were placing unnecessary stress on our real-time pricing system, which resulted in rising customer frustration and increased wait times.

While the problem may have seemed complex, the solution was simple: find a full-featured address validation system that could operate in real time. There was one additional requirement. We did not want to maintain any of the validation data on our own servers. We wanted a Web-based solution so we could truly leverage the power of the Internet to check the customer information in real time without the burden and expense associated with having to create and maintain our own corporate database. We sought these capabilities in a reliable, platform-independent package that could be implemented and operated quickly. Melissa Data Corp.'s Data Quality Web Service (DQWS) delivered that functionality.

A Speedy Solution

After securing a trial account, we were writing sample applications in one day. Melissa Data maintains the U.S. Postal Service databases at its facility, and takes care of all the servicing and updating functions. This made their Web-based service a perfect fit for all our needs.

The data validation service provides instant verification, correction, and standardization on name, address, nine-digit ZIP code, phone number, and "geo-coder" (latitude and longitude) information.

With DQWS, Bandwidth.com passes along all the standard address information received from customers to the application via an HTTP post. XML technology and the Internet provide easy and secure online data verification. In return, DQWS sends a message, including the cleansed address and any errors that occurred during validation. The cleansed information is then sent to Bandwidth.com's pricing engine to determine the Internet service availability options for the given address.

If a bad or incomplete address is given, the customer is alerted instantly and asked to resubmit the form.

For example, if a customer were to input an incorrect address for the phone number given, the validation engine would issue an alert. Sometimes people do that because they want someone to call them back instead of completing the transaction over the Internet. Web validation improves our ability to be flexible in order to successfully complete every transaction. And by doing it more quickly and efficiently, we're able to assist more customers than we ever could before.

The Web validation service has performed so well that we recently began using it to perform street searches for addresses where the range submitted by the customer was rendered invalid. This function has proved invaluable to our partners and us because it provides alternative addresses when there's no exact match.

Only five times in the entire first year of operation did a customer submit a valid street address that the Web service could not validate. This is especially impressive since Bandwidth .com validated between two and three million records using the data Web service last year. Not only is the service incredibly accurate, but our typical response time remains at one second or less. This is extremely important for the real-time nature of our application.

Lessons Learned

Maintaining a speedy response time is of the essence because some of our biggest partners are some of the biggest names in personal computing, including Dell Inc., Hewlett Packard Co., and IBM Corp.

The Dell Web site alone delivers peak loads of 1,000 to 2,000 requests an hour from customers wishing to secure high-speed Internet access to use with their new high-tech gear. Because of the exemplary response time now possible with address validation, Dell customers have the opportunity to be informed of the variety of Internet connectivity options in their area and select one – all before the end of the call.

Summary

Address validation upfront is a must for a high-quality, real-time procurement process, as service providers' regions of coverage vary according to customer addresses. To sell our services online with confidence, we need the assurance of address accuracy. Before implementing address validation, customers going to our Web site would fill out and submit a lead form, then get in the queue and wait for a callback. It wasn't until that time that any information discrepancies were mentioned and then they had to be cleared up over the phone, which always took precious time.

About the Author

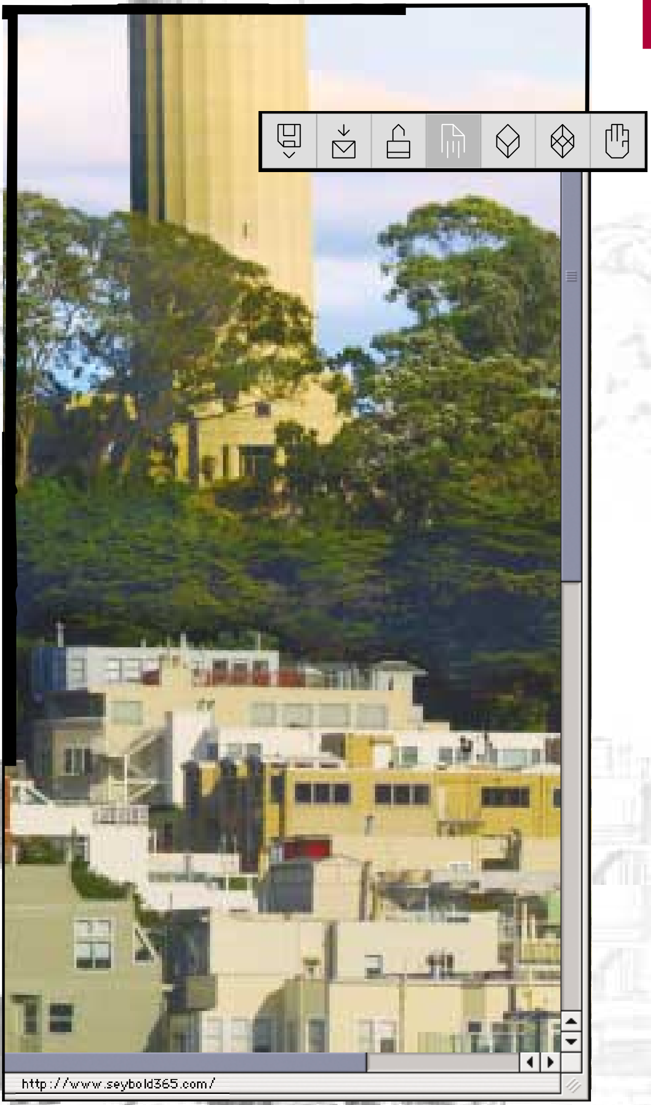
Scott Barstow is chief technology officer for Bandwidth.com, based in Cary, N.C.

■■■ sbarstow-press@bandwidth.com

CMYK

From Concept to Reality

Exclusive Savings!
10% off current Passport Pricing
Register at
www.Seybold365.com/register
Use Promotion Code ADBKYJ47



Don't blink.

The publishing industry is changing by the minute. Now digital on-demand printing allows true one-to-one marketing, automated workflow speeds production cycles and digital asset management banks are filling up fast. The old rules and categories no longer apply—inter-departmental collaboration and cross-media campaigns are changing the way we think about publishing. We've come a long way from color separations and film labs, and we're moving faster all the time.

Learn it here first.

Seybold SF is about expanding the boundaries of possibility. In 2004, we'll help you push them even further, with intensified continuing education designed to keep your skills sharp and your technologies current. If you're a professional in this business, you need to be here—we've developed four new conference tracks designed specifically for your needs and priorities:

- **Creatives** involved in designing and creating compelling content for print and online
- **Publishing professionals** seeking new strategies, best practices and workflows
- **Business professionals** looking for more effective ways to communicate within the organization and beyond
- **Content owners** who implement, manage and support enterprise publishing systems

Leave no colleague behind.

In this business, attending Seybold SF is a necessity, not a luxury. Our special registration offers make it easy for you to bring along your whole team, along with friends and associates at other companies, so they can share in the experience—and in the future of the industry.

Conference: August 16–19, 2004 • Pavilion: August 17–19, 2004 • Moscone West, San Francisco

The Business Event for Publishing Professionals

August 16–19, 2004. Make sure you're on the same page as the rest of the industry—because it's about to turn.

Seybold℠ SEMINARS SAN FRANCISCO 2004

Priority Code ADBKYJ47

MediaLive INTERNATIONAL

Cape Clear Data Interchange 4.7

A solid tool that combines the best of two technologies

No matter how advanced enterprise systems become, the sophistication of the tools available to build them, or the advances of architectural approaches and best practices from which to start, the life blood of any system is the raw data. Without the data, what's the point? Whether the system is simply collecting information from a set of HTML forms, or is one participant in a highly complex, distributed environment consisting of several integration touch points, there will always be the challenge of getting data into and out of the system. Data Interchange from Cape Clear provides a novel approach to this problem by applying the best of what XML and Web services have to offer.

WRITTEN BY
BRIAN BARBASH

Data Interchange is exactly what its name implies, a means to transform data from one format to another. It can transform data to and from XML, SOAP, Comma-Separated Value (CSV), fixed width, structured text (EDI, SWIFT), and Microsoft Excel. The basic process involves four steps:

1. Define the input data format
2. Define the output data format
3. Create a mapping between the formats
4. Deploy the transformation service to the Cape Clear Server

Figure 1, from the Cape Clear documentation, illustrates the life cycle of a typical transformation. During development, if the input or output document is not predefined by an XML Schema, DTD, or WSDL file, the developer must create a Text Schema using the Text Schema Editor. Once laid out, a transformation is created using the XSLT Mapper. Finally, the service is deployed to the Cape Clear Server.

At run time, documents may be submitted in a number of ways. When the document is received, if it is not already in an XML format, it is transformed into XML based on the Text Schema previously defined. The resulting XML document is mapped to the output document using the mapping rules defined in the XSLT Mapper. If the output document is specified by a Text Schema, the data is transformed a second time to its native text or Excel format.

Developing Schemas

For the purposes of this review, I will develop some simple transformations to execute a business process. The first transformation involves a fixed-width text file that contains the names, addresses, phone numbers, and e-mail addresses of a group of people.

Using the Text Schema Editor, it is easy to develop an XML Schema representation of the text document. First, a sample text file is loaded into the editor. From there, defining fields is as simple as highlighting the text that constitutes the field in a row. For this example, each field is fixed length. The Text Schema Editor automatically sets the width of the field based on the highlighted text. Additionally, each field is given a name and a data type (any valid XML Schema type may be used). Optionally, one can specify the default value for empty fields, a regular expression that defines the format of the data within the field, whether data is marked with quotes, what characters to trim in fixed-width files, the alignment of the data within the field, and whether the field should be processed with regards to case. A start and end tag may also be specified, which can be useful if the data is, for example, preceded by a label such as "Address:". It is also important to note that the Text Schema Editor has the power to handle document headers and footers with a variety of information. For example, if the incoming text file contained routing or system information as a data block in the header, it may easily be parsed into an XML structure or ignored, depending on data and system requirements.

Once created, the text schema serves as the parsing rules for a submitted text file. When the file is received, the rules in the schema are applied and the document is transformed to XML. After the input file has been transformed, Data Interchange may leverage the full power of XSLT in the mapping process.

CAPE CLEAR

Company Info

Cape Clear Software Inc.
880 Winter Street, Suite 300
Waltham, MA 02451
Tel: 781 622-2258
Fax: 781 622-2378
Toll-free: 888 CAPE-439
Sales: sales@capeclear.com

Licensing

On a per server basis as part of the Cape Clear Business Integration Suite

Pricing

Starts at $20,000 per server

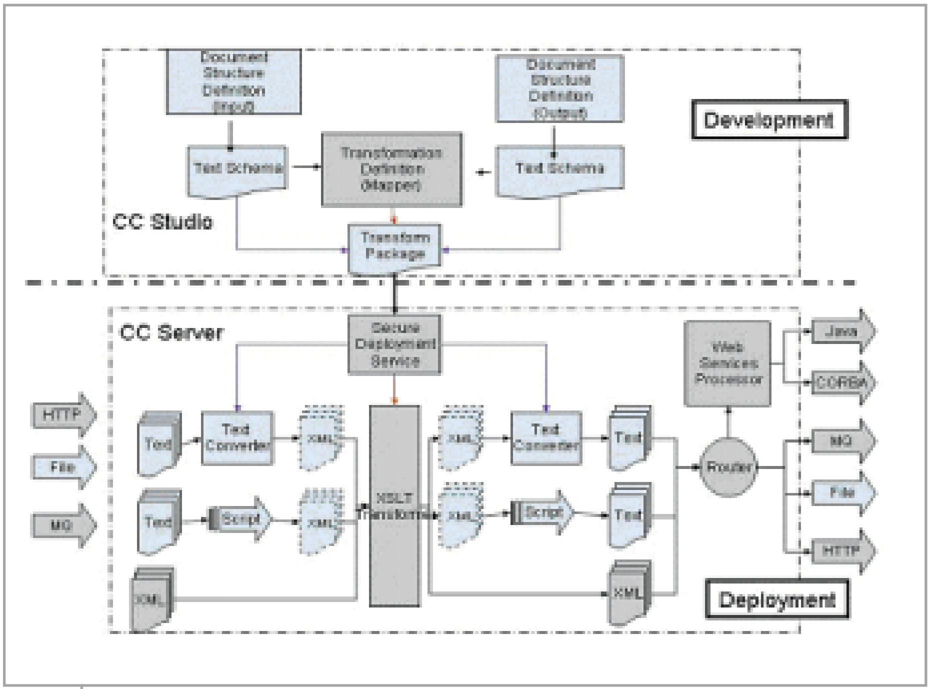


FIGURE 1 **Document transformation**

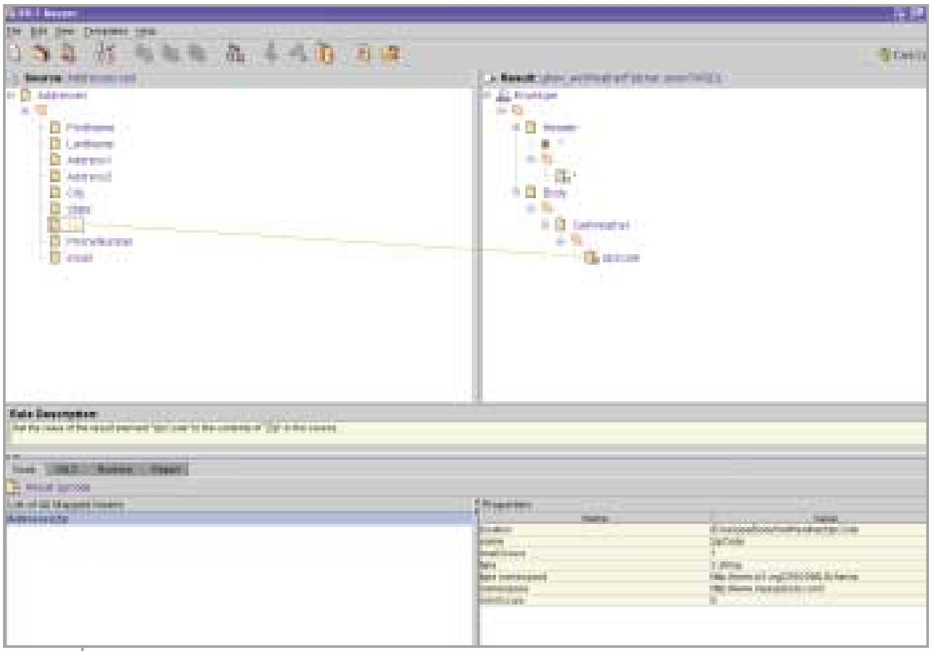


FIGURE 2 **XSLT Mapper**

Mapping Documents

The XSLT Mapper is the tool in which a developer graphically creates a transformation from a source file to a destination file. The interface, shown in Figure 2, is intuitive and follows the general paradigm of mapping interfaces found in similar tools. To create a transformation from the source document to the result document, simply drag the desired node from left to right.

The Mapper supports complex transformation logic by supporting individual transformation rules and XPath statements. XPath statements allow the developer to further isolate the source data into node sets for transformation. Individual transformation rules allow the application of any available XSLT function such as concatenation, substring, arithmetic operators, etc.

