INTERVIEW: DIRECTOR OF SOFTWARE TECHNOLOGY AT IBM RESEARCH PAGE 36

WebSphere® DEVELOPER'S JOURNAL

The World's Leading Independent WebSphere Developer Resource

WebSphereDevelopersJournal.com

SEPTEMBER VOLUME 1 ISSUE 8

DISPLAY UNTIL NOVEMBER 30, 2002
$15.00US $16.00CAN
09>
0 71486 03422 3

SYS-CON MEDIA

EVENT-DRIVEN ENTERPRISE JAVABEANS

Adding the power of asynchronous processing to J2EE application servers

BY AMIT PODDAR PAGE 30

PROLIFICS

WWW.PROLIFICS.COM/WEBSERVICES

WILY TECHNOLOGY

WWW.WILYTECH.COM

WebSphere® DEVELOPER'S JOURNAL

ADVISORY BOARD
Richard Arone, James Martin, Greg Williams

EDITOR-IN-CHIEF	JACK MARTIN
EDITORIAL DIRECTOR	JEREMY GEELAN
EXECUTIVE EDITOR	GAIL SCHULTZ
CONTRIBUTING EDITOR	PATTI MARTIN
SPECIAL PROJECTS EDITOR	RICHARD ARONE
MANAGING EDITOR	CHERYL VAN SISE
SENIOR EDITOR	M'LOU PINKHAM
EDITOR	NANCY VALENTINE
ASSOCIATE EDITORS	JAMIE MATUSOW JEAN CASSIDY
ASSISTANT EDITOR	JENNIFER STILLEY

WRITERS IN THIS ISSUE
Ron Ben-Natan, Andrew Cohen, Alex el-Homsi, Greg Flurry, Jan Hittle, Jack Martin, Jon Maurer, Jim O'Hara, Swaraj Pal, Amit Poddar, David Samuel, Soutik Singha

SUBSCRIPTIONS
For subscriptions and requests for bulk orders, please send your letters to Subscription Department.
SUBSCRIBE@SYS-CON.COM
Cover Price: $15/Issue Domestic: $149/YR (12 Issues)
Canada/Mexico: $169/YR Overseas: $179/YR
(U.S. Banks or Money Orders)
Back issues: $15 U.S. $20 All others

PUBLISHER, PRESIDENT, AND CEO	FUAT A. KIRCAALI
COO/CFO	MARK HARABEDIAN
VP, BUSINESS DEVELOPMENT	GRISHA DAVIDA
SENIOR VP, SALES & MARKETING	CARMEN GONZALEZ
VP, PRODUCTION & DESIGN	JIM MORGAN
VP, SALES & MARKETING	MILES SILVERMAN
VP, EVENTS	CATHY WALTERS
ASSISTANT CONTROLLER	JUDITH CALNAN
ACCOUNTS PAYABLE	JOAN LAROSE
ACCOUNTS RECEIVABLE	KERRI VON ACHEN
ACCOUNTING CLERK	BETTY WHITE
ADVERTISING ACCOUNT MANAGERS	MEGAN RING-MUSSA ROBYN FORMA
ASSOCIATE SALES MANAGERS	CARRIE GEBERT ALISA CATALANO KRISTIN KUHNLE LEAH HITTMAN
CONFERENCE MANAGER	MICHAEL LYNCH
REGIONAL SALES MANAGERS	MICHAEL PESICK RICHARD ANDERSON
ART DIRECTOR	ALEX BOTERO
ASSOCIATE ART DIRECTORS	AARATHI VENKATARAMAN LOUIS CUFFARI CATHRYN BURAK RICHARD SILVERBERG
ASSISTANT ART DIRECTOR	TAMI BEATTY
WEBMASTER	ROBERT DIAMOND
WEB DESIGNERS	STEPHEN KILMURRAY CHRISTOPHER CROCE CATALIN STANCESCU
CONTENT EDITOR	LIN GOETZ
CUSTOMER SERVICE MANAGER	ANTHONY D. SPITZER
CUSTOMER SERVICE REPRESENTATIVE	MARGIE DOWNS
JDJ STORE MANAGER	RACHEL MCGOURAN

EDITORIAL OFFICES
SYS-CON Publications, Inc.
135 Chestnut Ridge Road, Montvale, NJ 07645
Telephone: 201 802-3000 Fax: 201 782-9637
SUBSCRIBE@SYS-CON.COM
WebSphere® Developer's Journal (ISSN# 1535-6914)
is published monthly (12 times a year)
Postmaster send address changes to:
WebSphere Developer's Journal, SYS-CON Publications, Inc.
135 Chestnut Ridge Road, Montvale, NJ 07645

SYS-CON MEDIA

Trust Your Instincts

BY JACK MARTIN

Anyone working in the industry is keenly aware that sales pipelines are collapsing on a wholesale basis for both large and small technology companies. It's a risky gambit and sometimes fatal, but stepping on the gas when others hit the brakes can allow a technology company to overtake its mightiest rivals. Using history as a guide shows the players that performed the best following the 1985 and 1990 technology slumps typically ramped up spending and avoided cutting costs across the board.

Instead, these companies hit the gas while their competitors were slamming on the brakes. They pioneered the next generation of technology while their competitors cut back on research. They staffed up while others were laying off. They made strategic acquisitions and invested while rivals conserved their cash. The result: those that were gutsy turned in impressive gains following the slumps. And not just by outspending timid rivals. Some devised strategies and products tailor-made for tougher economic times and cost-conscious buyers.

Others fed off the misfortunes of rivals by snapping up distressed companies at fire-sale prices or by ramping up advertising and establishing brands while rival products went unheard. Great leaders don't manage for the present, they manage for the future. The conventional wisdom is to pull the wagons into a circle and try to keep the cash up, but technology isn't a conventional business.

The giants can afford to play rough when times are tough. With billions in cash, they are rich enough to invest while weaker rivals are busy surviving. No company has used its financial strength to better advantage this time around than Dell Computer Corp. When the recession began, Dell did a sharp about-face and dropped its prices to become the low-priced PC brand. The goal: to grab market share while others struggled.

It worked. Dell dropped its average prices by 17% last year, gaining 2.5 share points. What made it all possible was that Dell started off the recession with the lowest costs in the industry and $5.4 billion in cash, so it could invest in technology and plant upgrades to improve its already efficient supply chain. It was the only PC company that could slash prices and still make money.

Still, even the wealthiest players can't spend like a drunken sailor in a downturn. The winners of the past cleaned house at the first sign of a tech wreck. They quickly brought costs in line with lower demand and overhauled their operations to make them more efficient. Then they blazed new paths.

This downturn is the worst the industry has endured. It's broader than ever before, slamming not only computers, software, and semiconductors but also telecommunications. It's longer and deeper, too. U.S. corporate tech spending has plunged nearly 17% from its peak in the fourth quarter of 2000, according to the U.S. Commerce Department, versus less than 5% drops in the two previous slumps. And now the downturn is dragging through its sixth quarter, exceeding the past two. Goldman, Sachs & Co. doesn't expect a return to 2000 tech spending levels until 2004. A few companies will pay the ultimate price and go out of business. You need to get your expectations and expenses down to where there is clear and present demand today.

Downturns can provide the perfect conditions for changing the rules of the game. Lower demand and depressed prices can alter market dynamics overnight. Market leaders can be caught off-balance and companies that understand the new cost-conscious customer better than their competitors can also turn lemons into lemonade.

To get an even quicker payday, some companies are snatching undervalued businesses. Many tech stocks have dropped 50% or more during the past two years. Those who acquire now, when there is little competition from other buyers, can come away with bargains.

In the dark days of this technology slump, it's worth remembering the lessons of 1985 and 1990. For those with the money and the nerve, this is a rare opportunity to race ahead of the competition. Trust your instincts and don't slow down.

ABOUT THE AUTHOR... Jack Martin, editor-in-chief of ***WebSphere Developer's Journal***, is cofounder and CEO of Simplex Knowledge Company, an Internet software boutique specializing in WebSphere development. Simplex developed the first remote video transmission system designed specifically for childcare centers, which received worldwide media attention, and the world's first diagnostic-quality ultrasound broadcast system. E-MAIL... jack@sys-con.com

Personalize Your Web Site with WebSphere Personalization

BY SWARAJ PAL AND SOUTIK SINGHA

Deliver the ultimate user experience

ABOUT THE AUTHOR

Swaraj Pal is a senior WebSphere consultant at Noospherics Technologies, Inc. He has seven years of IT experience and has worked with WebSphere for several years. Swaraj writes articles on the WebSphere suite of products for various magazines and white papers on WebSphere performance and tuning using the Mercury suite of products.

E-MAIL
swarajpal@yahoo.com

In today's world it's important to capture information about the behaviors of your Web site's visitors. Users reveal some behaviors while registering and completing transactions; others are revealed when you look at what it is about your Web site that interests users. Content delivery based on user preference is the perfect way to satisfy your customers. Personalization techniques fulfill this functionality based on the analysis of stored information on the users' preferences.

Through personalization, you can get closer to your customers. By capturing someone's likes and dislikes, your application can suggest proper document recommendation, promote products, and send e-mail notifications. For example, you can use personalization to suggest products based on their last purchase. This adds business value to large e-commerce–based Web applications that run on WebSphere Application Server. The main benefits are:

- Site visitors get rich personalized information based on user profiles and related business applications.
- User profiles are created based on user sessions and contents.
- Business rules and the collaborative filtering process lead to a personalized experience for registered users.
- Easy-to-use tools are ready for personalized business rules for Web campaigns that can be managed by business managers or content matter experts.

Where will personalization be most valuable within your existing Web site? Here's a list of questions to help you identify the areas where your Web site, and your business, could be improved.

Questions for Managers, Content Coordinators, Site Architects

1. Does your Web application recommend products to customers?
2. Does your Web application support access for authorized use?
3. Does your Web application e-mail special notices to customers?
4. Does your Web application show past purchases and product services?
5. Is there any order status for customer purchases?
6. Can your portal show each employee's self-service?
7. Is there any policy to increase length of visits?
8. Do users feel at home with personalized content?

This is where WebSphere Personalization comes in. This article describes the product and its role in attracting more customers. It also examines in depth some of the basics of existing WebSphere-based J2EE applications to achieve site e-intelligence through WebSphere Personalization techniques.

Personalization Methodology

This article addresses two major phases of WebSphere Personalization – planning and implementation – and covers related issues (see Tables 1 and 2).

Table 1 introduces personalization techniques, including two kinds of rules engines for rule development and implementation. Before we look at the details of both kinds of engines, we need to understand what rules are.

- ***Rules:*** Business-logic statements that act on user and content data.
- ***Rules engine:*** The core component of the Personalization environment, the rules engine caters to the execution environment for the defined rules. Business users set content-matching rules for site visitors.
- ***Rules-based personalization:*** Rules-based personalization is a content-based matching mechanism for the site's visitors. More clearly, content is mapped to the different assigned roles based on business logic. For example, if a site visitor is one of the company's executives, confidential pages are served.
- ***Recommendation-based personalization:*** Recommendation-based personalization stands upon the algorithm of collaborative filtering. This is a data-mining operation from the user session and recommendations are made in real time from sets of dynamic attributes such as clicks, purchase history, preferences, and product matching. This type of personalization needs a recommendation engine such as LikeMinds, which is included with WebSphere Personalization. LikeMinds also allows online market-basket analysis through its Item Affinity engine. In B2B- or B2C-based Web applications, this is immensely helpful in delivering an optimal site experience.