Another powerful feature of the XSLT Mapper is the ability to use XSL templates. Templates allow the encapsulation of transformation logic into reusable modules. For example, if processing a document with multiple sets of personal information (i.e., ship-to customer, bill-to customer, sales representative, etc.), the logic to extract the first and last name of each party could be encapsulated into a template. The template can then be applied when appropriate throughout the document without manually repeating the logic.

Performing the Transformation

Now that the XSLT mapping has been created, it may be deployed to the Cape Clear Server. Cape Clear is packaged with a Web service implementation that performs transformations based on mapping documents. To use this prepackaged service, a new project must be defined in Cape Clear Studio. Once the project is created with this transformation Web service, the "Package in a WSAR file" wizard is used to associate the input and output XML Schemas and the input and output XSLT files.

Once packaged and deployed, the XSLT transformation is made available through a Web services interface. In the default configuration, source documents are submitted to the server using SOAP with attachments. The attachment is read, transformed using the Text Schema into its XML representation, mapped using the XSLT Mapping previously defined, and sent to the appropriate output format.

Summary

Cape Clear's Data Interchange applies the best of XML and Web services technology to the problem of transforming and integrating data between systems. The tools for designing and developing mappings are easy to use and provide robust support for many types of file structures. Overall, this is a solid tool that would work well in many integration efforts.

About the Author

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

■■■ bbarbash@sys-con.com

Quality Management for Web Services

The requirements of interconnected business

WRITTEN BY
JAY WEISER

Web services provide organizations with a flexible, standards-based mechanism for deploying business logic and functionality to distributed consumers. Consumers, whether internal or external, can access necessary components such as account information, credit card validation, and much more. When business functionality is distributed, however, quality management becomes imperative. Mission-critical functions, sensitive data transmission, and fee-based services must work quickly and accurately for all users all the time. To ensure that this high level of quality is met, organizations must employ strong test processes to ensure that necessary Web services are developed and deployed to meet the highest standards.

Defining the Service

Quality starts with requirements. In order to design and deliver Web services, developers must know what functionality is needed and under what conditions. The more specific the requirements, the better the development effort. “The Web service must be fast” is useless as a requirement as it is subject to interpretation. “The Web service must have a 2-second response time” is better as this can be measured. “The Web service must provide accurate responses under 2 seconds on a 24x7 basis with up to 500 concurrent users” is even better. Having measurable requirements in place allows developers to work toward a known goal. Defining boundaries like “up to 500 users” gives them scope. Without these boundaries, there is no easy goal and code is likely to be over or underdeveloped.

Likewise, testers must know what functional and performance user requirements have been defined. Developers work on interpreting the user requirements to generate code. Test strategy should be done concurrently to ensure that testers work from the same requirements documents used by development. If test strategy is based on developed code as opposed to initial requirements, the resulting end product may be based on a single individual’s misinterpretation. By creating tests and testing strategies based on the original set of user requirements, at least one more person interprets the project’s needs. By working from the same initial requirements, if differences in requirement interpretation arise, they can be resolved early in the process as opposed to when development is near completion. In this way, the testing group acts as an early check and balance on the software development process.

Once service-level agreements or other quality of service contracts have been established, both parties (provider and consumer) need to agree on what measurements are to be taken and how, what metrics will be reported, what success criteria will be used as a measure of service quality, and what ramifications exist should the provider fail to deliver the agreed upon level of quality. Following deployment of the developed Web service(s), both parties have a vested interest in monitoring the ongoing delivery of the functionality. Web service (WS) testing and monitoring must cover the agreed upon metrics, which should include:

- ***Availability (accessibility):*** The applications must be running and reachable.
- ***Performance (scalability):*** Must respond to requests within acceptable time limits for expected consumer load.
- ***Accuracy (functionality):*** Must be accurate regardless of load.

- ***Standards compliance (interoperability):*** Must work with all potential client types (SOAP stacks).
- ***Integration:*** Must work with all other services and applications within the enterprise.

Managing the Quality Process

Managing the flow of development and testing from inception through deployment requires a great deal of communication. Requirements must be disseminated, business risks assessed, tests defined, results published, and any number of others. Each group and each individual within a given project have a different input into the system and each of these needs something back. For example, developers need the requirements, assumptions, limitations, etc., initially defined in order to code effectively. In return, they provide testable code to quality assurance. They may also provide knowledge in terms of problems encountered, expected problem areas, or other pieces of useful data. In the best scenario, they also provide test client code, scripts, and other unit-level intellectual property thus allowing QA to work more quickly and effectively than they would if starting from scratch.

Throughout the life cycle, different individuals provide data and extract information. Business analysts may input requirements and business risks and report on requirement test status. Project managers input priorities and look at current coverage, status, readiness, and other information to make the call as to when to go live. Even third-party auditors may be involved to ensure compliance with regulations, etc. A collaborative effort must be in place to ensure that everyone gets the information most relevant to them, in a manner that they can understand, while also allowing them to provide relevant data themselves.

> “A collaborative effort must be in place to ensure that everyone gets the information most relevant to them...”

Initial Testing

Unit (Component) Testing

As Web services are generally remote services that provide distributed computational functionality, they do not usually have GUIs from which to test. Developers have several options open to them to test the code they write: create test clients by hand, develop test clients with an Integrated Development Environment plug-in, use dynamically generated clients, or employ model (WSDL) based tools.

Regardless of the testing method, developers are testing the core functionality of the pieces of code they create, ensuring that the Web service does what it is supposed to do as a component. This may or may not include concurrency testing to ensure that multiple users can access the Web service simultaneously.

As coding and unit testing progress, the code, test results, and test clients need to be organized and passed on to the next stage. QA can use this intellectual property to shorten its ramp-up time prior to the next stage of testing. Development should also pass any other relevant information as stated earlier.

Functional Verification

Testing that a Web service or integrated set of Web services functions properly as a whole is generally the next step. The initial requirements lay out what the service should do. Testing that the Web services provide accurate responses to various input data sets ensures that the code is acceptable.

If test clients exist from previous unit tests, they can be reused and potentially modified to step through multiple iterations, testing both positive (works right when proper data is supplied) and negative (generates errors properly when conditions dictate) transactions. These tests should be scheduled to coincide with business risk and project priority as defined appropriately in the earliest stages of the project. Results of these tests then provide an input to the decision-making process.

Issues that arise during testing need to be evaluated with regard to several factors. These include questions like: Do the initial requirements need to be modified? Does any resulting code change invalidate testing that has already occurred? Should this issue be included in future regression testing? And finally, is the developer getting relevant information to help determine the issue’s root cause and thus facilitate resolution?

Most of these questions can be resolved through the use of formal processes in conjunction with full-featured test management and defect tracking mechanisms. The necessary individuals can assess the effects of code and requirement changes on testing, schedules, resources, etc. The final question is best answered by reuse of initial unit testing technology. If the developer is receiving defect information using the same tool, script, or code that was used initially, the developer’s analysis of that information will be quicker due to his or her familiarity with result formats.

Compliancy Testing

Web services are intended to be generic and accessible by any standards-based Web services client. In theory, a Java-based Web service client can access a .NET-based Web service and vice-versa. This, however, is not always the case as there are minor differences between the .NET SOAP stack and the Apache (Java) SOAP stack.

To ensure that a Web service is compliant, tests should be conducted using clients of all anticipated client types. For internal Web services, an organization may be certain that only one flavor of client will access a given Web service, eliminating this need. In all other cases, the unit tests above should be initiated from browsers, Java clients, and .NET clients using various browsers, JDKs, and .NET Frameworks to ensure that the Web service will interact accurately with any and all service request types.

Concurrency Testing

Once functional verification has been completed for single transaction sets, concurrency testing needs to be done. Since their very nature is distribution of business logic, Web services are accessible to numerous consumers. Nearly all Web services can be accessed by more than one consumer at a time. Concurrency testing ensures that the Web service works properly with a small number of concurrent users.

This testing is normally done as a precursor to load or performance testing. There is no point in designing a load test for one or more Web services if they can't handle two simultaneous sessions. Concurrency is simply a small test of a Web service that can be performed manually, through unit tests run concurrently or through small-scale tests of a load testing package. If the performance testing application is reusing code or scripts from earlier testing, using this mechanism makes good sense.

Scalability Testing

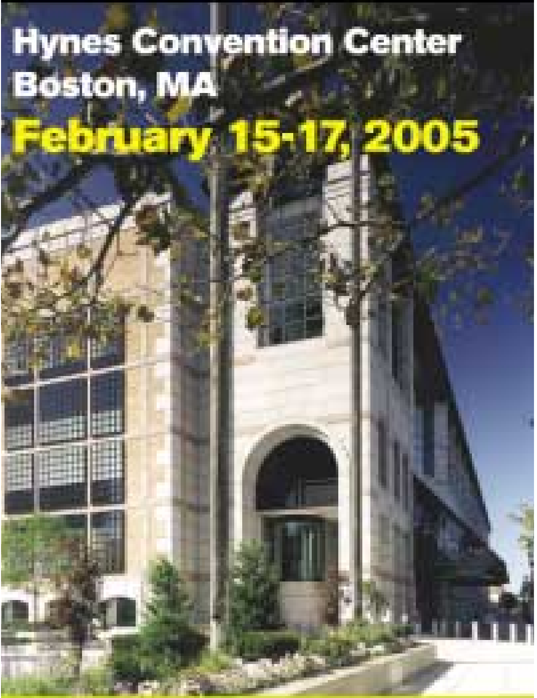
Once a development group knows that their Web services can be accessed concurrently, load testing can begin. The purpose of scalability tests is to ensure proper behavior of the Web service under expected loads.

Scalability tests come in a variety of flavors and each is designed with a slightly different goal. Load testing scenarios include:

- ***Increasing workload:*** Tests start with a small number of concurrent Web service calls and systematically raise the number during the test duration. This test type identifies bottlenecks and determines breaking points.
- ***Uptime:*** Tests provide a steady load on the system for an extended period of time. This test type is used to search for things like memory leaks that may only appear after hours or days of service usage.
- ***All-Day scenarios:*** Tests mix all the various Web service function call transactions in varying numbers throughout an expected normal time period. These tests are usually the final real-world scenario to ensure acceptable performance as measured by the defined service levels.
- ***What If testing:*** Testers may play with any number of unexpected or worst-case scenarios:
  - – What if all potential consumers make the same function call simultaneously?
  - – What if consumers make Web Service calls at varying or unforeseen connection speeds?
  - – What if consumers make calls in an illogical order?
  - – What if consumers make requests from different client SOAP stack types?

This test type is used to predict the less predictable issues that can arise when new applications are deployed to large sets of consumers. These tests are slightly less relevant for internal services, but are still of value.

In all cases, performance testing should also include some level of functional verification. Many complex appli-



cations create errors that occur sporadically under load conditions. If all Web service calls are not analyzed for accuracy, these failed functions may be missed. They are exactly the sort of errors that load tests are designed to generate, therefore it only makes sense to monitor all traffic for errors.

When errors or bottlenecks are uncovered, development needs to get relevant data like exception information, logs, communication headers, etc., to help determine what happened. Once again, reuse of technology allows development to receive this information in a familiar format.

Production Monitoring

Most of the industry literature about service-level agreements falls into the production monitoring space. Once a Web service has been deployed, people understand that it should be monitored to ensure the previously set thresholds are achieved in terms of availability, performance, accuracy, and interoperability as discussed previously. In most operational cases, however, the group monitoring the Web services is not the same as the group who developed and tested it. This disconnect between application development and operations means that the production monitoring must be created from scratch, often with little or no information learned during the development and testing process.

Once again, technology reuse can be a large asset. By providing operational personnel with the same technology that was used to test the application before, time and effort can be saved. More importantly, threshold information, likely problem areas, and other information learned through testing can be transmitted to the production world, giving them a head start on what and how to test. In the event of a failure in production, the reused technology once again means that developers see the information in a familiar format. As always, this shortens the analysis cycle, thereby reducing time to repair and keeping downtime to a minimum.

The Integrated Quality Management Life Cycle

In short, knowledge sharing and technology reuse are core components of ensuring the quality delivery of Web services. By integrating people through good practices and formal processes, each and every individual has the information needed to perform tasks in an optimal fashion. By reusing technology from initial unit testing on the developer's desktop through functional and scalability testing all the way through to production monitoring and back again, testing cycles are not only shortened, but information flow in both directions is enhanced.

Putting all of the pieces together in this fashion keeps Web service development and deployment quality at its highest. Business runs smoothly with critical functionality distributed through these high quality Web services. Consumers get the correct and timely responses to function calls each time, all the time. That's why you started using Web services in the first place, isn't it?

Summary

Today's world of interconnected business entities requires standard computing mechanisms. Web services allow well-defined remote functionality for access by internal and external consumers to be easily developed and deployed. Since various Web services can integrate across numerous providers, quality of service must be maintained at a high level. Managing the quality life cycle from design to delivery ensures that businesses can continue to rely on these deployed services now and in the future. Employing collaborative, integrated, and reusable technology ensures that high levels of quality are achieved and maintained optimally throughout the process.

About the Author

After six years of active duty in the Navy nuclear submarine community, Jay Weiser worked for several software development companies in a variety of roles: technical writing/documentation; technical support; network management; database and systems administration; Webmaster; e-mail and groupware administration; and marketing, quality assurance, and product management, both as individual contributor and as manager. As director of strategic solutions/sales at Segue Software, Jay has a great depth of experience with the entire Segue product line and testing methodologies.

FREE*CD! ($198.00 VALUE!)

Secrets of the Web services Masters

Every WSJArticle on One CD!

WebServices Journal
PRESENTS
$198 VALUE
Web Services
Resource CD
Secrets of the Web Services Masters
Inside:
SPECIAL
Every Issue of WSJ Ever Published on One CD!

— The Complete Works —

CD is edited by WSJEditor-in-Chief Sean Rhody and organized
into 40 chapters containing more than 400 exclusive WSJarticles!

All in an easy-to-navigate HTML format! **BONUS: Full source code included!**

ORDER AT WWW.SYS-CON.COM/FREECD

**PLUS $9.95 SHIPPING AND PROCESSING (U.S. ONLY)*

SYS-CON MEDIA

Only from the World's Leading i-Technology Publisher

Is Your Client Rich?

Someone will be looking over your shoulder

As we look to make more practical use of Web services, the need has emerged for a better user interface; one that's neither too fat nor too thin. An interface that allows developers to make the most out of the client's native features, while at the same time, not bogging the client down with services that are better kept at the back end.

WRITTEN BY
DAVID S. LINTHICUM

We call this new hybrid interface a rich client. A *rich client* is a small piece of software that runs on the client to leverage and aggregate back-end Web services, allowing them to appear as a single, unified, native application. Indeed, a new interface is needed as both developers and end users begin to understand the limitations of traditional Web-based interfaces, which are the current interfaces-of-choice for many distributed applications.

Why a Rich Client?

Truth-be-told, Web interfaces were never really designed to support true interactive applications. The Web was built as a content provider serving up documents and not dynamic application services. If you think about it, you're reloading document after document to simulate an interactive application, and always have to go to the back-end Web server to request new content. Very little occurs at the client.

As the Web became popular and we looked to support business applications using the Web interface, we began to create new mechanisms to deliver dynamic content including dynamic HTTP/HTML pushers (e.g., CGI, ASAPI, ISAPI) and new browsers that supported complex dynamic behavior. We are at such an advanced state today that entire enterprises run most of their relevant business applications using Web interfaces.

However, with the advent of Web services, and the need to leverage dynamic behavior within the interfaces, traditional browsers fall short. Their get/push model for driving interfaces is not as well suited for Web services, which are – at their essence – remote functions, and are better suited for more visually rich types of interfaces, such as the traditional GUI client/server interfaces that were popular a few years ago.

Rich clients are not a revolution, but an evolution of technology. Today we look to leverage dynamic behavior and deliver that experience directly to the end user, aggregating Web services within an interface that appears as much like a native application as possible.

As stated above, rich clients provide capabilities that thin clients are not able to provide, including windowing features and data navigation control such as buttons, check boxes, radio buttons, toggles and palettes. They are also able to integrate content, communications, and platform-independent application interfaces for distribution through emerging SOAs. The rich client becomes a Web services terminal of sorts, allowing applications to communicate and even execute on one another within a distributed environment.

Who's Rich?

Of course, all of the major vendors who support Web services are all over this, including Microsoft and Macromedia, both creating dynamic rich clients for Web services. In addition, new Web-based forms technology, such as Microsoft InfoPath; and the new versions of Acrobat products will also serve as rich clients.

Microsoft Office, for instance, is able to retrieve information directly from Web services as if the services were local to the client. This capability means that sales figures will magically appear inside of Excel, and you'll see the daily inventor report show up dynamically within a PowerPoint presentation. No need to scrape a screen or download a file. Moreover, behavior will be delivered in such as way as well, for instance, having dynamic visibility directly into a risk analytics application directly from Microsoft Word.

Macromedia, known for their popular Flash interface, has created a language called Macromedia Flex Markup Language (MXML), which allows Flash developers – numbering in the tens of thousands – to create, run, and execute Web services within the client-side Flash environment. This means access to corporate applications and information from some very creative interfaces, running anywhere that supports Flash.

What's Next?

This is great news for those of us who are developing Web services or implementing an SOA. With the use of rich clients, suddenly those services have a much higher value. Indeed, you can mix and match services within a rich client to create some very valuable applications. Perhaps, someday, the use of static and dynamic HTML and heavyweight protocols such as HTTP will not be the primary way we view distributed applications. Rich clients give us the ability to view applications that look and act like native client pro-

> "Truth-be-told, Web interfaces were never really designed to support true interactive applications"

> “In the future, we’re going to live in a world where all applications have visibility into all other applications”

grams, albeit running remotely. That would be a step in the right direction.

However, there are some hurdles to cross, including standards and performance.

As rich clients become more popular and vendors see them as a revenue opportunity, you can count on any number of rich client standards to appear from the major players. As different standards appear, it will be difficult to mix and match Web services within particular rich clients due to differences in the standards. Moreover, the appearance of many different rich client standards could both confuse and alienate end users. We’ve been here before with many new technologies.

Performance is also a consideration. While rich client applications may appear as native applications, it may take a few seconds before communications can be made to some of the back-end Web services supporting the rich client. Thus, many who use rich-client applications may find them slow and unresponsive when compared to native client applications that can respond in milliseconds. This limitation will have to be addressed if the technology is to become pervasive.

I’m not sure this is a huge shift in thinking; only a return to the way we think applications should look and feel. What is new is the notion that composite applications are the way to go when building new or enhancing existing applications, and that it’s okay to create an application out of many remote applications. We’ve never been able to make this fly until recently, albeit we’ve had previous generations of technology that leverage the same sort of architecture.

In the future, we’re going to live in a world where all applications have visibility into all other applications...if they have permission, that is. A world where application behavior and information are sharable commodities and most users won’t need to know where their applications or their information resides, as long as they have access to each. And that is a rich experience. ⓔ

About the Author

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

■■■ e-mail: linthicum@att.net

Building Your First Trading Partner Community

A competitive edge down the road

Web services seem like a great way to tackle the challenge of trading partner integration. In theory, participants can continue to use the enterprise applications they prefer, and expose them to their partners' systems.

WRITTEN BY
SCOTT BUSHEY &

TYSON HARTMAN

However, as of late spring, Web services were most prevalent among services companies and least common among retailers. The reality is that few companies have experience with integration across organizations. The default seems to be that the largest participant or customer determines the process, but rarely is thought given to the actual process of integrating applications across organizations, how to handle exceptions, or how to resolve differences. Barriers to success may range from a disparity in skill levels, to familiarity with Web services, or maturity of the IT organization.

The difference in adoption rates among services companies and retailers is ironic given that in our experience, a much older technology used by the retail and manufacturing sectors – electronic data interchange (EDI) – offers useful lessons for Web services. What's more, evolving Web services standards such as WS-I Basic Profile specifications can help eliminate haggling over definitions.

The good news is that companies can take a step-by-step approach to using Web services to establish a functional trading partner community now and enhance it later.

Relationship Management

EDI has been used as a means to automate communication among trading partners (see sidebar, "What is EDI?") since the 1960s. Company-specific document definitions quickly proliferated, and any time savings were quickly overtaken by the need to adjust systems to accommodate new, partner-specific document definitions. EDI required staff dedicated to relationship management, and gave rise to the value-added network (VAN) to facilitate issues resolution.

Relationship management is a consideration as well for companies choosing Web services to connect their supply chain. At the outset, trading partners will need to agree on certain rules of engagement to minimize the impact of disparities in experience with Web services, to ensure some degree of standardization to achieve interoperability, and to accommodate participants' choice of technology and platform. These efforts help reduce barriers to entry for future participants, too.

Most trading partners rely on the largest participant to serve as the arbiter – typically a customer requiring supply-chain integration or some other process that connects several vendors. The arbiter effectively becomes the center, or "hub," for trading partner interaction, and the other participants become "spokes."

It's a relationship model that's served well for EDI; Web services offer the advantage of a non-application–specific exchange with tremendous flexibility – and theoretically, fewer demands on staff time to handle technical issues.

There are three primary issues to address at the start of a Web services–enabled trading partner community:

- Establishing a standards base
- Lowering barriers to entry for new participants
- Interoperability

Standards

Agreement on several fundamental aspects of interaction will go a long way toward getting a trading partner community up and running. These elements include what data will be exchanged, the format of that data, the transport mechanism(s) used to communicate it, the relevant interfaces to other participants' systems, and the overall approach to implementation.

Fortunately, the major players in the technology industry have defined the fundamentals of a basic Web service. As members of the Web Services Interoperability Organization (WS-I), companies including Microsoft, Sun, Avanade, and IBM have collaborated on the WS-I Basic Profile 1.0 international specification. Basic Profile 1.0 addresses Web service elements including messaging (using SOAP and XML); service description using Web Services Description Language (WSDL); and service publication and discovery using Universal Description, Discovery, and Integration (UDDI) (see sidebar, "A Word about WSDL").

With a suite of pretested specifications, trading partners can focus on choosing tools and on establishing criteria for minimum conformance. Check whether vertical standards further reduce the need for negotiation in the early stages of trading partner community formation. For example, the mortgage banking industry's Mortgage Industry Standards Maintenance Organization (MISMO) and the insurance industry's Association for Cooperative Operations Research and Development (ACORD) have established business process standards as well as message schema that may be useful to reference.

That said, some "horizontal" business process execution integration specifica-

What Is EDI?

Electronic Data Interchange (EDI) is the computer-to-computer exchange of business data in standard formats. In EDI, information is organized according to a specified format established by the parties exchanging data, allowing a computer transaction that requires no human intervention or re-keying of information. Today, more than 300,000 organizations use the 300+ EDI transaction sets to conduct business.

A Word about WSDL

Setting a standard for Web services description is perhaps the biggest enabler of successful integration. It is crucial for companies to know what interface requirements they must meet in order to integrate with trading partners, and establishing a single interface standard for all "spokes" of the community is much preferable to having multiple interfaces – and the potential for complexity and even failure. It's another reason we recommend clients consider adopting Basic Profile 1.0.

tions under development may introduce complexity and undermine chances for the success of the trading partner community. Basic Profile 1.0's basic Web services–enabled functions can be augmented with specifications for more sophisticated actions later on. Standards to consider after the community is up and running could include advanced security procedures such as encryption, certificate standards, and digital signatures.

Once participants have agreed on standards, it's important to withstand exceptions. The Web services–enabled trading partner community provides a universal foundation for integration. Deviation from core specifications will make it difficult to add new participants later, not to mention threaten the chances of smooth integration among the inaugural partners. It's important not to underestimate the amount of effort involved in setting these standards, but the investment will pay off in multitudes over time.

Conformance – Lowering Barriers to Entry

Besides standards for participation, a level playing field is vital to the success of the trading partner community. Standards selection requires complementary skills to implement them. That's why it's extremely important to take stock of trading partners' capabilities and even certain aspects of their infrastructure. Participants may or may not have IT organizations that are well-staffed. Personnel may or may not be well-acquainted with technology and industry standards. And they may or may not have past experience with Web services and related technology.

The "hub" of the community must acknowledge the difference in competencies, and take the lead in establishing the relative skill of each participant. Then the group must factor this assessment into standards selection, technology choices, and timeframe for implementation. In some instances, "hub" companies may provide some form of support to partners with less advanced IT capabilities, supply sample code, or even recommend a complete "black box" solution.

A high-level infrastructure assessment will help determine which Web services characteristics to delay for future implementation. For example, a "hub" may specify digital encryption for every Web service call, but "spokes" may not have the public key infrastructure (PKI) to support that sophisticated function. The community may default to Secure Socket Layer (SSL) as a means for network security to lower the barrier to entry into the community, sacrificing the benefits of PKI in the short term in order to expedite the community's launch.

The format of Web services ultimately must reflect the capabilities of the group, as well. There are two methods for exchanging data: the simpler document–literal approach, and the more complex remote procedure call (RPC) approach. Document–literal exchanges specify the format and content of data (such as a purchase order) and tend to be more message-based. RPC–based Web services invoke action on the data conveyed and tend to be more stateful. Document-literal transactions have a smaller payload and are easier to implement.

Of course, the vision for Web services is a multiplatform trading partner community – one where participants may elect the development environment they prefer, whether Python, J2EE, or Microsoft .NET, on the platform of their choice such as Linux or Microsoft Windows Server 2003. Nevertheless, until the Web services experience is universal, the multiplatform environment can be the most challenging. Taking inventory of participants' platforms can help the community understand whether there could be

WSJ ADVERTISER INDEX

ADVERTISER	URL	PHONE	PAGE
Active Endpoints	www.activeendpoints.com		17
Assande	www.assande.com		23
ClearNove	www.clearnova.com/thinkcap		31
CTIA	www.ctia.org		Cover III
IDG	www.linuxworldexpo.com		8&9
Information Starage+Security	www.ISSJournal.com	888-303-5282	37
IT Solutions Guide	www.sys-con.com	201-802-3021	25
Mindreef	www.mindreef.com		11&13
Open Link Software	www.openlinksw.com/virtuoso	800-495-6322	Cover II
Parasoft	www.parasoft.com/soaptest	888-305-0041	5
Prolifics	www.prolifics.com	800-675-5419	Cover IV
Seybold	www.seybold365.com/register		27
Speech Tek 2004	www.speechtek.com	877-993-9767	51
StrikeIron	www.strikeiron.com	919-405-7010	6
SYS-CON e-newsletters	www.sys-con.com	888-303-5282	21
SYS-CON Publications	www.sys-con.com/2001/sub.cfm	888-303-5282	41
Web Services Edge East	www.sys-con.com/edge	201-802-3004	34
WebAppCabaret	www.webappcabaret.com/wsj.jsp	866-256-7973	3
WSJ Resource CD	www.sys-con.com/freecd	888-303-5282	35
XML-J Resource CD	www.sys-con.com/freecd	888-303-5282	45

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

“Proactive support is essential in the post-implementation phase of the trading partner community”

additional difficulties. In turn, they might redefine aspects of Web services collaboration so that they can be supported easily by all partners.

Avoiding Interoperability Issues

Once standards are in place and steps have been taken to inventory capabilities and ensure every participant has relatively equal footing, implementation can begin. Frequent communication among “hub” and “spokes,” necessary in the early planning stages, is absolutely essential in this phase. The rule of thumb we advise is for companies to talk at least once a week and plan to test frequently all the way through production.

Communication can smooth the way for a fairly uneventful implementation. It also sets a precedent for problem resolution. Programming may begin and code may not work – unfortunately, because it is possible for a partner to hide its lack of capabilities by writing code that looks like Web services via scripting, but that doesn’t actually implement them, because there is no logic and no infrastructure in place for SOAP and/or WSDL.

In one unfortunate situation, a client’s partner didn’t invest in the necessary development environment and tools, and managed to write scripts that fooled the teams until tests drew responses that were erroneous. The client’s first response was to examine the code line-by-line to find the error; however, we examined the URLs to which the partner’s WSDL files were pointing and discovered that the partner had utilized extensive scripting that did not truly implement Web services.

An agreed-upon inventory of skills, experience, and platforms – as well as regular communication – should uncover the need for assistance and also help teams more quickly determine the source of problems by eliminating certain possibilities. A carrot-and-stick approach may be required, to offer – and take away – additional support for non-conforming partners who need access to other experts in the community.

Several tools for code comparison also help you diagnose certain issues quickly. For example, Mindreef SOAPScope and Microsoft .NET Web Services Studio 2.0 make it possible to compare trading partners’ WSDL interface to the agreed-on standard, validate their interfaces against WS-I standards, capture Web services traffic sent during testing, and analyze it to make sure it meets standards and is correctly formatted. Since Web services give participants some flexibility in designing their interfaces to other partners’ Web services, this automated comparison saves a good deal of time. Some tools will allow resending of RPC or document–literal messages for quick diagnostic testing without running the application itself.

Another bump in the road to interoperability is access to trading partners’ infrastructure. A host of issues can bring work to a halt, from changing firewall settings to setting permissions on servers, to network and machine administration ensuring a stable connection to partners outside the firewall. One aspect of Web services development that’s foreign to all organizations is the need to expose development – as well as production – to the outside world. The challenge of consistent Web services management within a trading partner community shouldn’t be overlooked.

As implementation advances, it’s important to keep track of each participant’s progress as well. One trading partner might have migrated its work to its test bed, while another remains in development, and a third has moved on to its quality assurance environment. Participants in the Web services–enabled trading partner community must keep track of where each one is in its work, even using a manual mechanism to do so.

Finally, it’s important to build into Web services some mechanism for error tracking among organizations. This may be as simple as an asynchronous error notification that annotates events with a tracking number, so that companies can reference both instance number and time of event. Referring to the time of event alone has limited value, in cases where the error results in some action not taking place – such as a rejected purchase order or invoice.

Moving Forward

As participants’ expertise grows, no doubt the trading partner community will choose to enhance aspects of its Web services once up and running. In our experience, security usually is the first subject of augmentation, whether through the addition of encryption or digital signing. Next in line is the exposure of more internal business applications in order to expand the trading partner community or add more business processes. Other architectural services may be introduced for reliable messaging, routing, and management as well.

Proactive support is essential in the post-implementation phase of the trading partner community. When Web services operate around the clock, year-round, identifying a problem or error as soon as possible will help a trading partner address an issue before it ripples throughout the supply chain. Mechanisms to trigger proactive support range from human intervention – such as periodic review of event logs – to automated tools, to application-initiated notification via e-mail.