WebSphere and Personalization

The IBM WebSphere Personalization for Multiplatforms link is a good place to find what you need to build your infrastructure. Visit www-3.ibm.com/software/webservers/personalization/doc/v40/PZNINSTL.HTM to learn more about Personalization installation.

Let's start with the prerequisites. For WebSphere Personalization v4.0, the basic requirements are:

- WebSphere Application Server version 4.0 AE
- Database for WebSphere
- Personalization database
- WebSphere Personalization
- Supported Web browser

For product documentation, fixpacks, and readme files, visit www-3.ibm.com/software/webservers/personalization/library.html for detailed information, including the installation procedure.

Actually, WebSphere Personalization installation is straightforward. Figure 1 shows components that can be customized during installation. The Personalization server hosts Web sites for which you define rules and campaigns through the Personalization workspace. The WebSphere Personalization server has the Rules engine, Resource engine, and related APIs as an add-on for WebSphere Application Server. On the other hand, the workspace is nothing but a browser interface to the Personalization server for rule creation and editing, campaign creation, and assignment of rules to content spots created in WebSphere Studio or WebSphere Studio Application Developer (WSAD).

You can specify WebSphere Studio for Personalization for Personalization 4.0 applications using rules and resources. IBM provides a Personalization 4.0 application development plug-in for WebSphere Studio Application Developer. This creates classes for accessing user and content resources from the implicit or explicit profile repository, creating personalization content spots, and assigning those content spots to specific Web pages. It can also publish the content spots and pages to the runtime environment.

Figure 1 depicts the customization screen where you can select options to meet your requirements. We'll mention only some of the key areas for installation here. You must have a supported personalization database because it needs tables and two schemas: EJB and PZNADMIN. Figure 2 shows the settings for the Personalization database during installation, and Figure 3 shows installed Personalization components.

For this article, we used WSAD 4.0.3. WSAD Personalization provides three wizards (see Figure 4):

- Content Spot wizard
- Content wizard
- User wizard

Figure 5 shows the WebSphere Administrative domain with the default server instance where Personalization is installed.

PLAN	ACTOR
1. Area identification for Personalization Solution	Business Managers, Administrators, and Developers
2. Data identification	Same
3. User level demarcation	Same
4. Right personalization technique identification (Rules engine or Recommendation engine or both)	Same

TABLE 1: PLANNING ISSUES

ACTION	ACTOR	SCOPE
Resource implementation	Developers	WebSphere Studio or WSAD with wizards
Web site preparation	Same	Same
Rule development	Business managers	Personalization workspace
Rule implementation and testing	Same	Same
Publishing	Same	Same

TABLE 2: IMPLEMENTATION ISSUES

Personalization Component Tour

The major logical core elements are User Profile, Content Model, Matching Algorithm, Content Repositories, and Feedback (see Table 3).

The explicit repository stores explicit profiling data. In explicit profiling, site visitors are asked to submit their choices. On the other hand, implicit profiling gathers visitors' behavior from the user session and the information is stored in the implicit repository.

Other than core elements, you need a product tour to familiarize yourself with the Workspace and Runtime components – then you can decide how to use them effectively.

Workspace contains the Campaign Manager and the Rule Composer. The Campaign Manager lets you assign rules to the content spot on the Web pages where campaigns are collections of active rules for a Web site within a specified period of time. The Rule Composer helps you create and edit classifiers, actions, and bindings:

- ***Classifiers:*** Classifiers define a set of events or conditions related to site visitors; these are evaluated and, based on evaluation, specific pages are served.
- ***Actions:*** Selection or updating of the content.
- ***Bindings:*** Personalization defines bindings as a way to ascertain that the conditions defined in the classifiers have been met. Based on this, specific actions are executed.

The Personalization Runtime contains the Resource engine and the Rules engine for rules-based personalization, and also the Recommendation engine for collaborative filtering.

ABOUT THE AUTHOR

Soutik Singha is a senior e-business architect at Noospherics Technologies, Inc. He has 10 years of IT experience and has worked with Java and WebSphere for several years. Soutik writes articles on the WebSphere suite of products for various magazines and was a speaker at WebSphere Developers Conference 2002 in Palm Springs.

E-MAIL
soutik@yahoo.com

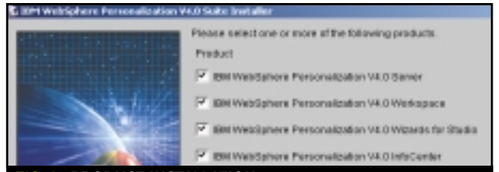

FIG. 1: PRODUCT INSTALLATION

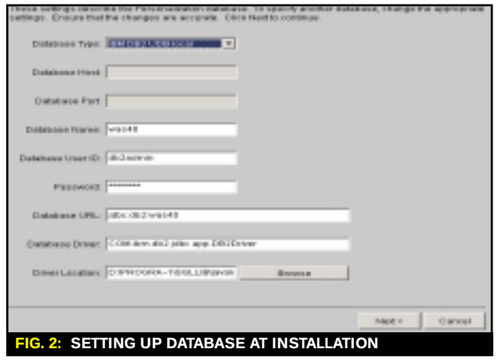

FIG. 2: SETTING UP DATABASE AT INSTALLATION

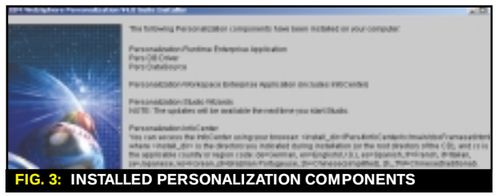
FIG. 3: INSTALLED PERSONALIZATION COMPONENTS

FIG. 4: WSAD MENU BAR

Personalization Development Support

Now the big question: What does the development cycle of WebSphere Personalization look like? WebSphere Personalization–related development can be achieved through VisualAge for Java, WebSphere Studio, and WebSphere Studio Application Developer. With each tool, remember that you need the following:

1. At least one Web application containing at least one JSP to be personalized
2. A valid user resource definition
3. Content definition
4. Content spot identification
5. Publishing and deployment in the WebSphere Application Server
6. Publishing Personalization resources
7. Definition of a business rule
8. Application of the rule to the content spot of the application
9. Publication of the rule
10. Test of the application

I describe briefly the main processes involved in WebSphere Personalization development using WebSphere Studio Application Developer (WSAD). I suggest you use WSAD 4.0.1 with WebSphere Personalization wizards for Studio Application Developer (available for download at ftp:// ftp.software .ibm.com/software/websphere/personalization/support/fixes/fix401/PznStudioDeveloperWizards.exe).

Once the Personalization wizards are installed and configured, you'll see the following three buttons in WSAD:

- ***User wizard:*** For creating user resources for defining attributes of site visitors for Personalization rules.
- ***Content wizard:*** For assisting content resource creation for Personalization rules.
- ***Content Spot wizard:*** Java wrapper beans placed on JSPs or within servlets. During development, you can assign a rule of the appropriate content to be mapped to a specific content spot using the workspace. During runtime, the content spot invokes the assigned rule.

Once you install the WebSphere Personalization wizards for WebSphere Studio Application Developer, you'll be able to build any Web application from scratch using personalization techniques, or convert any existing Web application to be personalization compatible.

The first step is user resource creation, for which you need the User wizard. This helps you create user resources and generates the following files:

1. User Resource (Java)
2. User Resource Manager (Java)
3. BeanInfo (Java)
4. Server Resource Configuration (.hrf)

Use the Content wizard to generate the following files:

1. Content Resource (Java)
2. Content Resource Manager (Java)
3. BeanInfo (Java)
4. Server Resource Configuration (.hrf)

The last step is Content Spot wizard, which defines where business rules will be applied.

The Personalization workspace, which can be invoked by http://<hostname>:<port>/PersWorkspace/ index.jsp, is needed for Campaign Management and Rule Composing, where <hostname>, is your Web server name and <port> is the port on which the Web service is running (see Figure 6).

The events related to Personalization workflow, in the context of WebSphere Application Server, are:

1. A user requests a JSP or servlet in which a personalization content spot bean has been embedded.
2. This request goes to WebSphere Application Server.
3. The JSP 1.x or servlet processor passes the request to the personalization content spot bean.
4. The Personalization context, which accesses resources for rule processing, is initialized.
5. The content spot bean calls the Personalization Rules engine to process the rule, which is an implementation of IBM Business Rule Beans (BRB).
6. The Rules engine processes the rule.
7. The Rules engine returns the result.
8. The JSP scriptlet or servlet uses the rule result to generate a Web page as a response.
9. The Web server serves the page to the user.

Other Areas in Personalization

WORKLOAD MANAGEMENT

Yes, you can do workload management in personalization. The Personalization engine supports applica-

CORE ELEMENTS				
User Profile	**Content Model**	**Matching Algorithm**	**Content Repositories**	**Feedback**
• Site visitors • User attributes • Groups and hierarchies	• Products, article, application • Content attributes • Groups and hierarchies	• User profile, rules, collaborative filtering and market base analysis • Right content delivery	• Explicit or implicit (*See* note below)	• Personalization correctness

TABLE 3: CORE ELEMENTS

IMPORTANT CLASSES AND INTERFACES FROM PERSONALIZATION API	OBJECTIVE
com.ibm.websphere.personalization.resources.Resource com.ibm.websphere.personalization.resources.ResourceDomain2 com.ibm.websphere.personalization.resources.ResourceManager2	• Accessing resources from customer data store
com.ibm.websphere.personalization.resources.BaseResource	• Accessing the Personalization database • Obtaining the dynamic properties that a resource inherited from groups in a resource hierarchy
com.ibm.websphere.personalization.resources.MVResource com.ibm.websphere.personalization.resources.MultiValueUtils	• Accessing the resources that have multivalue properties

TABLE 4: PERSONALIZATION API

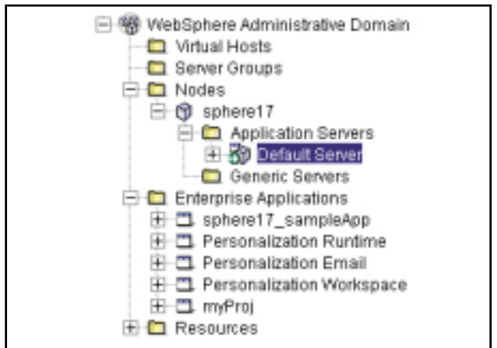

FIG. 5: .WEBSPHERE ADMINISTRATIVE DOMAIN WITH PERSONALIZATION

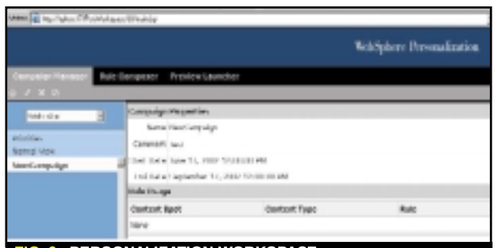

FIG. 6: PERSONALIZATION WORKSPACE

tion server clustering. Here's how. Within the EJB Module of the Personalization Runtime Application there are two Personalization EJBs (BRBeans and PznBeans). Therefore, when an application server containing the Personalization Runtime Enterprise Application is cloned, Personalization Runtime will automatically be workload managed.