Although retailers may lag other industries in Web services adoption, next-generation Web services tools are getting easier to use, and have richer features. Ease and cost-effectiveness will tip the scales in favor of Web services and away from EDI and value-added networks, and gaining experience now with Web services integration will serve retailers looking for a competitive advantage very well.

Reference

- Forrester Research, “Who Has How Many Web Services,” May 10, 2004.

About the Author

J. Scott Bushey and Tyson Hartman are consultants with Avanade Inc., a technology integrator for Microsoft solutions in the enterprise. Bushey, an Avanade Principal Solutions Developer, is responsible for architecting and developing .NET solutions, focusing on Web services. Hartman is an Avanade Fellow, one of Avanade’s senior-most technologists responsible for guiding Avanade’s .NET vision and go-to-market solutions.

■■■ jsbushey@avanade.com

■■■ thartman@avanade.com.

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!*

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

□ **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

*WHILE SUPPILES LAST. OFFER SUBJECT TO CHANGE WITHOUT NOTICE

RECEIVE YOUR DIGITAL EDITION ACCESS CODE INSTANTLY WITH YOUR PAID SUBSCRIPTIONS

3-Pack

Pick any 3 of our magazines and save up to $210⁰⁰ 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 $340⁰⁰ 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 $270⁰⁰ Pay only $399 for a 1 year subscription plus 3 FREE CDs

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

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

SYS-CON MEDIA

Web Services Integration Brokers and Enterprise Application Integration

Preparing for the future

Integration brokers are middleware platforms for complex enterprise application integration (EAI), enterprise information integration (EII), and business-to-business (B2B) integration.

They support flexible any-to-any integration and they provide orchestration engines that allow organizations to implement business processes that span various applications, including enterprise resource planning (ERP) and customer relationship management (CRM), and various platforms, including Windows, Linux, and Unix.

Integration brokers couple orchestration engines with flexible transformation and routing features to support multipoint application, process, service, and/or data integration. Integration broker vendors – such as BEA, IBM, Microsoft, Oracle, Savvion, SeeBeyond, Vitria, webMethods, and others – provide process-definition tools as well as connectors for interfacing to various application, operating, and networking environments.

Integration Broker Value Proposition

Integration brokers provide an environment within which business processes can be rapidly integrated and just as rapidly revised. A flowcharted orchestration can become the template for rapid business-process reengineering. An orchestration-enabled organization can rise to new challenges and reform itself rapidly into whatever new business model or value chain suits the task at hand. By redefining rules, roles, and routes within a process map, organizations can integrate, aggregate, and orchestrate existing Web services in new ways.

WRITTEN BY
JAMES KOBIELUS

However, general-purpose integration-brokering infrastructures are still the exception – not the rule – in most organizations. Most orchestrated processes are limited to specific organizations, platforms, and application suites. System integrators typically use integration brokers as the basis for custom integration projects. By the same token, many enterprises invest in application suites that have embedded orchestration functionality.

Nevertheless, enterprises should begin to implement general-purpose Web services orchestration infrastructures, both within and between organizations and domains. Orchestrated process models can drive structured interactions across any and all tiers of the network application environment, including presentation, business, and integration environments.

Integration Broker Functionality

Multipoint service orchestration is the principal feature of integration brokers, and is an important ongoing requirement in most heterogeneous application environments. The term “orchestration” refers to the rule-driven flow of information, context, and control among diverse environments within distributed business processes. Integration brokers execute reusable orchestration process definitions that control multistep interactions across complex environments. When executing an orchestration, an integration broker may enforce application-level transactional context and apply rules regarding content transformation, message routing, and process invocation.

Multipoint orchestration brings isolated functionality silos together into a cohesive architecture, with integration brokers providing critical connection points in the shared communication bus. While point-to-point connections can be used to bridge isolated application silos, they don’t enable efficient creation of composite business processes. In addition, point-to-point integration complicates the process of upgrading service interfaces. Coordination across diverse applications, services, and other networked resources is difficult and costly without a general-purpose integration-brokering environment.

Integration Broker Architecture

Integration brokers go by many synonymous names in today’s middleware, platform, and tool market. Some of these other names include integration platforms, integration servers, orchestration servers, business process management (BPM) tools, EAI tools, enterprise service bus (ESB) middleware, process managers, process engines, and workflow environments. What all such products have in common are the following baseline components:

- ***Orchestration engines:*** All products provide runtime, server-based orchestration engines that support multipoint, any-to-any application, service, process, and/or data orchestration. All support asynchronous, messaging-based interactions among application and process endpoints, and many also support synchronous interactions. Usually, the runtime engine enables integration through message/document validation, transformation, and routing.
- ***Resource connectors:*** All provide a broad range of resource connectors for connection to diverse application, platform, middleware, transport, and data environments. Increasingly, vendors boast of offering more than 200 connectors out of the box with their integration brokers.
- ***Visual tools:*** Most vendors provide visual orchestration process definition, mapping and transformation, administration, and reporting and analysis tools. Developers often use graphical tools to specify the orchestration tasks, dependencies, and routing steps that drive runtime execution by integration brokers. Some orchestrations may involve long-running composite business processes, while others may involve short-lived atomic transactions. Often, orchestrations can be published as higher-level Web services.

Figure 1 provides a graphical overview of the components of an integration broker.

Integration Broker Features

Core orchestration features of integration brokers include:

- ***Resource connection:*** Interfacing to applications using the Web services framework and/or various transport and middleware protocols
- ***Message validation:*** Receiving, parsing, and validating parse messages and documents originated by applications
- ***Policy enforcement:*** Applying appropriate security policies and controls – such as authentication, authorization, decryption and encryption, digital signing, and content filtering – to in-flight messages before they reach the invoked target applications
- ***Content mapping and transformation:*** Mapping and transforming the contents of messages exchanged between those platforms (e.g., translating the outputs of one application into the correct input syntax/format of another application and rewriting packet headers and contents)
- ***Message routing:*** Transmitting and routing messages to target/server applications
- ***Transaction tracking and coordination:*** Executing, controlling, tracking, and/or monitoring the end-to-end transactional workflow of messages routed between two or more application endpoints.

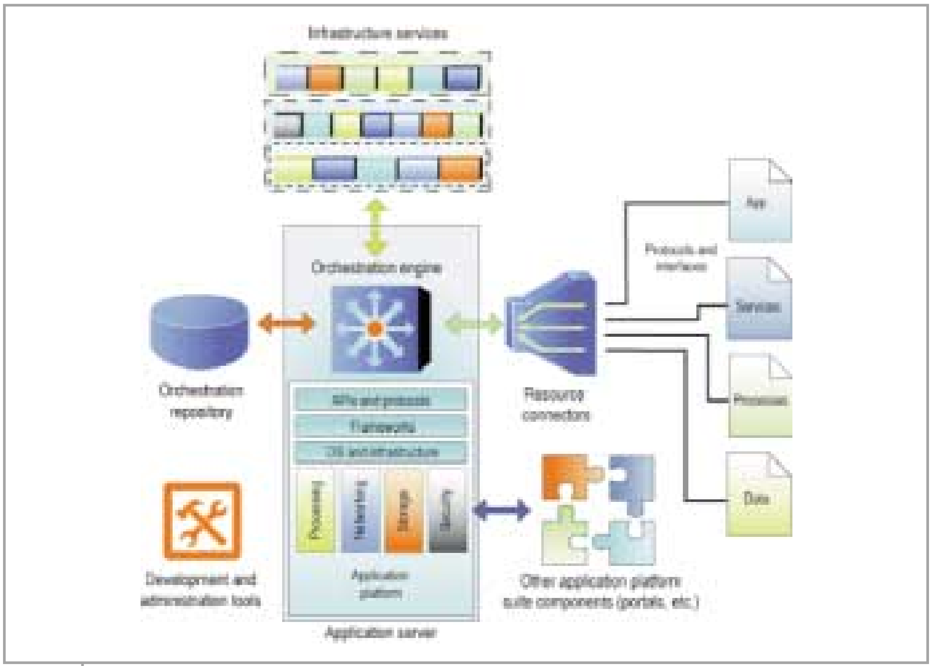


FIGURE 1 **Integration broker components**

Integration Broker Market

Fundamentally, an integration broker is any runtime rules engine that supports multipoint integration and orchestration of diverse applications, processes, services, and/or data. Even when defined this narrowly, today's integration broker market includes dozens of vendors. The integration broker market is crowded, immature, and volatile. The integration broker segment has only emerged as a well-defined niche of the middleware market in the past five to seven years, as pioneering vendors such as webMethods and Sonic Software began to provide workflow engines and tools that addressed integration requirements of e-business over Web environments.

The integration-broker market is just one component – albeit the fastest-growing segment – of the middleware market. As a technical approach, integration brokers complement such established integration approaches as object brokering (such as Common Object Request Broker Architecture [CORBA] and Distributed Component Object Model [DCOM]) and message-oriented middleware (such as Java Message Service [JMS], WebSphere MQ, BEA Tuxedo, TIBCO Rendezvous, and Microsoft Message Queuing [MSMQ]). In fact, many integration broker vendors got their start in traditional middleware approaches, and established middleware vendors such as BEA, IBM, IONA, Microsoft, Sonic, and TIBCO have decisively shifted their product focus to their integration broker offerings.

Today's core integration broker market includes two broad segments, based on how various vendors' offerings fit into broader product families: application platform vendors and middleware platform vendors.

Application Platform Vendors

Integration-broker vendors in this category provide tools that enable life-cycle development, deployment, and orchestration of services, applications, processes, and data. At the very least, these vendors provide integration brokers, application servers, and portal servers. Vendors in this segment include BEA, Fujitsu, IBM, InterSystems, Microsoft, Novell, Oracle, SAP, Sun, Sybase, and webMethods. They can be further subdivided into the major application-server vendors – BEA, IBM, Microsoft, Oracle, and Sun – and the others.

> "Integration brokers provide an environment within which business processes can be rapidly integrated and just as rapidly revised"

Middleware Platform Vendors

Integration-broker vendors in this category provide tools that focus primarily on orchestration of services, applications, processes, and data. These vendors provide integration brokers, resource connectors, and visual process definition and administration tools, just like the application platform vendors. However, middleware platform vendors lack application servers of their own, though they sometimes integrate with third-party application servers and integrated development environments (IDEs). Some vendors' solutions are primarily oriented to application integration (i.e., the EAI vendors), while others (the EII vendors) primarily address data integration (though many vendors' solutions do a bit of both). EAI vendors with integration brokers include Collaxa, Commerce One, Fuego, Intalio, IONA, Lombardi Software, Oak Grove Systems, Pegasystems, Savvion, SeeBeyond, Sonic Software, TIBCO, Ultimus, Vitria, and others. EII vendors with integration brokers include Ascential, Attunity, Cape Clear, Certive, Pervasive, PolarLake, TranSenda, Xaware, and others.

Increasingly, integration brokering is becoming a core feature of leading application server platforms, such as BEA WebLogic, IBM WebSphere, Microsoft Windows Server System, and Oracle Application Server. We expect this trend to accelerate as application server vendors strive to distinguish themselves through increasingly more sophisticated suites for developing, deploying, integrating, orchestrating, and managing distributed services. Web services–based middleware is fast becoming one of the core feature sets of the morphing network application environment, and every application server vendor wants to become its customers' principal "all in one" environment.

We expect to see many EAI integration broker vendors migrate toward becoming full-featured application platform vendors by adding application servers and IDEs to their core product families. In the past year, we saw webMethods – one of the pioneering integration-broker EAI vendors – take this step when it acquired The Mind Electric along with its Web services platform, middleware, and development tool offerings. We expect that many integration broker vendors will evolve in this direction through increasing integration with open source application servers such as JBoss and Apache Tomcat, and with open source IDEs such as the Eclipse platform.

Feature Innovation Raising Bar Across Integration Broker Market

We will also see continuing feature enhancements by all integration broker vendors as the industry common-denominator functionality continues to evolve. The following features will become standard in all or most commercial integration broker offerings (from both application platform and middleware platform vendors) over the next two to three years:

- Support for a broad range of EAI, EII, B2B, and human workflow requirements
- Implementation of standards-based orchestration process syntaxes, especially WS-BPEL, for enabling multivendor process definition, interchange, execution, federation, monitoring, and control
- Increasing development orientation toward visual process modeling notations – such as Unified Modeling Language (UML) and Business Process Modeling Notation (BPMN) – that abstract developers from underlying process definition syntaxes (such as WS-BPEL)
- Introduction of visual process-definition tools for use by nontechnical business process analysts
- Integration of the growing range of WSF standards (in lieu of non-WSF legacy interfaces) for identity, security, reliable messaging, transactions, semantic interoperability, and other core middleware, management, and policy functions
- Integration of Web services management (WSM) features such as lightweight message processing agents that monitor traffic performance and enforce service-level agreements (SLAs) through real-time, dynamic content routing and optimization
- On-demand downloading of software that can be used to rapidly configure a remote node to communicate with their integration brokers (a feature that will eventually be rendered unnecessary, due to universal platform implementation of the WSF stack of standards)
- Deployment of functionally specialized instances of integration brokers across enterprise application infrastructures
- Flexible configuration of specialized integration brokers throughout application infrastructures
- Inclusion of more sophisticated visual tools for defining, administering, and tracking orchestrations across multiple domains
- Interfaces to same-vendor and external UDDI and trading-partner registries for service registration, finding, and binding
- Incorporation of a broader range of process accelerator templates for various vertical markets

FREE*CD! ($198.00 VALUE!)

Secrets of the XML Masters

Every *XML-J* Article on One CD!

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

— The Complete Works —

CD is organized into 33 chapters containing more than 450 exclusive *XML-J* articles!

All in an easy-to-navigate HTML format! **BONUS: Full source code included!**

ORDER AT WWW.SYS-CON.COM/FREECD

**PLUS $9.95 SHIPPING AND PROCESSING (U.S. ONLY)*

SYS-CON MEDIA

Only from the World's Leading i-Technology Publisher

> “General-purpose integration-brokering infrastructures are still the exception – not the rule – in most organizations”

- Hosting of integration-brokering functionality for access by remote sites and B2B trading partners
- Integration with multivendor identity, access management, and security infrastructure through standard interfaces such as WS-Security, Security Assertion Markup Language (SAML), and others
- Administration of business rules separately from the business process orchestrations with which they are associated through incorporation of rules-inference engines that extract rules and metadata from existing application code
- Incorporation of portals and presentation-oriented orchestration development features – such as e-forms and worklists – into vendor product offerings
- Integration with e-mail, instant messaging, mobile messaging, and other asynchronous and synchronous communications environments to transmit alerts, notifications, and other messages to end users
- Access to orchestration platform functionality through same-vendor portals or, via portlets, through third-party vendor portal environments

When universal vendor support for WS-BPEL and other orchestration standards emerges, it won’t make much competitive difference among integration broker vendors. Open orchestration standards will be implemented universally within platforms and tools, and will be invisible to most developers and users, just as Extensible Markup Language (XML), SOAP, WSDL, and other WSF specifications – though ubiquitous – are essentially invisible to most information technology (IT) professionals.

WS-BPEL is one of many WSF specifications that integration broker vendors must integrate natively if they wish to survive and thrive in this market. An open orchestration infrastructure depends on a broad range of WSF specifications supporting critical network application environment infrastructure features, including standards-based application messaging, service contracts, service brokering, addressing, reliable messaging, and event notification. Consequently, standards such as WSDL, UDDI, SOAP, WS-Reliable Messaging, and WS-Eventing are just as important for integration broker interoperability as WS-BPEL. These specifications will have a powerful disruptive impact on the current integration broker market, reducing vendor lock-in. Over the next three to five years, ubiquitously WSF-based EAI and B2B environments will enable any-to-any mix-and-match of integration brokers, connectors, and design/development tools from diverse vendors.

> “To focus the decision process, enterprises should identify the integration scenarios in which to deploy integration brokers”

Recommendations and Conclusion

Integration brokers are central to enterprise platform, middleware, development, and management strategies. Consequently, companies must base their decisions on whether, when, and how to implement integration brokers on many business and technical considerations.

Bear in mind that all vendors discussed in this article provide the basic multipoint orchestration features discussed above: resource connection, message validation, policy enforcement, content mapping and transformation, message routing, and transaction tracking and coordination. Most vendors provide strong visual development and administration tools, diverse resource connectors, and browser-oriented business activity monitoring and control.

Beyond those core features, though, integration-broker vendors vary broadly in their development and deployment options, orchestration functionality, and legacy integration. To focus the decision process, enterprises should identify the integration scenarios in which to deploy integration brokers. Typically, integration brokers are appropriate in multipoint integration projects involving heterogeneous operating environments, application platforms, middleware protocols, and data resources.

Enterprise customers are implementing integration brokers for a growing range of middleware and orchestration requirements, and will increasingly call for standards-based multivendor interoperability. Federated, standards-based orchestration is coming to complex e-business environments everywhere. Integration brokers are the critical infrastructure that will make this possible.

About the Author

James Kobielus is a senior analyst for Burton Group (Alexandria, VA). He has over 18 years of experience in telecommunications and distributed computing as an industry analyst, consultant, and product manager. His current focus areas include operating environments, application development frameworks, Web services, enterprise application integration, middleware, and business process management/orchestration. James has published books on business process management and enterprise application integration. He also writes a regular column, “Above the Cloud,” for Network World.

■■■ jkobielus@burtongroup.com

XML JOURNAL SUPPLEMENT

This Month

Opening the Black Box of Integration

BY MIKE LEHMANN

The next wave of integration technology is becoming clear. Emerging Web services standards for orchestration and choreography offer a fresh, standardized way of tackling business process integration. Previously this was a black box, a proprietary area that was different from vendor to vendor. The new standards target an area central to every integration project, the actual business processes; this article navigates these emerging standards.

Easy XML Publishing into Your Enterprise Portal

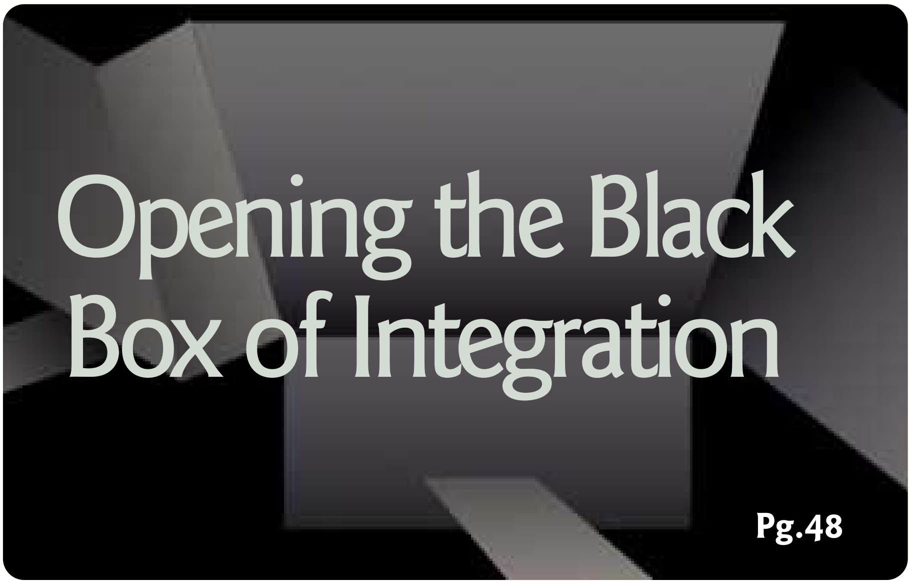
BY PETER MOSKOVITS AND SUE VICKERS

The work needed to integrate XML with a portal is considered highly technical and time-consuming. For example, to publish XML into a portal, developers usually think that programming is their only option. They go through the entire process of using Beans, compiling XML Schemas, and processing and binding XML instance documents. This article covers the options available to both page designers and developers to simplify XML publishing.

The Evolution of Data-Oriented Services

BY MATTHEW ZAGER

This article presents situations in which applying XML and Web service technologies in a data-oriented manner provides system interoperability and enhances the ability of systems to more quickly adapt to changing requirements. It also explores the use of Web services and presents self-services, a concept to allow clients to get exactly the right information in exactly the right format on-demand.



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

HOME
ENTERPRISE SOLUTIONS
CONTENT MANAGEMENT
DATA MANAGEMENT
XML LABS

Opening the Black Box of Integration

WRITTEN BY
MIKE LEHMANN

Moving toward a standardized integration architecture

If you've been working with integration technologies for any length of time, you're well aware of the freight train of standards that has been careening through the industry during the last five years. These standards, particularly in the Web services space, are on the verge of doing to proprietary integration servers what SQL and J2EE standards did to database and middle-tier servers of days gone by.

Database veterans remember when Cullinet ruled the roost; many felt it had no technical equal. SQL was an interesting idea, but not for real projects. Likewise, long-time middleware developers recall the ease of use of NetDynamics and the technical capabilities of Kiva. J2EE? Another interesting idea, but again, not for real projects.

Where are these products now? They've been almost entirely replaced – by solutions based on those "interesting" ideas of SQL and J2EE. It has become clear that organizations felt standards-based architectures were more important than proprietary solutions.

Now with real synergy starting to happen between J2EE and Web services standards, particularly the latest addition of business process Web services standards, a similar story is poised to unfold within the integration space. The proprietary black box of integration appears ready to be cracked open.

Checkboxes Don't Count

So far, the integration space has proved relatively immune to the standardization process. Yes, many vendors claim checkbox compliance to a laundry list of standards, but it's often just that: a checkbox. It's like putting a relational driver on top of a network database: it works, but is it really relational?

It's getting harder to play these kinds of games. Many organizations are integration veterans and they understand the value of integration, whether it's increased efficiencies, lowered cost structures, or better visibility into internal and external processes. They have deep insight into the integration problem domain and how it can and should be standardized.

Juxtaposed against this increased level of awareness is a set of interrelated standards that tackle a broad swath of the integration space. In the runtime arena, J2EE provides key integration technologies in every J2EE compliant application server. In the Web services space, an emerging standards stack covering quality of service, business processes, interoperability and management is targeted right at the core of many integration solutions.

Types of Integration

Let's look at how the integration space is organized; Figure 1 offers a simplified illustration.

Each integration type has specific technologies and corresponding standards that map to the problem domain.

Data integration, as its name implies, focuses on moving data between systems often at the database level, sometimes independent of application APIs. Technologies range from simple but proven file transfers to more sophisticated database replication solutions to advanced message brokers. Data integration standards vary from database ETL technologies, messaging infrastructure provided by J2EE containers to Web services facades on top of message brokers. This last category is sometimes re-branded using the latest catchphrase in this space, Enterprise Services Bus.

Moving up the stack, functional integration focuses on using business-level interfaces as the integration point. Almost every business application vendor, including Oracle, Siebel, PeopleSoft, and SAP, has a set of well-defined business APIs for core business functions. This has created a market of adapter vendors who wrap those APIs using J2EE standards such as the Java Connector Architecture and offer them through various integration servers. More recently, a number of application vendors have repackaged their business APIs as Web services.

Next in the hierarchy is process integration, which focuses on the flow or lifecycle of a business transaction. Typical technologies used in this space include human workflow and process engines. Standards have tackled process integration over the years, with the most recent industry consensus occurring around Web services orchestration standards like Business Process Execution Language (BPEL).

At the top of the stack sits process-to-process collaboration. This, too, is process-level integration but at a higher level, spanning businesses or organizations. A range of technologies – trading partner agreements and contracts, business protocol definitions, and business document definitions – exists to accommodate and formalize this increased level of abstraction. Representative standards in this space include industry-specific ones such as RosettaNet and UCCNet, and more generic standards like ebXML. More recently, Web services standards like WS Choreography Description Language (WS-CDL) have brought a Web services flavor to the space.

All of these integration technologies and standards share some common requirements. Every integration server requires transformation tooling; translation capabilities for technologies such as EDI; analytic capabilities for reporting and monitoring; and a runtime engine to provide a reliable, available, and scalable infrastructure.

Building the Standard Integration Platform

One cannot expect that the emergence of a standard automatically means industry adoption. There are many standards that have withered away without adoption. Further, there are some integration tasks that never will be resolved by standards no matter how abstract or all-encompassing they might be. That is the very nature of integration projects.

Starting with J2EE

Figure 2 illustrates a J2EE 1.4 container and the services it provides. Imagine you're an integration startup planning to offer a new integration solution in light of this platform.

Does it make sense for a startup to build a new container for managing threads, connection pooling, and other scalability characteristics when they are built into every J2EE container? Does it make sense to write a new messaging engine when you get JMS out-of-the-box? Does it make sense to write custom adapters when every container that supports JCA and third-party adapters can simply be plugged in?

Not really. The list goes on and grows in each J2EE release: security, transactions, Web services, deployment and management. These are all standardized and, by the nature of J2EE compatibility, consistent among J2EE vendors.

From a customer perspective there is no value if an integration server builds a new plumbing layer at the level of J2EE. J2EE is a popular, mature platform with large numbers of trained developers and administrators, well-known best practices and broad support across the vendor and consulting community.

Moving to Web Services

With J2EE providing the integration runtime environment, the next level to look at is how to bring two or more distinct systems together. Try as organizations might, not every endpoint in a multisystem environment will be J2EE. Integrating such heterogeneous systems can be addressed by Web services.

Figure 3 provides an abbreviated pictorial of Web service standards and how they can be put together to provide an integrated Web services platform.

You can look at this platform as having two major characteristics:

1. ***Endpoint Normalization:*** Web services provide a set of standards from the message definition through quality of service to service capabilities focused on normalizing the definition of application system interfaces. These standards are defined in an operating system, programming model, and application-independent manner. This provides a base framework for both data and functional integration.
2. ***Process Definition:*** A relatively new addition to the Web services stack is the process level. Web services process languages take advantage of the interoperability provided by a normalized set of Web services endpoints, enabling integrators to tie them together into business processes. This provides a base framework for the highest level of the integration hierarchy – process and process-to-process integration.

These are the foundation pieces in many integration solutions. Most start by normalizing to a consistent model of application interfaces and data and then provide process and human workflow infrastructure to weave the systems together.

A common critique of this Web services stack is that very few endpoints in enterprises today are Web services. The rejoinder is that there is a built-in assumption that, just like for JCA, both application providers and adapter providers will see the clear value proposition and market opportunity in offering this infrastructure. Considering the fast pace in which both types of vendors are announcing Web services features, this seems like a sound conclusion.

If one takes Web services seriously, it becomes hard to imagine building an integration solution that is not based upon them. The value proposition for building a parallel stack, like that for J2EE, is simply not there from a vendor perspective nor does it exist from a customer perspective.

Process Standardization – The Core of Integration

Process definition languages tackle standardizing a key characteristic of many integration solutions. Just like SQL was the language that exploited the relational database, the business process equivalent, BPEL, is the language that exploits Web services.

It is possible, particularly with the broad adoption of XML

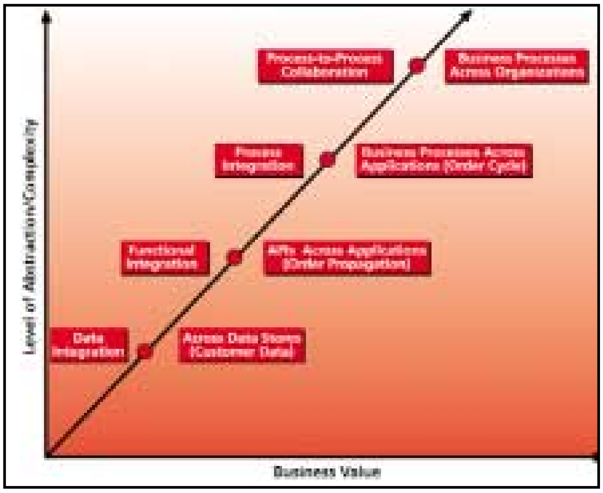


Figure 1 • Integration abstraction vs. business value

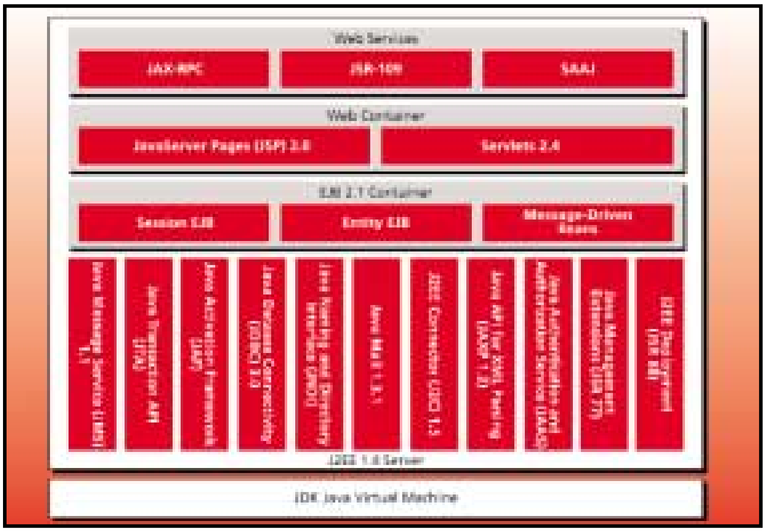

Web Services
JAX-RPC
JSR-109
SAAJ
Web Container
JavaServer Pages (JSP) 2.0
Servlets 2.4
EJB 2.1 Container
Session EJB
Entity EJB
Message-Driven Beans
Java Message Service (JMS) 1.1
Java Transaction API (JTA) 1.0
Java Activation Framework (JAF) 1.0
Java Database Connectivity (JDBC) 3.0
Java Naming and Directory Interface (JNDI)
Java Mail 1.3
J2EE Connector (JCA) 1.5
Java API for XML Parsing (JAXP) 1.2
Java Authentication and Authorization Service (JAAS)
Java Management Extensions (JSR 77)
J2EE Deployment (JSR 88)
J2EE 1.4 Server
JDK Java Virtual Machine

Figure 2 • Typical J2EE 1.4 server and its services

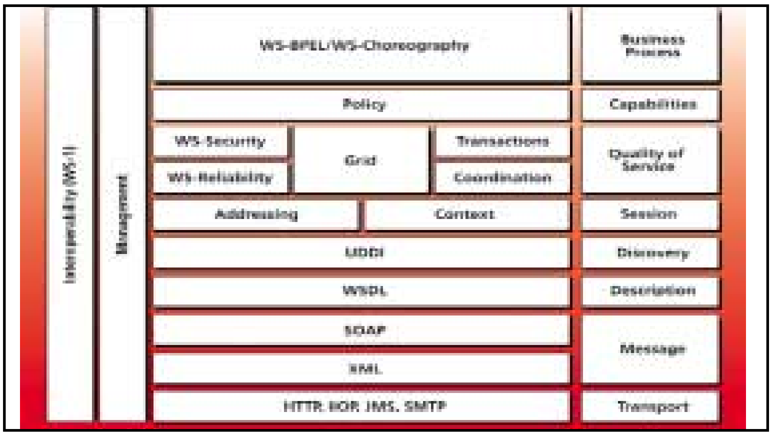


Figure 3 • Web service standards in context

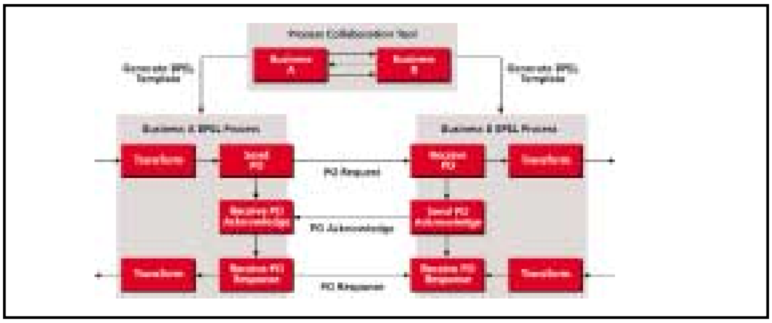


Figure 4 • Tooling a typical purchase order flow

across most integration solutions, to do a peripheral uptake of Web services. It gets progressively more awkward with process languages that depend on key pieces of the underlying Web services standards.