PERSONALIZATION APIS

A distinct feature of Personalization is data-modeling support. Depending on requirements, you can define your user models and content models (see Table 4).

Personalization Best Practices

The following are some best practices for WebSphere Personalization:

1. Identify your development tool for personalization.
2. Identify the proper content-management system for your application.
3. Use existing LDAP for user profiles for WebSphere Security implementation.
4. In the requirements phase, try to break down each page element and make sure you've identified the sourced content and profile information for the rules according to your need.
5. Start with a small prototype.
6. Database optimizations are required for at least read access and to take care of joins.
7. Take proper care of business users for rules implementation.
8. Minimize the number of rules per page.
9. Use multiple JSPs when layout changes based on a user's profile are required.
10. Use the MVC design pattern for a personalized site.

Conclusion

Personalization with WebSphere is an easy and lucrative option. The installation is straightforward and the development effort small. In addition, the whole concept works well with the WebSphere Application Server and the WebSphere suite of products.

If you want to provide a superlative user experience by extending the intelligence of your Web application and providing a personal touch, WebSphere Personalization is the way to go. Leverage it and move closer to your customers.

WEB SERVICES: A PRACTICAL APPROACH

BY ANDREW COHEN AND JAN HITTLE

Using a Web services architecture to integrate enterprise resources

Web services, designed primarily for companies to leverage their business services to a global market, also has value and benefits for companies at the enterprise level. Even if you choose to postpone your company's global Web services offerings, the integration and development benefits presented by Web services are worth investigating.

This article explores how to use Web services concepts and technologies to integrate your applications and make resources available to the entire enterprise by discussing these questions:

- What are Web services?
- What value do Web services bring?
- What technologies are used for Web services?
- How can you use Web services at the enterprise level?
- How can you integrate different technologies with Web services?
- What current development issues exist with Web services?

IBM has been a leader in the Web services field by helping to standardize Web service technologies, implementing tools to create Web services in IBM WebSphere, and providing one of the global Web service repositories.

Benefits of Web Services

Many businesses today are a conglomeration of disparate system architectures and technologies. Company mergers and differing departmental philosophies add to this mixture of systems and applications. Companies may even deploy multiple systems that provide overlapping or redundant business services to the organization. Web services can help you integrate your enterprise. By creating an enterprise-wide repository that registers all services, any service is available to any client, which speeds development, enables enterprise-wide access, and opens new revenue streams.

Web services can help to:

- Implement a language- and platform-independent strategy
- Integrate disparate systems and architectures
- Enable access to legacy systems
- Streamline business processes
- Expand business channels
- Build on the componentization and modularization of existing applications

Your company may not be ready to build a Web application based on Web services from an unknown company. In doing so you would be trusting the reliability of another company in a 24/7 marketplace. However, using Web services from a trusted business partner or building a Web portal for departmental exchange of information provides integration benefits to your organization and expedites your business processes. The advantage Web services brings to traditional enterprise application integration is an open standards–based framework to apply to the integration effort. The advantage IBM WebSphere brings to a Web services implementation is a full range of tools for creating and deploying Web services.

ABOUT THE AUTHOR

Andrew Cohen is a senior consultant responsible for delivering distributed WebSphere solutions for global clients. He is a member of the Prolifics WebSphere Consulting Division, a staff of specialized consultants retained by IBM to solve the toughest of their customer's challenges and deliver training, mentoring, and development services.

E-MAIL
acohen@prolifics.com

What Are Web Services?

A Web service is a component packaged and published in a service registry as a single entity that can be used by other programs. Using the Internet, a Web service can be published at the global level, allowing other companies to access it. Alternatively, by using an existing network or intranet a Web service can be accessible to an entire enterprise. Web services can be grouped together to provide greater levels of functionality, exposing a set of business processes, or even a set of applications.

An example of a Web service is a stock quote service that returns a stock price when given a specific ticker symbol. Any business process, such as a document retrieval process or an ordering system, can be engineered with Web services.

In its simplest form, the Web services model is a message-passing system. The breakthrough in Web services is that messages can be passed across different programming languages, different architectures, and different platforms.

Web services use a combination of new and existing technologies that allows any Web service to talk to another Web service. Since existing Web technologies provide the base for Web services, an existing Web server can easily become a Web services server. This, in combination with wide industry support, makes Web services a viable commodity.

The top part of Figure 1 shows Web services allowing Company A to place an order and receive an order confirmation from Company B. The bottom part of the diagram shows internal Web services at Company B being used to fulfill the order – verifying the customer, performing credit card validation through an external provider, and integrating all the legacy systems that control inventory management.

Companies have been solving integration challenges with proprietary solutions for decades. The power of Web services lies in bringing three things to the forefront:

- Best-practice architecture
- Open standards for true interoperability
- Tools to rapidly apply the standards

UNDERSTANDING THE WEB SERVICES ARCHITECTURE

In order to understand how Web services work, let's examine the components of Web services (see Figure 2). Web services are deployed in a service-oriented architecture (SOA) that consists of the following:

- ***Service provider:*** Publishes the availability of its services in a registry and responds to requests to use its services
- ***Service registry:*** Categorizes published service providers and allows searching of its registry
- ***Service requester:*** Uses service registries to find a needed service and binds to that service

Even at the enterprise level, this architecture is in place (although in enterprise configurations all of these roles could be performed by the same company) in order to complete the three Web services operations: publish, find, and bind.

The company acts as service provider, publishing its Web services in a service registry. This registry can be behind the firewall for internal access or at a company portal for business partner access. Once the service is published, application developers, acting as service requesters, can access the registry in order to find the needed service and retrieve the service's interface description. The bind operation occurs as the service requester interacts with the service at runtime, using the binding details in the service description to invoke the service. In this way, valuable business processes can be shared across an organization.

APPLYING WEB SERVICES TECHNOLOGIES TO THE SOA

When you combine the SOA of the service provider, service registry, and service requester with service-oriented technologies, you get the Web services stack shown in Figure 3, with the upper layers of the stack building on the lower layers:

- ***Network:*** At the bottom of the stack are the network layer and the basic communication protocols. A Web service must be available on a network. Communication protocols such as HTTP (Hypertext Transfer Protocol) allow transfer of data and access to applications over the network.
- ***XML messaging:*** XML (eXtensible Markup Language) allows data to be structured in a platform- and language-independent manner. Unlike HTML, which uses a fixed set of tags to specify both the content and display of information, XML allows you to define your own set of tags that accurately describe the information and enable the presentation of the information to change depending on its use. SOAP (Simple Object Access Protocol) specifies an XML-based messaging protocol and provides an envelope for transferring XML data.
- ***Service description:*** WSDL (Web Services Description Language) describes a Web service in an XML-based format.

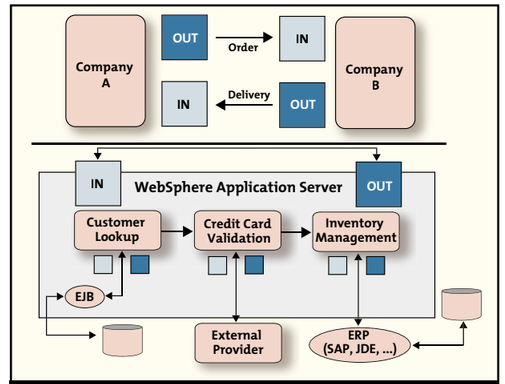

FIG. 1: USING WEB SERVICES WITH BUSINESS PARTNERS AND AT THE PRIVATE LEVEL

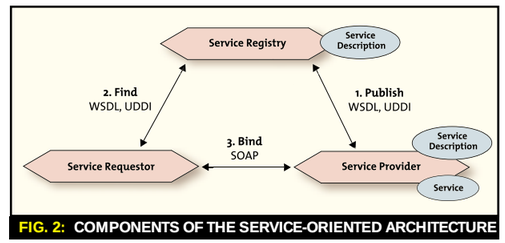

FIG. 2: COMPONENTS OF THE SERVICE-ORIENTED ARCHITECTURE

ABOUT THE AUTHOR

Jan Hittle is a WebSite Publisher and has authored several papers and best practice documents on topics including Application Development, Distributed Systems, Component-based Architectures and WebSphere Solutions.

E-MAIL
jhitt@prolifics.com

- ***Service publication and discovery:*** UDDI (Universal Description, Discovery, and Integration) specifies how to publish and find Web services.
- ***Service flow:*** As the technology matures, Web services will be combined to provide greater degrees of functionality. WSFL (Web Services Flow Language) uses XML to describe how Web services interact to solve a business problem by modeling the basic workflow.

The vertical bars in Figure 3 represent requirements that need to be addressed by all layers of the stack: security, management, quality of service, and transactions.

Developing Web Services in the Enterprise Environment

DEFINING WEB SERVICES

To develop Web services in the enterprise environment, you must first define in your enterprise UDDI which services you want to make available. Any existing EJBs (Enterprise JavaBeans) or COM components, which are inherently modular in nature, are prime candidates for publication in a UDDI registry. In addition, you need to examine your legacy systems. Functionality in those systems can be wrapped as message-based Web services.

PUBLISHING WEB SERVICES

Once the Web services are available, you need to publish them. There are tools available to automate this process, writing the necessary WSDL at the specified location. For IBM WebSphere, the Web Services Toolkit and WebSphere Studio Application Developer (WSAD) contain tools that will write the WSDL. By completing the steps in WSAD's Web service wizard you can:

- Define a Web service based on an existing Java class, EJB, or Document Access Definition eXtension (DADX)
- Generate WSDL and deployment descriptors for the Web service
- Create a service proxy and client mappings
- Create a sample test client, as illustrated in Figure 4

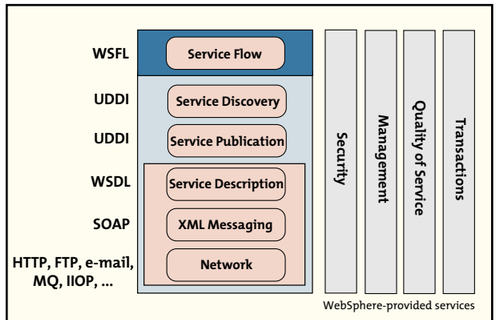

FIG. 3: THE WEB SERVICES STACK

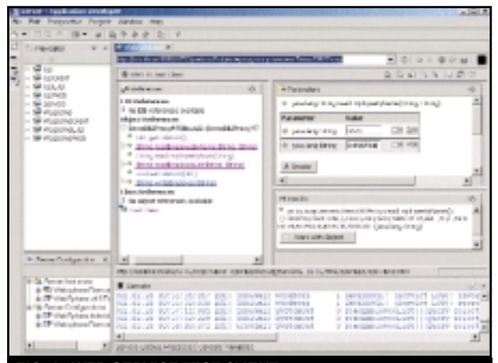

FIG. 4: WEB SERVICE TEST CLIENT

CREATING THE UDDI

By publishing the Web services, you create the UDDI. At the enterprise level, you can implement one or more of the following UDDI scenarios:

- A private UDDI placed inside the firewall for applications within the company
- An enterprise-wide UDDI placed inside the firewall for interactions within the company and with trusted business partners
- A company portal UDDI placed on the firewall, allowing interested business partners to find Web services across the Internet but restricting access to published operations

DEVELOPING SERVICE-BASED APPLICATIONS

Once the services are published they can be called from any enterprise application (see Figure 5). Even though Web services can be bound dynamically, a widely accepted practice is to look up the Web service during development and use its interfaces and methods in the application.

Current Development Issues

As of this writing there are two major divisions of Web services tool sets:

- J2EE technologies used by vendors such as IBM, Sun, and Oracle
- Microsoft's .NET technologies

The J2EE technologies have been around the longest, making their solutions more robust in many areas, but since Web services specifications have been agreed to by all major vendors, different technologies can work together.

There are several issues at the current stage of Web services, especially when implementing Web services at the global level:

- ***Reliable messaging:*** How to determine if a particular message was actually received and was sent only once.
- ***Dependability:*** How to determine the dependability of an unknown company in a 24/7 marketplace.
- ***Security:*** How to make Web services secure if they use the HTTP protocol. HTTPR is a new protocol that has been proposed to address security concerns.
- ***Transactions:*** How to complete transactions in the Web's stateless environment. In the past, transactional applications used a two-phase commit structure to determine when a complex transaction was complete and ready to commit to the database. Since this model is not available on the Web, new models for transactions are being discussed. One of these models uses a concept called *dependency spheres*, which allows both synchronous and asynchronous distributed messages to occur within a single transaction.
- ***Payment for Web services access:*** How to be paid for Web services used by other companies.

As Web services evolve, expect to see answers and specification changes that deal with these issues.

Using Web Services to Integrate Technologies

To illustrate how Web services can be used to integrate disparate system architectures, the sample application described in Figure 6 uses both EJB and COM technologies. The sample application was built with:

- ***IBM Web Services Toolkit*** (a version was also built using WebSphere Studio Application Developer.)
- ***IBM WebSphere 4.0***
- ***Microsoft SOAP Library*** (for the Microsoft clients)

Contact information (name, address, and so on) in the sample application is stored in Microsoft Outlook and a DB2 database. Using Web services, you can update the contact information using either HTML or Visual Basic clients and those changes will be reflected in both data sources.

The sample application consists of:

- ***Proxies:*** These provide directly callable methods that map to SOAP services. They contain the service names and URLs of the Web service servers.
- ***Helper beans:*** Helper beans call the proxies and hold and parse the data returned by the Web services. They contain read and write methods that make the service calls. They also contain get and set methods for data access.
- ***JSPs:*** JSPs contain the user interface and call the helper beans in order to access and change all data. They also perform some data validation.

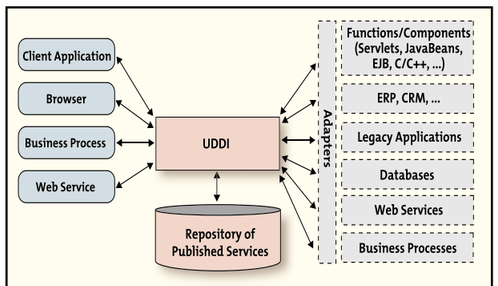

FIG. 5: USE THE UDDI TO ACCESS ANY PUBLISHED SERVICE FROM ANY CLIENT

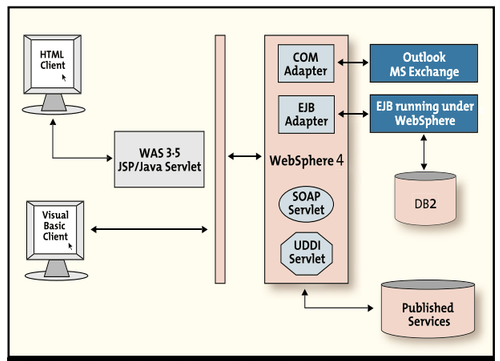

FIG. 6: SAMPLE APPLICATION ARCHITECTURE

- ***SOAP servlets/Apache pluggable service providers:*** These handle all incoming SOAP requests. The requests are parsed and the appropriate implementation is called. The service is defined in a deployment descriptor used when registering the service.

Accessing Web Services

When the sample application was implemented, it contained two SOAP services, one for each data source:

- ***DemoOutlook:*** Provides an interface for reading and writing an entry in the Contacts section of Microsoft Outlook
- ***DemoDB2:*** Provides an interface for reading and writing an Employee record from the Employees table in the DB2 sample database, which is part of IBM WebSphere

USING A JAVA CLIENT

Access to a SOAP server through Java was done using a service proxy. Since the proxy interface is static, it needs to be changed or regenerated if the SOAP service is updated. The Java SOAP client does not access the WSDL file at runtime.

USING A MICROSOFT CLIENT

Access to a SOAP server through VB or VBS was done using the Microsoft SOAP library. Unlike the Java client, the Microsoft client is dynamic. By simply initializing a SOAP object with the WSDL URL, all the service methods are available to be called.

Future Directions

One of the benefits of building an enterprise-based UDDI is that it can be upgraded for other uses. Even though its initial benefit is for internal enterprise integration, with trusted business partners it can later be used to move to global Web services. Major vendors such as IBM and Microsoft provide online access to the global UDDI registry in order to publish services or to search the registry for needed services or potential business partners.

Web services are the future of e-business. Future visions of the Internet see applications primarily being hosted on Web servers, thereby encouraging access by all types of devices, not just personal computers. Web services can be used to provide access to those Internet applications, even allowing syndication of them to many Web sites.

Web services also allow for greater automation of business processes since computer-to-computer communication across systems and architectures is easily implemented. This combination of communication, open standards, and multiple vendor support paves the way for Web services not only to expand, but also to transform e-business.

Resources

- *SOAP:* www.w3.org/TR/SOAP
- *SOAP 1.1 Specification:* http://xml.apache.org/soap
- *UDDI:* www.uddi.org
- *Web services:* www.webservices.org
- *Web services by IBM:* www.ibm.com/webservices
- *Web services on WebSphere:* www.ibm.com/websphere/wow
- *Web services zone:* www.ibm.com/developerWorks/webservices
- *Microsoft .NET:* www.microsoft.com/net

A detailed look

Building DB2-Based Web Services Using WebSphere

PART 2

BY RON BEN-NATAN

In my previous article (*WSDJ*, Vol. 1, issue 7), I gave you a glimpse of the Web Services Object Runtime Framework (WORF), a set of tools for implementing Web services with DB2 and WebSphere. WORF is deployed on WebSphere Application Server (WAS) and uses Apache SOAP 2.2. It implements a layer that runs on WAS and is responsible for taking database access definitions and translating them on-the-fly to Web services constructs supporting SOAP messages and WSDL documents.

ABOUT THE AUTHOR

Ron Ben-Natan, CTO of ViryaNet, Inc., holds a PhD in computer science in the field of distributed computing and has been architecting and developing distributed applications for more than 15 years. Ron's hobby is writing about how technology is used to solve real problems; he has authored numerous books, including *IBM WebSphere Application Server: The Complete Reference* (Osborne/McGraw-Hill).

E-MAIL

rbennata@hotmail.com

The mapping between the database definitions and the Web service is done in a Document Access Definition eXtension (DADX) file. WORF uses the DADX definition to provide an implementation of a Web service through a servlet that accepts a Web service invocation over SOAP, an HTTP GET, or an HTTP POST. This servlet implements the Web service by accessing DB2, invoking the SQL operation defined in the DADX file, and returning the results as a SOAP response.

Last month's focus was on the general framework and the reasons why DB2-based Web services are so attractive. This month I'll continue with a more in-depth discussion of WORF and show you how to build DADX files, deploy them on WAS, and more.

System Requirements and Installation

The system configuration required to implement DB2-based Web services through WORF includes the following:

- DB2 Universal Database Version 7.2
- DB2 XML Extender (only for advanced XML manipulation)
- WebSphere Application Server version 4.0
- Apache SOAP 2.2
- WORF

Assuming you've already installed DB2 and WAS, the rest is simple and should take no more than five minutes. Installing SOAP 2.2 involves downloading the soap-bin-2.2.zip file (see Resources), unzipping it, and copying soap-2_2/lib/soap.jar into {WAS_HOME}/AppServer/lib.

Next, you need to install WORF. After you download worf.zip (see Resources), unzip it and copy worf.jar and soap-samples.jar to {WAS_HOME}/AppServer/lib. You should then install the sample Web application services.war as a new enterprise application within WAS. You can do this using the administrator's console (regardless of whether you're running the single-server edition or the advanced edition). Figure 1 shows the first step of the installation using the single-server administrator's console. Remember, after creating the enterprise application you need to regenerate the Web server plug-in and restart the server. You're now ready to go.

Example Scenario

I'll continue using the example scenario from Part 1, which is taken from the world of call centers. In this example, a call center database stores problem reports and work statuses. For every problem called in by a customer, a CALL is created. The information describing the call is maintained in the CALL table. Once the call is logged, the support center starts working on the problem. Work is encapsulated in an ACTION, which records information such as who is assigned to handle the problem, the scheduled start date, the estimated time of arrival (ETA), and so on. Information pertaining to what work is done in the context of an action is also recorded, and all of this activity is maintained as a record in an ACTION table. The DADX files assume the table structures shown in Table 1.

The operations for which DB2-based Web services are provided are:

- Given an employee ID, return a list of all work assigned to this resource.
- Update the status of an action.
- Given an employee ID, call a stored procedure that computes the yearly first-time fix rate (i.e., how frequently the engineer resolves the problem on his or her first attempt). This complex business function can't be phrased as an SQL operation and is implemented as a stored procedure.

Building DADX Files

At the heart of WORF are the DADX files, which specify how to create a Web service using SQL operations. DADX supports all SQL operations, including queries, updates, and calls to stored procedures.

A DADX file is an XML document. The root node is a DADX node, but the real work is done within the operation nodes. Listing 1 shows a very simple DADX file defining a Web service called getWorkForEmployee, which receives an employee ID and returns a collection of two-tuples, each containing the call number and the action number.

DADX also supports update operations. As an example, Listing 2 shows the DADX file used for the second Web service on our list – a Web service that updates the status of a call. The operation in this DADX file is an update operation that wraps an SQL update statement. Note that in addition to the operation, a documentation element has been added; this will find its way into the WSDL file that's automatically generated by WORF on your behalf. Finally, Listing 3 shows the DADX document for the third Web service wrapping a call to a stored procedure. Incidentally, a DADX file may contain many operation elements, each transformed into a Web service by WORF.