Take for example BPEL, a Web services process automation language that lets integrators weave together a series of services into an executable business process. Participating services are described in Web Services Description Language (WSDL). Data types used in the process are described in XML Schema within WSDL. Correlations between running processes are described in WSDL.

The coupling of the language to WSDL is characteristic of many higher-level Web services standards – WSDL is the way information about the services is expressed. While this could be captured in another way – and often is in existing integration servers – Web-service-centric solutions will use the native Web services mechanism.

The inevitable question comes again: Why invent a new mechanism to describe processes and their characteristics when standards already exist? Why create new concepts – your own black box – when the common language of describing them is universally shared at the Web services layer?

Standards Alone Don't Make a Solution

Of course, standards themselves do not build integration solutions. What they provide is a foundation upon which to build solutions. Process languages are a prime example where there are many solution areas that have to be built for them to be useful: modeling tools, runtime environments, best practices, tuning techniques and many more.

Take for example a typical purchase order scenario where two organizations work hand in hand to fulfill the order. What becomes immediately apparent is that trying to coordinate two independent long running processes has a complexity demanding higher level tools or even higher level process languages.

Figure 4 shows how a business analyst tool could be used to model the collaboration between the two organizations, generating BPEL templates used to implement the processes in each. This scenario, in fact, corresponds to a Web services standard the next level up from BPEL called Web Services Choreography, which is proposed to describe process-to-process collaborations.

At this level of abstraction the playing field for integration servers is suddenly quite different. The competition is not how well you innovate outside the standards; it is how well you innovate within the standards. Organizations do not care who has the best, most flexible process language; they care who makes using the standard process language the easiest and most productive.

Moving Beyond the Black Box

By no means is this integration standardization train near the end of the line. The destination is reasonably evident but there is a significant distance yet to be traveled. Web services have reached into many areas but at the same time have left large parts of integration untouched.

For example, Web services have not tackled transformation in a serious way, although XQuery seems like a logical candidate. They have just begun the journey into business-to-business and process-to-process collaboration standardization with new work happening in choreography. And, Web services have had fits and starts in the area of human workflow and little of note in the area of business rules.

Despite this, the breadth of the standardization problem tackled is significant enough that integration vendors and customers are quickly looking further up the stack for differentiation. Once the foundation has commoditized, as described here, value has to be provided by higher-level capabilities.

Like with any commoditization, base level differentiation will eventually be provided around ease of use, feature/functionality, performance, reliability and the like. Already, one of the key features platform vendors are offering is an integrated J2EE and Web services runtime for their integration solutions.

It is obvious this base level of standardization opens new opportunities for innovative solutions. Just like the emergence of SQL and J2EE created new platform vendors and strong markets for supporting tools, practices and technologies, the combination of J2EE and Web services is now in the midst of doing the same to the proprietary black box of integration.

Author Bio

Mike Lehmann, principal product manager for Web services, Oracle Application Server, has more than 14 years experience in the IT industry. He has worked as a developer, consultant, and project manager in emerging technologies. Mike frequently authors articles, tutorials, and papers on Web services and J2EE and speaks regularly at industry events. He holds a BS in computer science and an MBA from the University of British Columbia.

MIKE.LEHMANN@ORACLE.COM

Talk about possibilities!

ScanSoft®

UNISYS

Brooktrout Technology®
Your Hook into the New Network®

CONVERGYS

aculab

Gold Sponsors

AT&T intel®

VoiceObjects
simplifying voice technologies

NUANCE

Silver Sponsors

GENESYS®

west

NORTEL NETWORKS
BUSINESS WITHOUT BOUNDARIES

Bronze Sponsors

speechTECHNOLOGY MAGAZINE

WebService JOURNAL

wireless BUSINESS&TECHNOLOGY

.NET DEVELOPER'S JOURNAL

Media Sponsors

- >> Learn the latest trends in speech technology!
- >> 77 Educational sessions
- >> 19 Hands on tutorials
- >> Over 100 exhibitors

ACT NOW for Biggest Savings
and Free EXPO Pass with priority code WSJ1.

Registration information:

Chris Nolan

1-859-278-2223 or
Toll free 1-877-993-9767
chris@amcommexpos.com

10TH YEAR!

SpeechTEK 2004
The Voice Solutions Showcase

September 13-16, 2004 >> EXPO 14-15, 2004 >> New York Marriott Marquis

www.speechtek.com for more information

WRITTEN BY **PETER MOSKOVITS**
SUE VICKERS

Easy XML Publishing into Your Enterprise Portal

Sharing the power with non-techies

Enterprise portals provide a single interface to aggregated and componentized information. They significantly reduce the navigational issues inherent with Web sites and make it easier to publish information from disparate sources. The basic building blocks of enterprise portals are portlets, which are reusable, personalized Web components displaying content from various data sources.

One data source, XML, has become a widely used tool in enterprise portals. The work needed to integrate XML with a portal is often considered a highly technical and time-consuming task. For example, to publish XML into a portal, developers usually think that programming is their only option. They go through the entire process of using Beans, compiling XML Schemas, and processing and binding XML instance documents. Developers go through this process without considering the level of complexity needed to publish the content.

This article covers the options available to both page designers and developers to simplify XML publishing. These options range from using portlet building tools that are highly productive and geared toward page designers, to coding using the Java portlet standard Java Specification Request (JSR) 168, which offers developers complete control over the portlet application.

Portlet Building Tools Ease Portal Development

Today, creating portal pages does not require you to be highly technical. It is a natural and important step in the evolution of portals to give nontechnical users, such as page designers, the power of presenting information declaratively from complex sources. The use of wizard-based tools in developing portal pages, running in the browser or an integrated development environment (IDE), is the first step along this path. These graphical tools are built with the page designer in mind and provide fairly simple, but effective publishing mechanisms. These tools often rely on abstracted, reusable data sources, such as XML, Web Services, CSV, or SQL. XML, one of the newest reusable data sources for portals, originally designed for large-scale publishing, plays an important role in integrating data into an enterprise portal.

A few of the most important characteristics and capabilities of portlet building tools include:

- Leveraging existing XML and Web services feeds and rendering directly into the portal
- Accessing XML and Web services feed through parameterized URLs
- Creating reusable portlets with events
- Accessing protected URLs
- Caching portlets
- Supporting filtering and layout formatting
- Accessing XML through the Java Portlet Standard JSR 168

Portlet building tools play an essential role in developing enterprise portals as manually implementing some or all of the functionality in the list would require extensive coding. Also, programming requirements for enterprise portals tend to exclude the page designer from the portlet building process.

Leveraging existing XML feeds and rendering directly into your portal

When integrating XML data or any other type of data source into an enterprise portal, portlet building tools offer choices for page designers. Many companies offer XML feeds for viewing the latest product information or for providing up-to-the minute news feeds. Portals can leverage these XML feeds and integrate the content as a portlet and publish it directly onto a portal page without coding. Often these tools or utilities automatically process the content from the XML feed and offer many different types of layouts to display the content.

The feed from *XML-Journal* shown in Listing 1 is an example that provides the latest news information relating to XML.

Using the wizard, a page designer provides the URL to this RSS feed (www.sys-con.com/story/category.cfm?id=1080&rss=1) and an optional XSL filter to transform the data. The portlet tool consumes the XML, applies any necessary filters, and maps any parameters if necessary (see Figure 1).

Once the XML has been consumed by the tool and the page designer selects a layout appropriate for this feed, the tool creates the portlets,

Author Bios

Peter Moskovits is a principal curriculum developer in the Portals and Hosted Tools group at Oracle Corporation. He joined Oracle in Hungary in 1998 as an Oracle University instructor. Peter has been working with Oracle Application Server Portal since its early WebDB versions. He is an Oracle Certified Professional and holds an MS degree in computer science.

Sue Vickers is a group manager for Oracle Application Server Portal and has been with Oracle since 1998. Her team specializes in declarative and programmatic portlet development as well as application integration. She works closely with the Oracle development team that actively participates on the committees for JSR 168 and WSRP. Sue joined the Portal team from Oracle Support, where she was the WebDB team lead and an award-winning analyst.

which are added to the portal page. To the end user, this news feed is just another portlet on the portal page (see Figure 2).

Accessing XML and Web services feeds

There is often a need to access XML and Web services feeds through parameterized URLs. This need is derived by a requirement to provide end-users with dynamic content or to filter content rendered on a page based on the user currently logged-in. Portal features enable page designers and end-users to drive content through a URL in the form of page parameters or private portlet parameters. The page parameter drives content to all portlets receiving this parameter and private portlet parameters target a specific portlet on a page. The parameter that needs to be passed along with the URL, should often be a portal system variable, such as the name of the portal user.

At other times the parameter is a personalization parameter. Let's suppose the portlet retrieves weather information from the following URL: http://myweatherinfo.com/rss_weather_info.xml. To be able to retrieve personalized weather information, the requested URL must contain a parameter indicating the ZIP or area code: http://myweatherinfo.com/rss_weather_info.xml?zip=94065.

In this scenario the portlet should support user-level personalization, so that users can specify the geographical location in which they are interested. Let's consider another scenario. When the URL of the portal page is invoked, a parameter called zipcode is passed to it indicating that the page should display information relevant to that particular ZIP code. In this case it is not the portal user, but the calling environment, that specifies the value of the parameter and passes it to the portal page: http://myportal.com/page?zipcode=94065 (see Figure 3). The parameter that is passed to the portal page has to be passed further to the portlet. The task of mapping the page parameter, zipcode, to the portlet parameter, zip, needs to be performed by the page designer.

Note: The ability to map page parameters to portlet parameters declaratively is a portal feature. As a result of this, the portlet can contact the data source and pass the required parameter to it: http://myweatherinfo.com/rss_weather_info.xml?zip=94065. As this example illustrates, portlet building tools can empower non-technical page designers to map portal page parameters to portlet parameters, ensuring the seamless information flow.

Creating reusable portlets with events

Making hyperlinks more dynamic than they are in the HTML world helps enterprise portals elevate interactivity and reusability to a higher level. Let's take a news portlet that uses an XML feed as its data source as an example. The portlet contains a hard-coded hyperlink that takes the users to another portal page. Upon realizing how useful your portlet is, you may want to add another instance of it to a portal page. It works fine as long as you don't want to navigate your users from the two portlets to two different portal pages. Different portal vendors provide slightly different solutions to this problem. A typical solution for making portlets interactive and reusable is to use portlet building tools to create events.

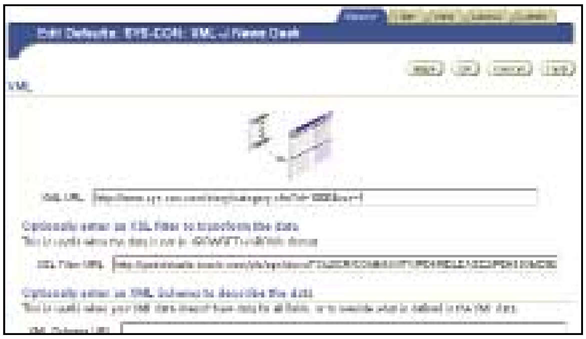
Any user action performed in a portlet, such as a clicking on a hyperlink or a button, may trigger an event in the portal. Page designers then have the ability to map events to actual links: portal pages or physical URLs. When the portal receives the event, it navigates the user to the page as specified by the page designer (see Figure 4).

Access to protected URLs

The XML data source you may want to use in your portlet is often not publicly accessible. The ideal wizard-based tool should provide support for XML data sources that require basic, or form-based, authentication.