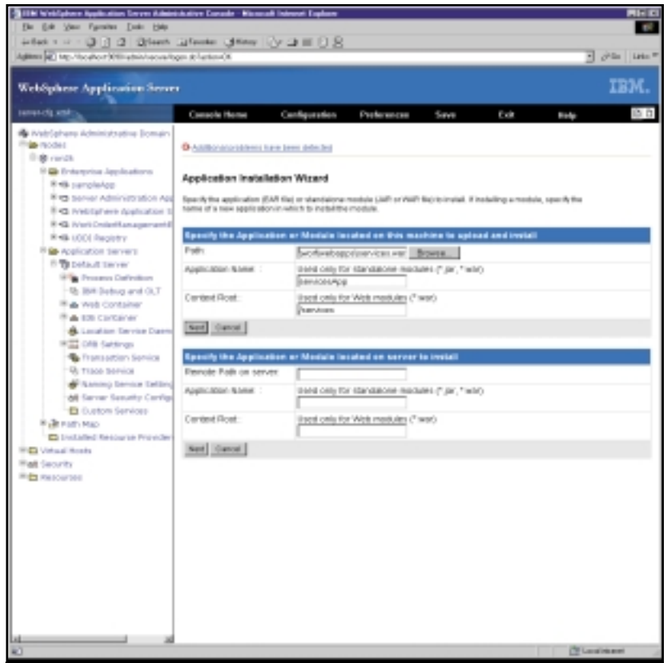

FIG. 1: STEP 1 OF INSTALLATION

CALL	ACTION
CALL__T	CALL__T
BOOK_DATE	ACTION_NO
SYMPTOM__C	WHO_ACTION__C
PRIORITY	ETA_DATE
ACT_RESP_TIME	ACTION_STATUS__C
PROBLEM_DESCRIPTION	
CALL_STATUS__C	

TABLE 1: TABLE STRUCTURES

Deploying and Calling WORF Web Services

You need to do a bit of setup before you can start accessing your DADX-based Web services. Specifically, you need to set up a WORF group and specify the database properties that will be used to access the database. After all, the DADX files define the SQL commands but don't specify the database instance to which the SQL should be applied.

Database properties are specified within a groups.properties file, as shown in Listing 4. This file tells WAS which JDBC driver to load and how to connect to the database. It also defines reload parameters allowing you to modify the DADX files and have WAS reload the definitions automatically. In our example, the groups.properties should be placed in a folder called callWS under {WAS_HOME}/App Server/installedApps/servicesApp.ear/WEB-INF/classes/groups. Your DADX files should also be placed here.

You need to inform the WORF that you've created a new group and the associated Web services. For each group, add a DxxInvoker servlet to your web.xml file within the Web application. Each invoker servlet handles a set of Web services that access the same DB2 instance. Listing 5 shows the additions to the web.xml file required to support the CALL Web services; now you're ready to start using the Web services.

WORF supports both HTTP bindings and a SOAP binding. When invoking a WORF Web service using an HTTP GET binding, use a URL of the form http://<host>/callWS/getWorkForEmployee.dadx/getWorkForEmployee?employee_id=112001. This invokes the first operation shown in Listing 1, passing in the employee ID to be used by the SQL query. The same form of invocation would be used in an HTTP POST request, except that the parameter would be passed within the request body.

To use the SOAP binding, activate a URL of the form http://<host>/ call WS/getWorkForEmployee.dadx/SOAP. The input arguments are passed in as a SOAP request within the body of the request. If all operations are packaged

together in a single DADX file (let's assume all operations are inside the CALL.dadx file), the invocation over HTTP GET would take the form of http://<host>/callWS/ CALL.dadx/getWorkForEmployee?employee_id=112001. Invocation over SOAP would always use a URL of the form http://<host>/callWS/ CALL.dadx/SOAP, while the SOAP request would embody the actual operation that needs to be invoked.

WSDL and Schema Files

WORF is used not only at runtime for wrapping the SQL operation as a Web service in the context of an invocation; it also generates all that's required to deploy your Web service. WORF can automatically generate a Web Services Description Language (WSDL) file, which allows others to learn how to use your Web service. You can publish this WSDL document in a UDDI registry.

For example, to get the WSDL file for the query shown in Listing 1, simply access a URL of the form http://<host>/callWS/getWorkForEmployee.dadx/WSDL. This will return a WSDL document that describes your Web service. If you want to publish your Web service on a UDDI registry, you actually need the WSDL delivered in two files – a service file and a binding file. To retrieve these documents use the following two URLs:

```
http://<host>/callWS/getWorkForEmployee.dadx/WSDLservice

http://<host>/callWS/getWorkForEmployee.dadx/WSDLbinding
```

WORF even takes care of generating schema files for you. Regardless of whether you use advanced XML manipulation with the XML Extender (in which case metadata is often held as DTDs), you can easily retrieve XSD files for your Web service. As an example, to generate an XSD file for the getWorkForEmployee Web service simply access a URL of the form http://<host>/callWS/getWorkForEmployee.dadx/XSD.

LISTING 1: getWorkForEmployee.dadx

```
<?xml version="1.0" encoding="UTF-8"?>
<DADX xmlns="http://schemas.ibm.com/db2/dxx/dadx">
 <operation name="getWorkForEmployee">
  <query>
   <SQL_query>
    select call__t, action_no from action
    where action_status__c != 'A501'
    and who_action__c = :employee_id
   </SQL_query>
   <parameter name="employee_id" type="xsd:string"/>
  </query>
 </operation>
</DADX>
```

LISTING 2: updateCallStatus.dadx

```
<?xml version="1.0" encoding="UTF-8"?>
<DADX xmlns="http://schemas.ibm.com/db2/dxx/dadx"
      xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
 <operation name="updateCallStatus">
  <wsdl:documentation>
   Update the status of a call
  </wsdl:documentation>
  <update>
   <SQL_update>
    update call set call_status__c = :status_c
    where call__t = :call_id;
   </SQL_update>
   <parameter name="call_id" type="xsd:string"/>
   <parameter name="status_c" type="xsd:string"/>
  </update>
 </operation>
</DADX>
```

LISTING 3: computeFirstTimeResolution.dadx

```
<?xml version="1.0" encoding="UTF-8"?>
<DADX xmlns="http://schemas.ibm.com/db2/dxx/dadx"
      xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
 <operation name="computeFirstTimeResolution">
  <wsdl:documentation>
   Call a stored procedure to compute the yearly
   first-time resolution rate
  </wsdl:documentation>
  <call>
   <SQL_call>
call YR_RES_RATE (:emp_id)
   </SQL_call>
   <parameter name="emp_id" type="xsd:string"/>
  </call>
 </operation>
</DADX>
```

LISTING 4: groups.properties

```
# /dadx group properties
dbDriver=com.ibm.db2.jdbc.app.DB2Driver
dbURL=jdbc:db2:callWS
userID=callWS
password=callWS
autoReload=true
reloadIntervalSeconds=5
```

LISTING 5: web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
 <web-app id="WebApp_1">

  ...

  <servlet id="CallWS">
   <servlet-name>CallWS</servlet-name>
   <servlet-class>com.ibm.etools.webservice.rt.dxx.serv
.........let.DxxInvoker</servlet-class>
   <init-param id="InitParam_CallWS">
    <param-name>faultListener</param-name>
    <param-value>org.apache.soap.server.DOMFaultListen
    er</param-value>
   </init-param>
   <load-on-startup>-1</load-on-startup>
  </servlet>

  ...

  <servlet-mapping id="CallWSMapping">
   <servlet-name>CallWS</servlet-name>
   <url-pattern>/CallWS/*</url-pattern>
  </servlet-mapping>

      ...

 </web-app>
```

Advanced XML Manipulation

WORF supports two types of Web services: SQL-based operations and XML-based operations. The DADX documents shown above have all been SQL-based operations. XML-based operations allow you to compose XML documents from relational data and store it. You can create as complex a mapping as is required to allow you to make the most of your relational store without compromising your XML layouts. This uses advanced XML collection operations supported by the DB2 XML Extender, a package that allows you to store and retrieve XML documents from DB2 using flexible metadata definitions. To use these advanced features you must have the DB2 XML Extender option installed, and you need a more thorough understanding of how to use DAD files. This requires a bit of learning; for more details, see the Resources section.

Summary

WORF allows you to Web service–enable your DB2 database and leverage your investment in the database schema and its procedures by autowrapping them with Web services deployed on WAS. You can do this easily by writing a simple DADX file and inserting it into an appropriate groups directory.

Because so many of today's systems are database-centric, many of the Web services you may need to build involve either database operations or storing/retrieving data from your data store. I hope that when you have such a task you remember WORF and the work it can save you. While this two-part series introduced you to the subject, it is by no means a complete overview of WORF, XML Extender, DADX, or DAD files. The resources below provide all you need to build and deploy DB2-based Web services.

Resources

- Download DB2 v7.2 at www14.software.ibm.com/webapp/download/category.jsp?s=c&cat=data.
- Download XML Extender for DB2 at www-3.ibm.com/software/data/db2/extenders/xmlext/downloads.html.
- Download a trial version of WAS 4 at www7b.boulder.ibm.com//wsdd/downloads/#WAS.
- Download Apache SOAP 2.2. at http://xml.apache.org/dist/soap/version-2.2.
- For more information and to download WORF, see www-3.ibm.com/software/data/db2/extenders/xmlext/docs/v72wrk/WORF.html.
- For more information on XML Extender and DAD files, see ftp://ftp.software.ibm.com/ps/products/db2/info/vr7/pdf/letter/db2sxe70.pdf.
- For more details on DADX, download the DADX specification at ftp://ftp.software.ibm.com/ps/products/db2extenders/software/xmlext/docs/v72wrk/webserv/dadx.html.

Truly asynchronous operation

Creating Message-Based Web Services with WebSphere Studio Application Developer

PART 2

BY GREG FLURRY

Part 1 of this article (*WSDJ*, Vol.1, issue 7) showed how to create and use SOAP message–based Web services in WebSphere Studio Application Developer (WSAD). The standard behavior of such services is synchronous. Despite the provision for asynchronous operation of the message-based Web service proxy in Part 1, the operation wasn't actually asynchronous. This article shows how to provide for truly asynchronous operation using threads.

ABOUT THE AUTHOR

Greg Flurry is a member of the IBM Software Group Emerging Technologies area. His responsibilities include introducing Web services technologies to the IBM WebSphere product family.

E-MAIL

flurry@us.ibm.com

Instrumenting the Web Service Client

We'll first instrument the Web service client so we can more easily understand its behavior. Listing 1 shows the Web service modified to introduce a two-second delay between receiving the request and returning the response.

Listing 2 shows the client from Part 1 modified so that the client calculates the time required to execute the proxy's send() and receive() methods. The modified client also prints those times.

If you run the modified client, you'll see something similar to the following results in the Console view:

- The Web service echoed the attribute; the value is "good"
- The proxy send invoked – interval 2273
- The proxy receive invoked – interval 100

The time for the send() method reflects the two-second delay introduced into the Web service. This is because the SOAP Message.send() method used in the proxy send() method called by the client is synchronous.

Creating an Asynchronous Message Client Proxy

To introduce asynchrony, we must introduce a "proxy" thread that makes the synchronous call to the Web service so that the proxy send() method, running in the "client" thread, can operate asynchronously with respect to the Web service. The approach shown in Listing 3 introduces a wrapper for the proxy from Part 1. The wrapper has the same signature as the proxy, so the client doesn't have to change. This approach reuses the proxy from Part 1 unmodified.

First, we create a class that will be used to synchronize the client and proxy threads. The class shown in Listing 3 implements a rather standard pattern for synchronizing threads, using synchronized get() and put() methods.

Listing 4 shows the implementation of the proxy thread. It implements methods needed to set the URL for the endpoint of the Web service, set the input to the Web service, and set the synchronizing object. Of course, it also has a run() method, which calls setURL(), send(), and receive() methods in the original proxy (from Part 1) to invoke the Web service and get the response. It then puts the response in the synchronizing object.

Listing 5 shows the proxy wrapper, which has the same signature as the original proxy in Part 1. The new send() method creates an instance of the synchronizing class and an instance of the new thread object and starts the thread. This causes invocation of the proxy thread's run() method, which in turn calls the Web service. The new receive() method calls the get() method of the synchronizing object to retrieve the response.

Since the revised proxy has the same signature as the proxy in Part 1, we need to make only a small change in the client to use the new threaded proxy. Simply change "Message ServiceProxy" to "ProxyWrapper" and save the revised client. Now run client. You'll see something similar to the following results in the Console view:

- The Web service echoed the attribute; the value is "good"
- The proxy send invoked – interval 300
- The proxy receive invoked – interval 2404

The two-second delay in the Web service is now reflected in the receive() method. This is exactly the asynchronous behavior desired.

Summary

This article walked you through the steps necessary to introduce asynchrony into SOAP message–based Web services. The solution shown isn't the most complete or efficient – exceptions aren't really handled correctly, and it features reuse of both client code and proxy code from Part 1. However, it's a good start that can be customized.

References

- "Creating Web-Based Messages with WebSphere Studio Application Developer – Part 1." *WebSphere Developer's Journal*. Vol.1, issue 7.
- *Web services zone* on IBM developerWorks: www.ibm.com/developerworks/webservices.

LISTING 1

```
package message;

public class MessageService {

    public void process(
        Envelope env,
        SOAPContext requestContext,
        SOAPContext responseContext) {

        Document doc;
        Node bodyNode;
        Node bodyNodeParent = null;

        try {
            bodyNode = (Node) env.getBody().getBodyEntries().elementAt(0);
            Element request = (Element) bodyNode.getFirstChild();

            Node bodyNodeParent = bodyNode.getParentNode();
            bodyNodeParent.removeChild(bodyNode);

            doc = bodyNode.getOwnerDocument();
            Node returnNode = doc.importNode(request, true);
            bodyNodeParent.appendChild(returnNode);

            // introduce some delay
           Thread.sleep(2000);

            OutputFormat format = new OutputFormat(doc);
            StringWriter response = new StringWriter();
            XMLSerializer serial = new XMLSerializer(response, format);
            serial.asDOMSerializer();
            serial.serialize((Element) doc.getDocumentElement());
            responseContext.setRootPart(response.toString(), "text/xml");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

LISTING 2

```
import message.*;
import proxy.soap.message.*;
import org.w3c.dom.*;
import org.apache.xerces.dom.*;

public class TestMes {

    public static void main(String[] args) {

        MessageServiceProxy proxy = new MessageServiceProxy();
        try {
            DocumentImpl doc = new DocumentImpl();
            Element inputEl = doc.createElement("test");
            inputEl.setAttribute("myAtt", "good");
```

```
.........proxy.setURL("http://localhost:8080/
.........MessageService/servlet/messagerouter");
            proxy.send(inputEl);

            long timeOfCall = System.currentTimeMillis();
            proxy.send(inputEl);
            long sendTime = System.currentTimeMillis() -
            timeOfCall;

            timeOfCall = System.currentTimeMillis();
            Element response = proxy.receive();
            long receiveTime = System.currentTimeMillis() -
            timeOfCall;

            String result = response.getAttribute("myAtt");
            System.out.println("The web service echoed the
            attribute; the value is \""+ result + "\"");

            System.out.println("The proxy send invoked --
            interval "+ sendTime);
            System.out.println("The proxy receive invoked -
            - interval "+ receiveTime);

        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }
}
```

LISTING 3

```
package proxy.soap.message;

import org.w3c.dom.*;

public class AsyncResponse {
    private Element response;
    private boolean available = false;

    public synchronized Element get() {
        while (available == false) {
            try {
                wait();
            } catch (InterruptedException e) {
            }
        }
        available = false;
        notifyAll();
        return response;
    }

    public synchronized void put(Element value) {
        while (available == true) {
            try {
                wait();
            } catch (InterruptedException e) {
            }
        }
        response = value;
        available = true;
        notifyAll();
    }
}
```

LISTING 4

```
package proxy.soap.message;

import org.w3c.dom.*;
```

```
public class ProxyThread extends Thread {

    protected String theURL = null;
    protected Element payload = null;
    protected AsyncResponse response = null;
    protected MessageServiceProxy realProxy = new
    MessageServiceProxy();

    public ProxyThread() {}

    public synchronized void setURL(String url) throws
    Exception {
        this.theURL = url;
    }

    public void setPayload(Element payload) {
        this.payload = payload;
    }

    public void setResponse(ProxyResponse response) {
        this.response = response;
    }

    public void run() {
        try {
            realProxy.setURL(theURL);
            realProxy.send(payload);
            Element res = realProxy.receive();
            response.put(res);
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }
}
```

LISTING 5

```
package proxy.soap.message;

import org.w3c.dom.*;
import org.apache.soap.*;

public class ProxyWrapper {

    private AsyncResponse response = new AsyncResponse();
    private String theURL = null;

    public synchronized void setURL(String url) throws
    Exception {
        theURL = url;
    }

    public void send(Element payload) throws SOAPException {
        ProxyThread proxyThread = new ProxyThread();
        try {
            proxyThread.setURL(theURL);
            proxyThread.setPayload(payload);
            proxyThread.setResponse(response);
        } catch (Exception ex) {
            ex.printStackTrace();
        }
        proxyThread.start();
    }

    public Element receive() throws SOAPException {
        return response.get();
    }
}
```

A solution today that plans for tomorrow

United Utilities of the United Kingdom: Field-Force Integration

BY DAVID SAMUEL

Because of the massive utility infrastructures that keep water, gas, and electricity flowing, consumers today can turn on a switch and enjoy the pleasure of well-lit, comfortable living spaces. It's up to Information Technology (IT) infrastructures and integration, however – not generators, pumps, and pipelines – to manage consumers' utilities accounts, including metering and provisioning, and make the delivery system work for producers and consumers alike.

ABOUT THE AUTHOR

David Samuel is general manager of IBM's Global Energy and Utilities industry. He provides leadership to a diverse IBM team that offers solutions at the intersection of business and technology to the energy industry, including regulated energy utilities, and the unregulated companies providing generation, energy wholesaling, and energy services. In addition, he is accountable for the marketing programs and thought leadership that enables the industry business.

FOR MORE INFORMATION

www.ibm.com/industries/utilities

Based in Manchester, England, United Utilities provides metering and billing services for 2.5 million gas, electric, and water consumers in northern England. United Utilities sells its services to numerous utilities providers, and in May 2002 won a major contract with British Gas for meter provision and maintenance – with the help of IT integration.

United Utilities Metering (United Utilities) is staking its success in provision and maintenance on a strong IT solution from IBM and IBM business partner Wheatley Associates that will help to manage and maintain the data it collects and records, as well as use IT to schedule its technicians' visits in a productive and cost-effective manner. United Utilities uses IT to track the types of meters installed (i.e., prepaid or credit, single-rate or double-rate, business or private residential), change the meters when requested, schedule meter repairs, and monitor the results of work assignments.

Managing this data speedily and accurately was crucial to United Utilities. Until recently, the company orchestrated the efforts of its 120-member field force with a scheduling application that required considerable manual intervention. Workers downloaded their daily schedules in the morning and uploaded reports on the day's activities at night. This meant that impromptu reschedulings had to be communicated by phone and then documented in paper reports. Also, because the system grouped schedules by postal code, it could not balance workloads by reassigning tasks to workers in other areas. Consequently, productivity languished.

These problems couldn't have come at a worse time. The deregulation of utilities in the United Kingdom was about to open the company's market up to competition from players nationwide. Undaunted, United Utilities decided to tackle its challenges by seeking a new account-management solution. "We needed a flexible and paperless solution that could transmit and receive data any place our network goes, any time of day, to help our staff be as productive as they could be," said Steven Bradwell, systems support manager of United Utilities Metering.

WebSphere Technology Integration

For help, the company turned to IBM and IBM Business Partner Wheatley Associates, an end-to-end systems supplier in the United Kingdom. Wheatley Associates offered a pervasive computing solution comprised of several parts. The new infrastructure and solution has improved the response time and data accuracy and has minimized errors. The importance of this accuracy is highlighted by the impact of an error in data recording. If a meter serial number or other piece of data is recorded incorrectly, then a repeat visit to that property has to be scheduled in order to regather and reprocess the data. Now, thanks to the new solution, error rates are reduced.

When a job is received from a customer, a booking is triggered in the appointment scheduling system (ASSIGN). ASSIGN then links to the TODAY (meaning today's work assignments) application, which includes a client application running on a Psion netPAD handheld device that uses IBM WebSphere MQ Everyplace to effect the reliable, assured, secure connectivity over the general packet radio service (GPRS) wireless network.

The technician now receives regular updates to job details throughout the day as opposed to batch daily updates. Once a job is completed, the data is entered (meter serial number, configuration, credit balance, final reading for removed meters, etc.), and the information is transmitted back to the server. If there is a discrepancy in the data, the technician can be notified quickly so the data can be verified.

Data accuracy is also improved by the use of bar code scanners which automatically populate the meter serial number data into the application, negating the need for manual entry. The application running on the powerful wireless Psion netPADs downloads all the data concerning that technician's jobs and cross-checks the entered data against the database. All of this data is transmitted to the server using IBM WebSphere MQ Everyplace. In the event of poor GPRS air coverage, MQ Everyplace will queue the data for later transmission and assures that no data is lost or duplicated.

As a result, data errors, processed manually on an exception basis, have been reduced dramatically and the time taken to highlight these (rare) errors reduced from 4–5 days to a couple of hours or less. With its new WebSphere integration systems, United Utilities is enjoying productivity improvements, better customer service, and more accurate information.

"Thanks to IBM WebSphere MQ Everyplace and WebSphere MQ integration," Bradwell says, "we can manage our remote employees efficiently and compete head-to-head with any other metering company on the basis of top-notch service. Prior to our new IT system, which gives us real-time updates through WebSphere MQ Everyplace, we could have lost two or three days before we were able to deal with data discrepancies."

Quality of service soars off the scale. Field service technicians have no trouble staying up-to-date with the latest changes in their work instructions. In the morning, technicians sign on to the TODAY application and log into the JUMBO central database server using IBM WebSphere MQ Everyplace through a GPRS gateway. MQ Everyplace then connects to WebSphere MQ to download the jobs for that day. The jobs are prioritized based upon customer availability. Technicians may leave the GPRS connection open, or work offline if their jobs entail travel beyond the range of the wireless network.

As the day goes on, the technician can connect to the server at any time. The application changes job assignments dynamically, so technicians can simply move from one job to the next without consulting the company for new instructions. "The advantage is that if we have emergency work, or if we have a technician who goes off duty, we can reschedule that work through the remainder of the work-force and not lose time," says Bradwell.

Yet another advantage of JUMBO is accuracy. Once a technician transfers information – such as repair results – to the central database, the database can then validate the information against data that has already been verified. If there are any discrepancies, the system sends a message back to the technician asking for a check. "Now our information is much more accurate, which results in lower administrative costs because there are fewer disputes," commented Bradwell.

A reliable messaging server connects the growing field force. JUMBO compiles the information to be sent and as soon as the technician connects to the system, MQ Everyplace sends the messages to the handheld and receives updates from the technician. WebSphere MQ Everyplace interfaces with TODAY, a proprietary Wheatley Associates Java application, which accesses and updates the third-party database.

Since the data is compressed by WebSphere MQ Everyplace prior to transmission, it takes up only a fraction of the bandwidth available on the GPRS network. As a result, the solution can accommodate 100 concurrent users – with plenty of capacity left over.