Since the same data source could be accessed by a number of portal users, authentication on a per user basis should also be possible. The user dependent authentication information may be stored in the portlet's preference store or in the repository (usual-

~continued on page 57~


XML

XML URL

Optionally enter an XSL filter to transform the data

XSL Filter URL

Optionally enter an XML Schema to describe the data

Figure 1 • Wizard

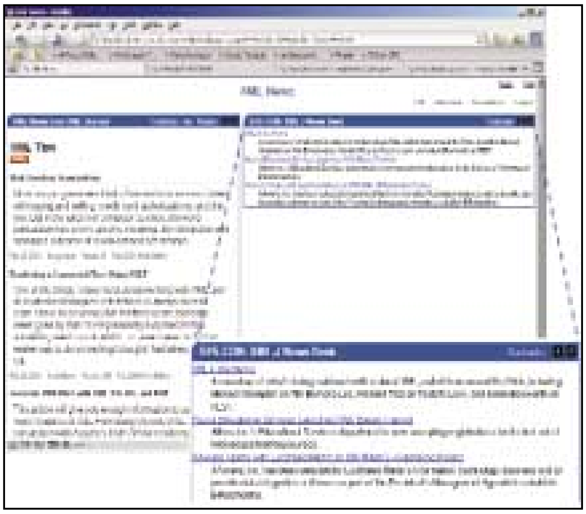


Figure 2 • News Feed

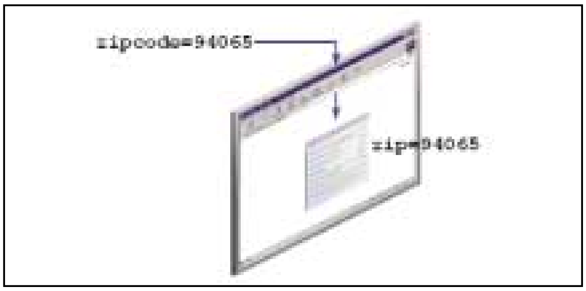


Figure 3 • Seamless information flow

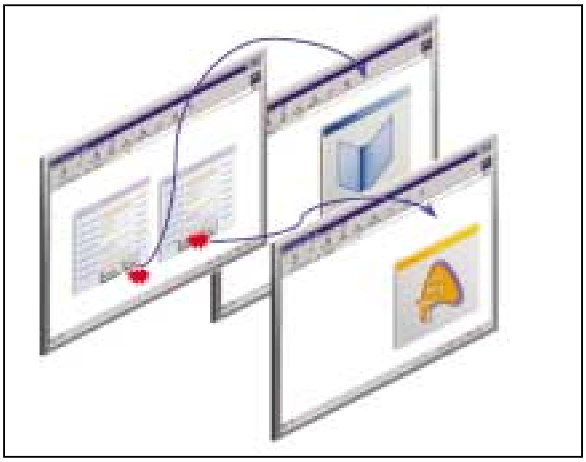


Figure 4 • Navigation

HOME
ENTERPRISE SOLUTIONS
CONTENT MANAGEMENT
DATA MANAGEMENT
XML LABS

WRITTEN BY **MATTHEW ZAGER**

The Evolution of Data-Oriented Services

Systems adapt quickly with XML and Web services

My colleague wrote an article for *XML-J* two years ago about an opportunity we had to solve our data management challenges with XML ("Managing Data Sources with XSLT," www.sys-con.com/story/?storyid=40411). The result of our work was our XML Data Services (XDS), an XML data access language and processing engine, which allowed us to quickly and easily manage the bi-directional transform between data sources and XML. A couple years have passed, and new challenges have arisen and been solved by leveraging our XML data access engine and Web services.

My primary focus over the last couple of years has been enabling legacy data sources and applications to provide visibility into the enterprise and enable business process orchestration. These systems include everything from large-scale Department of Defense databases to automotive engineering applications. In this article I present situations in which applying XML and Web service technologies in a data-oriented manner provided system interoperability and enhanced the ability of systems to more quickly adapt to changing requirements. I also explore the use of Web services as they apply to overcoming barriers within internal development and deployment environments. Finally, I present self-services, a concept to allow clients to get exactly the right information in exactly the right format on-demand. These concepts expose isolated enclaves of information as visible and accessible commodities for the benefit of the enterprise.

Web Service Basics

There are numerous excellent articles and online resources that define service-oriented architectures and Web services in great detail, so I will only briefly review them here to provide context. Web services are adopted for their use of platform-independent open standards enabling the interoperability of distributed disparate systems. Web services can then be deployed either independently or in a service-oriented architecture. Service-oriented architectures are composed similar to traditional distributed software architectures but with much more loosely coupled layers. Figure 1 depicts some typical layers we deploy in a service-oriented system.

Data-Oriented Service Basics

The examples presented in this article focus on the foundation layer in Figure 1, the data access service layer. A recent pattern in service development involves configuring an object-relational mediation framework to instantiate objects encapsulating data contained within a database. The instantiated objects are then marshaled to XML using an object-XML binding API for publication through the service interface. The object model has been relegated to little more than getters and setters for a single database table with no behavior linked to the objects. Our goal is to simplify the building and deployment of data-oriented services by focusing on the fundamentals, namely the source data, the XML Schema defined for publication, and the data access rules.

My first example examines the processes used to Web-enable a legacy Department of Defense database application. The specification called for the application to run in a Web portal environment. To achieve the separation of content from presentation, we published our data as XML using our XDS engine and used XSL transforms to generate our portal-compliant displays. An example use case is finding all personnel located in an area defined by a geographic bounding box expressed as parameters upperLeftLat, upperLeftLon, lowerRightLat, and lowerRightLon. Listing 1 is a snippet of an XDS mapping document implementing this use case (code listings are available at www.sys-con.com/xml/sourcec.cfm). It shows the binding of the results of an SQL query, encapsulated in the xdl: statement, and the XML output identified by the xdl:elements. The details of using the XDS grammar can be viewed in the *XML-Journal* article previously referenced.

This particular application was developed under a joint program for the Department of Defense so it experienced a high degree of visibility. The project was soon approached by developers of another program investigating the feasibility of our portal data being made available in a distributed services environment. The architecture already supported the generation of XML content, so the only thing we needed to provide was the service interface. We were asked to provide an RPC-style service exposing the parameters for the geographic bounding box and return the content as defined by our initial XML Schema. SOAP was the natural selection because of its light-

AUTHOR BIO

Matthew Zager is a senior software engineer with SAIC, an employee-owned research and engineering firm. He has over 6 years experience working on commercial and research projects using object-oriented and XML technologies.

weight XML-based messaging framework able to exchange RPC calls over a variety of underlying protocols. The abridged WSDL can be viewed in Listing 2.

We implemented the SOAP Web service using the Apache Axis toolkit, an implementation of the SOAP submission to the W3C [Axis]. The method shown in Listing 3 shows how we develop Web services using Axis to proxy requests to the XDS engine.

There has been extensive discussion regarding the definition of Web services and what constitutes a valid service. These discussions have given rise to a new model for Web services called Representational State Transfer (REST) Web services. REST Web services focus on the basic underpinnings of the Web in general, namely URIs, HTTP, and XML. In order to be REST compliant we need to define a simple URI to identify our XML resource, which can be accessed via basic HTTP requests. In our case we define a JavaServer Page (JSP) endpoint with a set of query string parameters, e.g., http://host/personnel.jsp?upperLeftLat=Y1&upperLeftLon=X1&lowerRightLat=Y2&lowerRightLon=X2.

The JSP implementation, shown in Listing 4, consists of using the XDS Tag Libraries to reference the mapping document and pass parameters to that mapping document. These simple tag libraries activate the XDS engine, execute the mapping and return the XML results through the JSP response output stream.

Figure 2 shows that by simply providing an Axis wrapper or a JSP endpoint we are able to quickly provide our XML-enabled data in both SOAP- and REST-compliant Web services. The isolation of XML generation and ease of service implementation demonstrates the ability of data-oriented services to adapt to changing requirements. These same principles also apply for new content creation and addition of capability. This is an example of how we successfully deployed our service, but bear in mind that there are a number of data-enabling technologies and data-oriented services do not only apply to databases.

Beyond the Basics of Data-Oriented Services

My second example involves a system that requires a trusted client to access data stored in a database that resides on the opposite side of a firewall between untrusting networks. The current solution to the firewalled database problem requires a client to remotely log into a machine within the trusted network, create a query file identifying the stored procedure to execute, and save this file to their home directory. The database maintainers have a process that runs periodically that will pick up the query file from the client's home directory, execute the specified query, and create a result file. The client will then copy the result file back to their working environment in order to process the results.

There are several problems with this solution, including stale database results, human in the loop, and additional work to handle the proprietary input and output formats. This situation is exactly what web services were designed for. Web services provide on-demand information in a standardized format allowing for machine-to-machine communications.

Based on my first example, it is straightforward to prepare XDS maps for the stored procedures and publish the results as XML in either SOAP or REST services. The clients and their applications are now able to access the stored procedures on-demand and have the results tunneled through the firewall in a standardized format.

The example database I am speaking of is not atypical in that it is quite large, often requiring very complex SQL, and full of special conditions. The demands on the database maintainers are becoming unmanageable, so much so that they are investigating ways to allow the clients ad-hoc query access. The proposed solution to empowering clients in a distributed environment is self-services.

Self-Services

Self-service is a concept in which the service provider exposes a sufficiently flexible interface that allows a knowledgeable and trusted client to control functional aspects of the service. This concept brings Web services from external system interfaces to internal heterogeneous application environments. Candidates for self-service providers include open source content providers with a large, diverse client base or global corporations with distributed IT environments.

While the self-service concept does not apply only to data-oriented services, I'll apply this concept using the technologies and constraints from my previous two examples. Instead of the service provider predefining a set of strict services, a generic service, as shown in Figure 3, is provided where clients are able to send XDS mappings and receive the requested content in the desired XML format. As with all engineering trade-offs, self-services expose more complexity to client in order to achieve the services flexibility.

XDS mappings are defined using an XML grammar, which allows for a seamless fit as the payload of a SOAP envelope. Listing 5 shows how I partition the parameters and mapping.

Once on the service side, the incoming SOAP envelope is parsed to create the parameter HashMap and extract the XDS mapping document. The XML results generated by the mapping are packaged in a SOAP response envelope and sent back to the client. Listing 6 shows the generic envelope parsing and XDS execution.

An HTTP POST of the XDS mapping document to our JSP endpoint implements our REST-style self-service, simple XML in and XML out. Listing 7 demonstrates the use of the XDS Tag Libraries, this time using the REQUEST identifier to specify the location of the mapping docu-

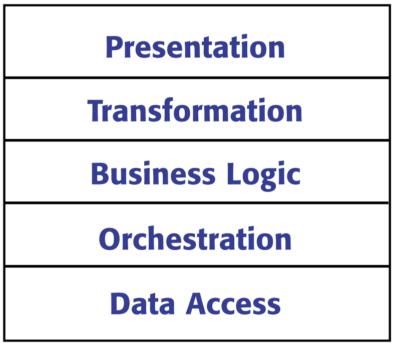


Figure 1 • Service layers

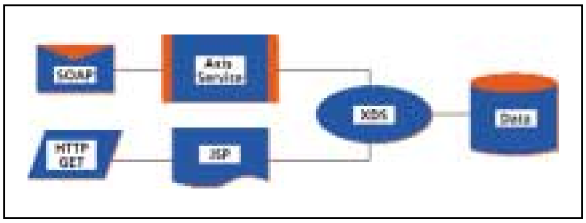


Figure 2 • SOAP and REST services

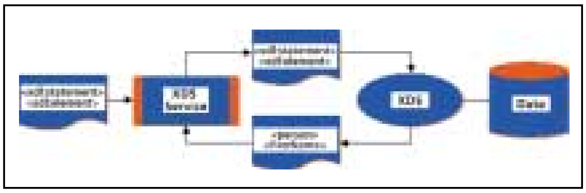


Figure 3 • XDS self-service

ment as the POST body of the HTTP request.

While other self-services may not require the intimate detail a database provider may expose, the key is the client can specify precisely what data they want. The clients do not have to settle for a pre-configured service requiring them to selectively process the result elements. The client is also able to define the response format avoiding costly translation layers.

As with all sensitive services, security standards should be leveraged to authenticate the client and protect the integrity and confidentiality of the messages. In addition to protecting the message itself, we want to protect the resource we are service enabling. This is accomplished by allowing different clients to have different capabilities based on the service provider's level of trust in the client. The higher the level of trust, the more freedom the provider may allow the client to exercise. The levels of trust include not only the right of the client to see specific data, but also the trust in the client that they will not do harm to the provider either intentionally or through ignorance.

Staying with the database example we define three levels of trust. The first-level clients are restricted to using stored procedures with defined parameters. The provider defines a set of stored procedures, the procedure's input parameters and output fields. The second-level clients are permitted to execute the stored procedures as well as execute dynamic queries against a set of database views created by the provider. The views restrict read/write permissions as well as hide the details of the data model. The third-level clients are allowed full access to the database and are able to construct ad-hoc queries. All clients are able to construct an XDS mapping allowing them to set the values of input parameters and define XML mappings for output data fields.

Conclusion

Web services have given developers a new tool enhancing system interoperability and data sharing. Engineering is about tradeoffs of complexity, maintainability, flexibility, and performance among others. The opportunity to work firsthand with many legacy systems has proven the value of data-oriented services and the possibilities self-services hold. I'd be happy to answer questions or hear your thoughts on these topics.

Resources

- Axis: Implementation of the SOAP submission to W3C. The Axis Development Team. http://ws.apache.org/axis/
- Fielding, R. "Architectural Styles and the Design of Network-based Software Architectures," PhD. Dissertation, 2000. www.ics.uci.edu/~fielding/pubs/dissertation/top.htm.
- King, C. "Managing Data Sources with XSLT." XML-Journal, Vol. 3, issue 5. www.sys-con.com/story/?storyid=40411.
- *SOAP Version 1.2*: W3C Recommendation 24 June 2003. www.w3.org/TR/soap12-part1/.
- *Web Services Architecture:* W3C Working Group Note 11 February 2004. www.w3.org/TR/ws-arch/.

ZAGERM@SAIC.COM

Listing 1 • XDS Mapping Document

```
<xdl:querysheet>

    <xdl:datasource ns="personDB"/>

    <xdl:statement ns="getPersons">
      <![CDATA[
      select person.firstname, person.lastname, passport.ppn
      from person, location, passport
      where person.id = passport.person and
            person.location = location.id and
            ((location.x > {$upperLeftLon} and
              location.x < {$lowerRightLon}) and
             (location.y < {$upperLeftLat} and
              location.y > {$lowerRightLat}))
      ]]>
    </xdl:statement>
    <xdl:element name="persons">
      <xdl:transaction datasource="%personDB" exception="stop">
        <xdl:session statement="%getPersons" mode="query">
          <xdl:for-each expr="*">
            <xdl:element name="person">
              <xdl:element name="firstName" expr="*1"/>
              <xdl:element name="lastName" expr="*2"/>
              <xdl:element name="passportNumber" expr="*3"/>
            </xdl:element>
          </xdl:for-each>
        </xdl:session>
      </xdl:transaction>
    </xdl:element>
</xdl:querysheet>
```

Listing 2 • RPC WSDL

```
<wsdl:definitions>
  <types>
    <xsd:schema>
    <xsd:import
namespace="http://schemas.xmlsoap.org/soap/encoding/"/>
    <xsd:complexType name="personResults">
    <xsd:sequence>
          <xsd:element name="persons" type="intf:personList"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="personList">
        <xsd:sequence>
          <xsd:element name="person" type="intf:personType"
minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="personType">
        <xsd:sequence>
            <xsd:element name="firstName" type="xsd:string"/>
            <xsd:element name="lastName" type="xsd:string"/>
            <xsd:element name="passportNumber"
type="xsd:string"/>
        </xsd:sequence>
    </xsd:complexType>
    </xsd:schema>
  </types>
  <wsdl:message name="requestPersonnelResponse">
    <wsdl:part name="requestPersonnelReturn" type=
      "impl:personResults"/>
  </wsdl:message>
  <wsdl:message name="requestPersonnelRequest">
    <wsdl:part name="upperLeftLat" type="xsd:float"/>
    <wsdl:part name="upperLeftLon" type="xsd:float"/>
    <wsdl:part name="lowerRightLat" type="xsd:float"/>
    <wsdl:part name="lowerRightLon" type="xsd:float"/>
  </wsdl:message>
.
.
</wsdl:definitions>
```

Listing 3 • Axis Service Implementation

```
public Document requestPersonnel(float upperLeftLat, float
 upperLeftLon, float lowerRightLat, float lowerRightLon) throws
 XDSException {
    //Load a HashMap with parameters
    HashMap pMap = new HashMap();
    pMap.put ("upperLeftLat", String.valueOf(upperLeftLat));
    pMap.put ("upperLeftLon", String.valueOf(upperLeftLon));
    pMap.put ("lowerRightLat", String.valueOf(lowerRightLat));
    pMap.put ("lowerRightLon", String.valueOf(lowerRightLon));

    //Instantiate XDS, set mode to XML DOM, execute with
       parameters
    XMLDataService xds = new
XMLDataService("requestPersonnel.xds");
    xds.setMode (XDSProcessor.DOM);
    return (Document) xds.get(pMap);
}
```

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

Easy XML Publishing

~continued from page 53~

ly LDAP repository), where the portal user information is being stored.