This capacity supports United Utilities' competitive efforts, which as a result of winning key contracts such as British Gas have brought an additional 7.5 million customers to the company. "We'll need to staff up our field force proportionately," says Bradwell. "But our Wheatley Associates solution already has the power to serve 10 million customers tomorrow as efficiently as it serves 2.5 million today. And we intend to keep growing in the future, with our rock-solid infrastructure based on WebSphere MQ."

EVENT-DRIVEN ENTERPRISE JAVABEANS

BY AMIT PODDAR

Adding the power of asynchronous processing to J2EE application servers

ABOUT THE AUTHOR
Amit Poddar is senior engineer with a large Internet company. He has 10 years of IT experience, with expertise in the development of networking protocols and server-side Java, and has designed and implemented large-scale fault-tolerance systems using IBM's MQSeries and WebSphere.

E-MAIL
apoddar1971@yahoo.com

An event is an asynchronous notification from an external source, such as a thread or another application. Commands from a user, a timer signaling completion of a specified time period, and the arrival of a message in a queue are examples of events. Two components that are generally involved in event-based processing are the event producer and the event consumer.

The event producer generates events; the event consumer processes them by taking an action. Event-consumer functionality can be divided into two subcomponents: the listener and the consumer. The listener continuously waits for an event to occur. The moment the event occurs, the listener passes it to the consumer for processing.

The event listener's relationship to the consumer can be either one-to-one or one-to-many. The one-to-many relationship is used mostly in scenarios in which the listener listens for multiple types of events and, based on type, dispatches events to the appropriate consumers. Application requirements determine the choice of one model over the other.

Whatever the model, the event-based system allows asynchronous processing, in which the event producer doesn't have to wait for the consumer to finish processing. Listeners can collect events in a queue and process them later. The producer component doesn't have to worry about the availability of the consumer.

Figure 1 shows a basic event-producer/event-consumer model. There are two message producers; one produces messages of Type A and the other of Type B, and both put messages into the message queue Q1. One listener listens for multiple types of messages, and there are separate consumers for each message type. The listener dispatches messages to the appropriate consumers.

EJB and Event-Driven Processing

Invocation of an enterprise bean in EJB 1.1 specifications is limited to synchronous calls from a servlet or standalone Java client. EJB 1.1 standards lack the support for asynchronous invocation of EJBs on an event such as the arrival of a message or a timer alarm. This limits the use of EJB technology for back-end systems where there's no user interaction, and event-based invocation of EJBs is desirable. This takes the power of asynchronous processing away from the J2EE environment. Although EJB 2.0 specifications provide support for message events, support for timer events is missing. In addition, many application servers on the market still aren't fully compliant with EJB 2.0 standards.

In this article, I'll discuss message and timer events in detail, along with how to implement EJBs based on these two event models.

Timer-Driven Bean

Many applications need to do processing at regular intervals. An example is a system that periodically processes orders from a database and sends invoice messages to a queue for printing. This kind of application uses a timer-based processing model.

You can easily implement this in a non-EJB environment by starting a timer thread that generates timer events at regular intervals. Additional threads act as the event listener and the consumer that processes the orders. The only disadvantage is that the database and the message queue operations can't be put in a single transaction because there is no Distributed Transaction Coordinator (DTC). A great feature of J2EE application servers is the support for distributed transactions. It's advisable to use an EJB hosted under an application server to do the job and maintain transactional integrity.

To implement the same functionality in an EJB environment, the most common approach is to write a Java client that acts as the event producer and listener, running outside the application server and invoking an EJB, which consumes the event by taking an action (see Figure 2). Moving the client functionality inside the application server is generally not advised because EJB specifications don't recommend spawning user threads inside an EJB container.

This model would work fine, except that if 50 different consumer EJBs are required, this system will need a client application with 50 threads running outside the application server to act as listeners. You might wonder why the listener can't be a single thread that queues all the events and consumes them one by one by invoking the event-consumer EJB serially. Actually, this can be done, but with the loss of the parallel consumption of the events. This wouldn't be acceptable in most real-world applications. The only practical solution is to have 50 listener threads, but this too has limitations. It adds another chance for failure because each call from the client JVM is a remote call to the JVM hosting the consumer EJBs. Also, the system has to run an extra JVM.

The easiest way to avoid having a client running outside the application server in a separate JVM is to use an EJB as a user thread. Each EJB runs in a separate thread. EJB specifications prevent creation of user threads inside an EJB container. The trick is using an EJB as a user thread. Whatever functionality is provided by the external client thread can easily be moved inside the application sever using an EJB as a thread.

You can easily implement this solution by developing the Timer-Driven Bean (TDB) component. This comprises two EJBs, one acting as the timer event producer and the other acting as the consumer (see Figure 3). In this solution, the external Java client is required only to start and stop the TDB component and doesn't need to be running all the time. All remaining components run inside an application server.

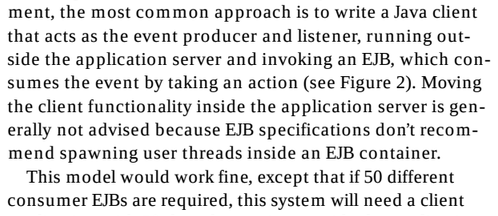

FIG. 1: BASIC EVENT-PRODUCER/EVENT-CONSUMER MODEL

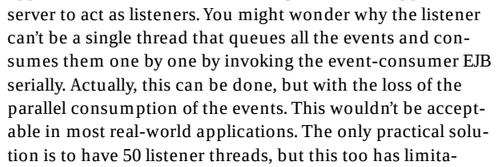

FIG. 2: TIMER EVENT-BASED PROCESSING

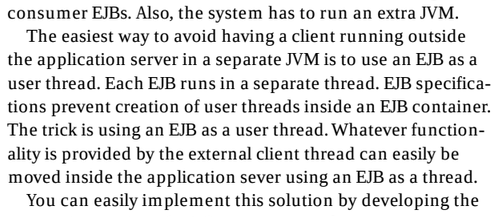

FIG. 3: IMPLEMENTING TIMER-DRIVEN BEAN

TIMER-PRODUCER BEAN

This EJB acts as timer-event producer (see Listing 1). (Listings 1–11 are available for download at www.sys-con.com/websphere/source.cfm.). On a timer event, it invokes the onTimer() method of the timer-consumer bean. After the onTimer() method returns, it sleeps for a specified time period to generate the next event and invokes the onTimer() method on the consumer bean again. The timer-producer bean is a simple component that doesn't contain any business logic. It can be safely deployed as a stateful session bean with the transaction attribute set to TX_NOT_SUPPORTED because it has no interaction with transactional systems.

TIMER-CONSUMER BEAN

This EJB acts as timer-event consumer and contains the actual business logic (see Listing 2). In the example scenario, it would be responsible for processing orders from a database and placing invoice messages in a message queue. Because this bean may interact with transactional systems such as databases and message queues, it should be deployed as a stateful session bean with transaction attributes based on application requirements such as TX_BEAN_MANAGED.

CONTROLLER BEAN

This EJB is required to shut down the timer-driven component gracefully (see Listing 3). It holds the running state of the TDB component. The timer-producer bean checks its state by querying the controller bean after every timer-event handling. If the state is set to shut down, the timer-producer bean exits the while loop gracefully.

It's important to stop the system gracefully – abruptly shutting down the application server may leave some transactions in an in-doubt state. The correct procedure is to shut down all TDB components gracefully before stopping the application server. Listing 3 shows a single static variable controlling the state of all TDB components. If you have different kinds of consumer beans and want to selectively shut them down based on type, you can maintain bean types and their corresponding state in a hashtable with the controller bean. Since the controller bean isn't involved with any transactional systems, it can be deployed as a stateful session bean with the transaction attribute set to TX_NOT_SUPPORTED.

TDB LAUNCHER AND TDB INVOKER

These two classes are required to start the TDB component (see Listings 4 and 5).The TDB Launcher requires two parameters: the number of beans to start and the timer frequency. The TDB Launcher spawns multiple TDB invoker threads. The TDB Invoker creates an instance of the timer-producer bean and invokes the startBean() method on the producer. Each invoker must run as a separate thread since the startBean() method is a blocking call. Because the timer-producer bean goes into an infinite while loop, there's no reason for the invoker to wait for the startbean() method to return, so the TDB launcher exits the JVM.

TDB SHUTDOWN AGENT

This class invokes the shutdown() method on the controller bean, which sets the static variable Controller.running to false (see Listing 6). After processing each event, the timer-producer bean checks the value of Controller.running by invoking the checkRunning() method on the controller bean.

As the variable is a static, the TDB shutdown agent only needs to create a single instance of the controller bean and set this variable to false in order to shut down all the TDBs.

Although the sample listings and description of TDB are based on a separate timer-producer and consumer bean, you can merge the functionality of the consumer and the producer into a single EJB.

Message-Driven Bean

Message-driven technology is most commonly used in areas where the complete processing can be broken down into smaller asynchronous processing units, each interacting with another through message queues.

A typical example is the system shown in Figure 4, in which a timer-based component (such as TDB) picks up orders from Database 1 and puts them in Q1. A message-handler component picks up order messages from Q1, generates invoices, and puts them in Q2. Another message-handler component reads invoice data from Q2, sends confirmation e-mails to customers, and updates order status in Database 2.

You can easily implement a message handler in a non-EJB environment by writing a listener and a consumer. The listener thread waits for messages and on arrival passes them to the consumer for processing. If the processing requires interaction with another transactional resource, such as a database, then the consumer running outside the J2EE environment lacks the transactional integrity provided by a DTC. To achieve transactional integrity, the message consumer should be implemented as an EJB that uses the DTC provided by a J2EE application server.

The most common approach to implementing message-driven bean (MDB) functionality in EJB 1.1-based application servers is to provide an external listener to the message queue and an EJB as a message consumer. The listener waits for a message to arrive in the queue. On message arrival, it invokes the message-consumer EJB, passing on the messageID as a parameter (see Figure 5). The listener doesn't process the message itself, but depends upon the consumer EJB to process it. Disadvantages of this type of implementation are:

- Each message is handled twice: first when it's picked up by the listener for browsing, and then again when it's actually processed and removed from the queue by the message-consumer EJB.
- The message-consumer EJB has to select messages from a queue based on messageID. The selectors put additional load on the messaging system and are generally slow.
- The listener has to make a remote call to the message-consumer EJB for each message. This adds an additional opportunity for failure.
- There is always an extra JVM for message-listener threads running on the machine.

To solve these problems, an MDB component can be developed in a fashion similar to the TDB implementation, where an external client is required only to start and stop the component (see Figure 6). The main difference between the TDB component and the MDB component is that the latter doesn't need an event producer because the events (messages in the queue) can be generated by other applications. A message-consumer bean (MCB) can also act as a message producer, as shown in Figure 4, where the MDB sitting between Q1 and Q2 acts as message consumer as well as message producer. The components are described below.

MESSAGE-INVOKER BEAN

The message-invoker bean (see Listing 7) continuously invokes the processMessage() method of the MCB until the shutdown flag is set to true. The moment the processMessage() method returns, the invoker calls it again. This bean doesn't contain any business logic. Since it doesn't interact with any transactional systems, it can safely be deployed as a stateful session bean with the transaction attribute set to TX_NOT_SUPPORTED.