Caching portlets

The ability to cache XML-based portlets is key to serving portal pages in a timely manner. It is especially true as the slowest portlet usually slows down the entire page. The simplest form of caching is expiry-based caching. Let's suppose you have to display weather or news information in your portlet. Although this type of data changes rapidly, the performance gained by caching the portlet content can result in an approximately 10-minute latency, which is probably acceptable to your users.

The importance of portlet caching increases dramatically when the data source itself (e.g., the provider of the XML feed) has performance issues. As long as the cached portlet content is displayed on the portal page to the end users, the delay that users would experience by accessing the news feed directly is eliminated.

Supporting filtering and layout formatting

Options such as filtering and layout capabilities provide a more productive experience when publishing XML content. Some filtering capabilities include the ability to define conditions, ordering, and limiting the size of the result set.

Wizard-based portlet building tools tend to be limiting on the user interface front. They are expected to offer a wide variety of flexible views (e.g., tabular, chart, tree, scrolling news) and pagination support. Some may also provide APIs to create custom layouts.

Accessing XML through Java using the Portal Standard JSR 168

Occasionally, there is a need to extend the functionality provided by portlet building tools so that developers can have more control over the integration with the application or Web service. Programming provides developers with more control to publish with XML or to use XML to manipulate Web services. Developers can either extend the portlet already created by the wizard-based tool or they can program the portlet from the beginning and have total control over the application. Extending the portlet created by the wizard-based tool offers developers with a more productive solution instead of programming from the start. The page designer can initially create the content and then send additional requirements to the development team to extend the layout of the portlet.

When creating portlets using XML where full control is needed and where the portlets will be rendered on multiple portals, developers can program portlets using the portlet standard, Java Specification Request (JSR) 168.

JSR 168 is a specification that defines a set of APIs to enable interoperability between portlets and portals, addressing the areas of aggregation, personalization, presentation, and security. JSR 168 defines:

- The portlet API (portlet container) provides a runtime environment to invoke portlets. Developers use this API to program portlets that render on compliant portals.
- URL-rewriting mechanism for creating user interaction within a portlet container.
- Security and personalization of portlets.
- A common API for interoperability and portability of Java portlets between portal vendors.
- Access to a variety of XML registries.

Since JSR 168 is a standard for building Java portlets, developers have access to all of the APIs of Java 2 Enterprise Edition (J2EE). Java provides the building blocks for XML publishing, developing Web services, and applications that access Web services. Using JSR 168 and J2EE, developers can write interoperable Java portlets that access and display data in an XML document or parse and transform XML documents. These portlets can be rendered on any portal platform that supports JSR 168.

Some available APIs include:

- Java API for XML Processing (JAXP)
- Java Architecture for XML Binding (JAXB)
- Java API for XML Registries (JAXR)
- Java API for XML-based RPC (Jax-RPC)

Conclusion

As this article has described, XML publishing into a portal can be a fairly easy process. Page designers can use portlet building tools that are very productive and simplify the process of integrating applications and content that could normally take weeks to code. For developers, Java offers a viable solution to XML publishing into a portal with the portal standard, JSR 168. JSR 168 provides all the APIs for publishing XML with Java as well as the APIs to render that Java application as a portlet into the portal.

PETER.MOSKOVITS@ORACLE.COM

SUE.VICKERS@ORACLE.COM

LISTING 1

```
<item rdf:about="http://www.sys-con.com/story/
  ?storyid=43708&amp;rss=1">
   <title>XML in the News</title>
     <link>http://www.sys-
con.com/story/?storyid=43708&amp;rss=1</link>
    <description>A round-up of what's being said and
      written about XML, culled from around the Web,
        including Michael Champion on Tim Berners-Lee,
        Wendell Piez on Postel's Law, and Adam Bosworth on
        REST.</description>
    <dc:date>2004-02-18T00:00+00:00</dc:date>
  </item>
  <item rdf:about="http://www.sys-con.com/story/
   ?storyid=43636&amp;rss=1">
    <title>Altova Educational Services Launches Web-Based
     Training</title>
    <link>http://www.sys-con.com/story/?storyid=43636&amp;
     rss=1</link>
    <description>Altova Inc.'s Educational Services
     department is now accepting registrations for its
      first set of Web-based training courses.</
      description>
    <dc:date>2004-02-11T00:00+00:00</dc:date>
  </item>
  <item rdf:about="http://www.sys-con.com/story/
    ?storyid=43638&amp;rss=1">
    <title>XAware Teams with Lockheed Martin on $98 Million
      E-Rulemaking Project</title>
    <link>http://www.sys-con.com/story/?storyid=43638&amp;
      rss=1</link>
    <description>XAware, Inc. has been selected by Lockheed
     Martin's Information Technology business unit to
      provide data integration software as part of the
President's Management Agenda to establish E-Rulemaking.</
  description>
    <dc:date>2004-02-11T00:00+00:00</dc:date>
  </item>
```

TIBCO Software Extends Business Activity Monitoring with TIBCO OpsFactor

(Palo Alto, CA) – TIBCO Software Inc., an enabler of real-time business and the world's largest independent business integration software company, has announced the availability of TIBCO OpsFactor 1.0. The new product gives users visibility into business operations that are being orchestrated by TIBCO's business integration software through the creation of dashboards that visually display real-time information, operational metrics, and key performance indicators that reflect business goals and conditions.

TIBCO OpsFactor leverages TIBCO BusinessWorks, TIBCO's real-time business integration platform, as the source for process content. By incorporating its own palette of monitoring and measurement tasks into the TIBCO BusinessWorks modeling environment, TIBCO OpsFactor lets users add activity sensors to any step or substep in a process, define threshold levels so exceptional events and conditions can be identified, and configure "rollup rules" so related metrics can be aggregated and assembled into meaningful information and alerts.

www.tibco.com

Infravio X-registry Drives Internal Collaboration, B2B Opportunities

(Cupertino, CA) – Infravio, Inc., has launched the Infravio X-registry, the company's new corporate Web services federated registry, used to stimulate internal collaboration and business-to-business use of Web services.

The Infravio X-registry provides companies with a single point of access for all technical and business information about Web services in their enterprise. It puts equal emphasis on provider and consumer information, leading to control across the services life cycle. X-registry can function as the sole corporate registry, or as a federated registry. When acting as a federated registry, the product provides a consolidated view of all UDDI and ebXML registries within the enterprise.

www.infravio.com.

Reactivity DeliversSolution to Secure SAP, PeopleSoft, and Oracle Web Services

(Belmont, CA) – Reactivity, Inc., which delivers instant and sustainable XML Web services security solutions, has announced that its flagship Reactivity XML Firewall seamlessly and securely deploys the Web services available from SAP, PeopleSoft, and Oracle applications. The Reactivity XML Firewall eliminates the complexity of integrating these enterprise application Web services, reducing deployment times from weeks or months to minutes, while providing a centrally controlled Web services security policy with optimized enforcement.

www.reactivity.com

Systinet Announces New Products

(Los Angeles, and Cambridge, MA) – Systinet, a provider of software for creating service-oriented architectures (SOAs), has announced the immediate availability of Systinet Gateway 1.0, Systinet UDDI Registry 5.0, and Systinet Server for Java 5.0 and C++ 5.0, a family of products that provides the fundamental building blocks for IT organizations to develop and implement SOAs for mission-critical business applications. Systinet's new products are the first to support Web services for multiple messaging-oriented middleware, the latest UDDI Version 3 specification, and new standards like WS-ReliableMessaging, enabling SOAs that are highly interoperable, reliable, and cost effective.

All of the products are immediately available.

www.systinet.com

Westbridge Technology Enhances Solution Security, Administration, Performance

(Mountain View, CA) – Westbridge Technology, a provider of Web services security solutions for service-oriented architectures (SOA), has announced the release of version 3.1 of its flagship XML Message Server (XMS). XMS 3.1 contains the latest security signature and rule updates designed to thwart attacks, XML acceleration enhancements, standards updates as well as improved tools that automate and simplify network management and administration. These capabilities help extend the leadership Westbridge has attained in providing the most comprehensive and advanced Web services security solution that can scale to enterprise-wide SOA.

www.westbridgetech.com

Sarvega Announces Integration of Products with CA Offerings

(Las Vegas) – Sarvega, Inc., a provider of high-performance XML networking solutions, announced today the integration of its XML Guardian Gateway with Web Services Distributed Management (WSDM) from Computer Associates (CA), providing customers with a comprehensive solution for processing, securing, and managing XML-based Web services. Using CA's WSDM 3.1 Observer Development Kit (ODK), integration between the two products will deliver native Web services management capabilities from within the XML Guardian Gateway.

CA's WSDM 3.1 ODK is a comprehensive toolkit that enables third parties such as Sarvega to integrate their products with CA's Web services management solutions. Sarvega's integration with CA's WSDM will allow enterprises to more effectively manage Sarvega's XML Guardian Gateway and Accelerator to deliver end-to-end Web services security and management as well as protect against XML-based content attacks.

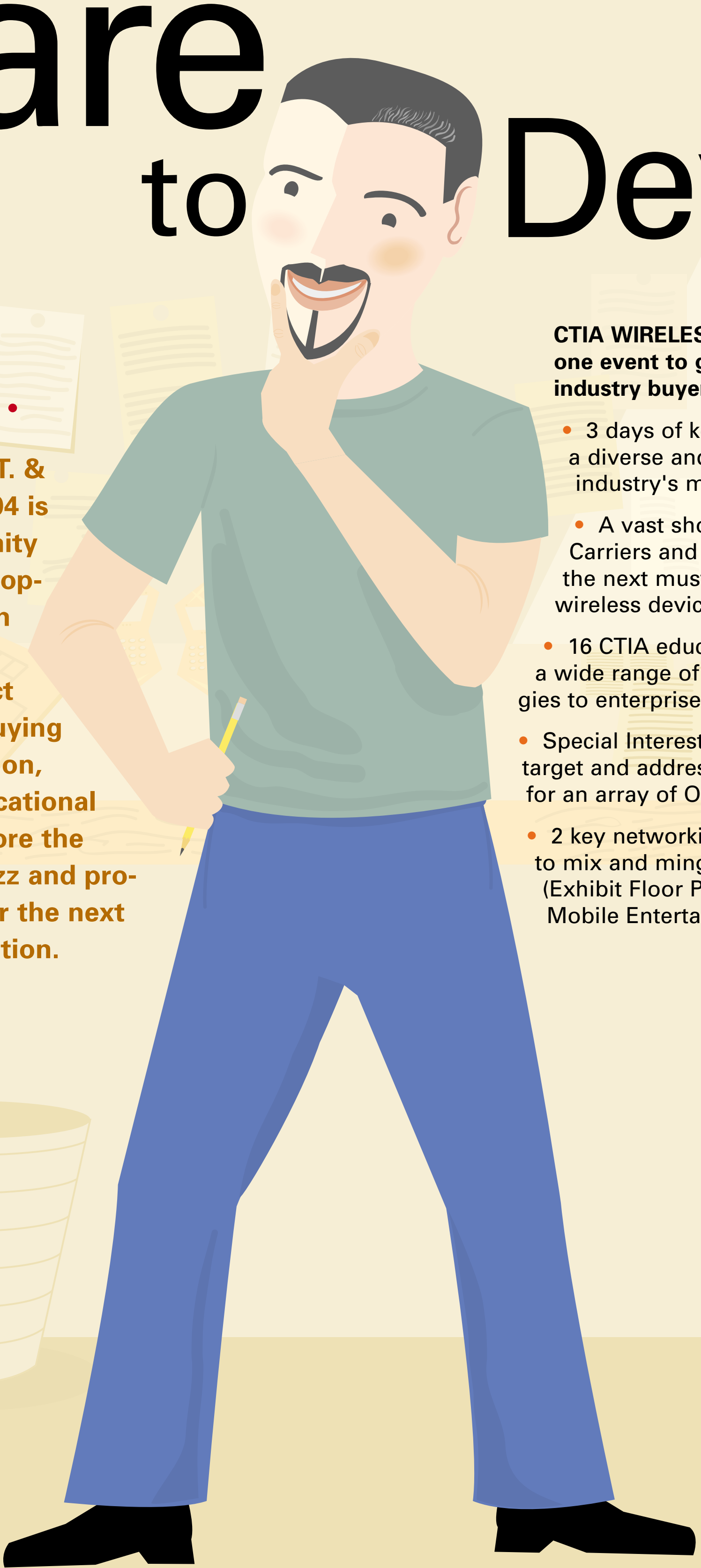
Sarvega's XML Guardian Gateway enables its customers to secure and protect mission-critical XML Web services traffic in a reliable, high-performance appliance solution. Sarvega's XML Speedway Accelerator delivers wire-speed XML processing, including XSLT transformation and schema validation.

www.sarvega.com

Forum Systems Releases Forum XWall 3.0 for Developers; Has WS-I Compliance

(Las Vegas) – Forum Systems, Inc., a provider of trust management and threat protection Web services security solutions, has released version 3.0 of its Web Services Firewall. Forum XWall v3.0 introduces the first implementation of WS-I Basic Profile 1.0 conformance detection and enforcement for both design-time and real-time at the edge of the network. By embedding WS-I conformance into the network instead of the application, Forum centralizes enforcement to ensure the interoperability of multiple Web services as they become externally deployed.

www.forumsys.com


Next Killer App

Open Up!

Portals–Get what
you want easily

company, product and service names maybe trademarks or service marks of others.

Ask the Experts

Prolifics is 1 of only 3 WebSphere Service Providers retained by IBM. Open up with portals! With 26 years of industry expertise, Prolifics' certified WebSphere portal experts deliver a personalized, secure, unified view to your enterprise assets.

Our portal solutions build agility into your business — enabling you to reach new markets, achieve customer satisfaction beyond expectation, and ease collaboration within your enterprise. Our teams of specialists work together with your business experts from gathering the business requirements, translating them to a technical specification, to delivering the final production application.

Portal Executive Assessment

Prolifics experts help you articulate the value of portals to your business sponsors and perform a quick business review identifying portal benefits as it applies to your organization.

Portal Workshops and Proof of Concepts

Prolifics demonstrates valuable portal features and translates the breadth of your portal objectives into a step-by-step roadmap to meet your current and future needs.

Portal Jump Start

Prolifics' consultants work with you from the start to install, configure and jump start your team on using WebSphere Portal Software.

Portal Development and Mentoring

Prolifics' innovative approach to building portals drives down your overall cost and transforms the way you conduct business. We mentor and arm your team with the skills to manage, maintain and extend the portal environment.

Prolifics

Winner of 2003 IBM BUSINESS PARTNER LEADERSHIP AWARD

116 John Street, New York, New York 10038
800.675.5419
solutions4websphere@prolifics.com
www.prolifics.com

Prolifics®

IBM Business Partner