MESSAGE-CONSUMER BEAN

The MCB (see Listing 8) is an EJB that acts as a message event listener and consumer; it contains the business logic of handling messages. The processMessage() method processes one message every time it's invoked and then returns. This method can also return when the receive (long timeout) method of the queueReceiver object times out because no message arrives in the queue during the specified time period. Use the receive(long timeout) method instead of the receive() method without any wait time specified, because the latter blocks forever and it isn't possible to shut down the system gracefully with no messages in the queue. When the processMessage() method returns, the message-invoker bean first checks the shutdown flag before calling the processMessage() method again. If the acceptable shutdown time for the system is 15 seconds, the timeout for the receive call should be set to 15 seconds. Note that when shutdown is called while the consumer bean is processing a message, the shutdown time will be governed by the message-processing time and not the receive timeout. The MCB interacts with external transactional systems such as databases and should be deployed as a stateful session bean with the transaction attribute based on application requirements such as TX_BEAN_MANAGED.

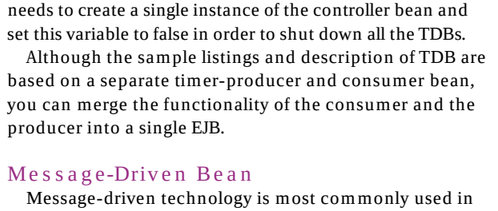

FIG. 4: TYPICAL PROCESS FLOW USING MESSAGE QUEUES AS HANDOVER POINTS

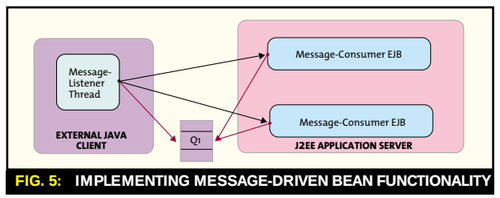

FIG. 5: IMPLEMENTING MESSAGE-DRIVEN BEAN FUNCTIONALITY

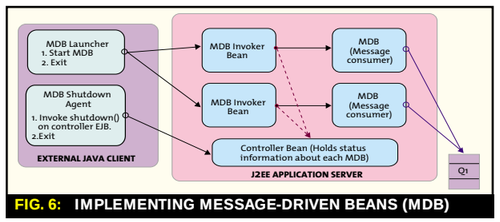

FIG. 6: IMPLEMENTING MESSAGE-DRIVEN BEANS (MDB)

Note: The timer-producer bean sleeps for some time to generate timer events, whereas the message-invoker bean doesn't need to do this. If there are no messages in the queue, the receive(long waitTime) call on the queueReceiver object blocks for the specified time period before returning. This prevents the MDB system from polling the message queues continuously without any delay, which would otherwise result in a high CPU utilization.

CONTROLLER BEAN

The controller bean (see Listing 3) is an EJB that is required to shut down the message-driven component gracefully. It holds the running state of the MDB components, and its functionality is exactly the same as that of the TDB controller bean.

MDB LAUNCHER AND MDB INVOKER

The functionality of MDB Launcher and MDB Invoker (shown in Listings 9 and 10) is exactly the same as that of the TDB Launcher and TDB Invoker.

MDB SHUTDOWN AGENT

The functionality of the MDB shutdown agent (shown in Listing 11) is exactly the same as that of the TDB shutdown agent.

As with the TDB system, it's possible to merge the functionality of the consumer bean and the invoker bean into a single EJB for the MDB system.

Summary

Using an EJB as a user thread, you can develop event-based systems in any EJB application server. TDB and MDB systems are two examples of event-based EJBs. The concept can be extended to more event types. Upgrades and migration to application servers based on EJB 2.0 specifications for MDB functionality can be avoided or delayed because the MDB component achieves the same functionality in EJB 1.1–based application servers.

Additional Features

More features can be added to this system, such as:

- ***Shutdown control:*** This can be specific for each individual TDB/MDB or a group of TDBs/MDBs responsible for a specific task.
- ***Monitoring capabilities:*** The system can be monitored with little enhancements. Each bean can report its statistics, such as the number of events processed, any exceptions encountered, and so on, to the controller bean, which can be displayed by a JSP.
- ***Code:*** The code for the launcher, controller, and shutdown agent can be merged for the TDB and MDB systems.
- ***Timer accuracy:*** If the time to process the timer event is high in the TDB system, the timer frequency will be affected by the event-processing time. To keep the timer accurate, the events can be sent to a separate queue. The queue can be either in memory or on a messaging system. In this case, the timer-producer bean will put a message in the queue instead of calling the onTimerEvent() method of the consumer bean. If an external queuing system is used, the TDB can be used in conjunction with the MDB system.

A Look at the Future

Jack Martin, editor-in-chief of **WebSphere Developer's Journal**, *talked to Daniel Yellin, director of Software Technology at IBM Research and an IBM Distinguished Engineer.*

WSDJ: WHAT'S YOUR LONG-TERM VIEW OF WHERE WEBSPHERE IS AND WHERE IT'S GOING?

DAN: Integration, optimization, and simplification. We're in the midst of a real change in the way people program. Early on, programmers were mostly concerned about functionality in their programs and data management. Today, in contrast, much of programming is about integration – integration of Internet front ends with legacy applications; integration of customer information from diverse databases; integration of business processes across departments. While there have always been large and complex programs, the complexity of building applications today is far greater due to the fact that you have to deal with many heterogeneous platforms and components, and many different layers of software. This makes it harder to build, debug, and optimize applications.

For example, take debugging an application today. When you enter data into a browser and the system crashes, where is the defect? Is it in the browser application? In the business logic running on a mid-tier server? In the messages sent to the back-end legacy application? Or in the database?

WebSphere's success is due to the fact that it understands what it takes to build applications in this environment. At IBM Research, we continue to look to the future and ask "how can we make it even easier to build applications?" Our goal is to dramatically simplify the way we program, debug, optimize, and maintain applications. Our research will continue to drive innovation in the WebSphere platform.

WSDJ: HOW IS WEB SERVICES AFFECTING YOUR WORK?

DY: Some analysts' reports say Web services is the most revolutionary technology since the Internet. I'm a programming languages guy, and I don't think the fundamental concepts behind Web services are all that revolutionary. Concepts such as standardized interface descriptions and online protocols to invoke services have been around for quite some time – RPC, DCE, and CORBA, for example. But Web services have learned from the past. They allow for loose coupling of components and for late binding. Most importantly, they are built on top of Internet standards.

Our IBM Research agenda is twofold. In the shorter term, we are working with our development partners in IBM to make sure that WebSphere continues to be the leading platform for implementing Web services. This involves such issues as optimization, security, and ease-of-use of Web services. We're also building tools to make it easier to exploit legacy applications with Web services. In the longer term, we're exploring whole new paradigms of computation that Web services enable. For instance, how do enterprises optimize a business process by dynamically discovering available services and negotiating their use, and configuring them to implement the business process? Or how do we use Web services together with the computational GRID to enable robust, cheap, and more "elastic" computing? In other words, how can you program an application without concern for where it executes? You focus on the business logic and the logical resources you need. The system automatically figures out where to execute the application and how to get at the resources.

WSDJ: SO USING WEB SERVICES REQUIRES A PHILOSOPHICAL CHANGE IN THOUGHT PROCESSES IN TERMS OF APPLICATIONS YOU'RE BUILDING. CONSTANTLY THINKING ABOUT FUTURE USE OF YOUR APPLICATION.

DY: That's right. In building an application with Web services one needs to think carefully how others will use this application in the future. A key thing that's going to happen is incorporating an industry vertical perspective into Web services. Think of the change this could make, for instance, in the ISV marketplace! Today, an ISV may build a piece of the solution, but it is hard to integrate this piece with the other pieces required for the entire solution. If an industry says there are going to be common ways to interface to different aspects of a business process using Web services, then an ISV can position its piece as part of that process. Its component can use other standardized Web services and offer standardized Web services for other components to use, making the integration of all the components of the solution much easier.

But there will never be just one standard or one way of doing things. So integration will be important, but it will require more semantics; for instance, the ability to know that "address'' and "street, city, country'' mean the same thing. In Research, we're looking at tools to facilitate semantic integration. We're also building a whole new generation of Web services and XML tools. Some of this technology is very exciting, like XForms.

WSDJ: WHAT'S MOST IMPORTANT IN THE WORK YOU'RE DOING AT WATSON?

DY: That's a hard one. We have thousands of researchers in IBM Research, all working on fascinating projects. Personally, I am very excited about research on simplifying the programming process of complex applications and automatically optimizing these applications. There are many facets to this. For example, we're creating a tool that can scan millions of lines of code and find errors in the code; not just typical errors found by compilers, but deeper semantic errors, often due to the way large programs are developed by multiple users. Another facet is Aspect-Oriented Software Development. In this paradigm, different "aspects'' of a program are developed separately, and the computer automatically integrates these aspects based upon the rules specified by the programmer. When you need to evolve the system, you can either modify or add a new aspect.

WSDJ: DO YOU THINK THERE WILL EVER BE A PROGRAMMING LANGUAGE IN PLAIN ENGLISH?

DY: Plain English is hard... I don't know if there's going to be one programming language that's going to do everything. I think it's going to be more like spreadsheets: that is, domain-specific models for specifying computations that are easy to use by a domain expert. Any business person can open a spreadsheet, put in numbers, and get done what needs to get done. There's no need to know about programming. Maybe if the business person is a little sophisticated, he uses scripting, but what he's doing is pretty straightforward. I think for more domains we're going to see things analogous to spreadsheets. We also have a lot of interest in declarative rule-based programming.

WSDJ: IF THIS HAPPENS, THAT HAPPENS?

DY: That's right. Ways of specifying English-like rules. One example is in customization, personalization of applications. A marketing expert may want to say, "if Jack is buying tennis balls, he may also be interested in tennis shorts." WebSphere already supports this functionality. We're doing a lot of research on enhancing that environment, making it simpler for the marketing expert to directly express these rules himself.

Another area for declarative rule-based programming, one in which customers have voiced interest, is when the application needs to change frequently. For instance, in some financial services applications, the business logic gets outdated every few years due to regulatory issues and changes in the marketplace. There is a strong desire to better separate the infrastructure and nonchanging pieces of the application from the volatile business logic. The latter would be coded in rules. We're beginning to see this evolution in the programming model, and we're figuring out how to integrate this model into the WebSphere platform in a natural way.

Yet a third area for rule-based programming is in reactive systems. Imagine a system that monitors all the messages flowing along an MQSeries pipe. It quickly analyzes these messages and, based upon rules, decides if some action should be taken. The rules are based upon time and context. For example, a rule might say, "If Jack is withdrawing more than $1,000, and he made two other withdrawals for more than $1,000 within the last 10 minutes, then notify security.''

There are many interesting research questions here. Rule-based programming, when there are a lot of rules, can get quite complex to reason about. How do we create an ideal environment for building rule-based programs? And how do we best integrate rule-based programming with traditional programs?

WSDJ: WHERE DO YOU SEE WEBSPHERE IN THE NEXT 12 TO 36 MONTHS?

DY: Well, although I'm not in development but in research, I think it's clear that the evolution of Web services will play an important role in the evolution of WebSphere. End-to-end solution development will become increasingly easy. That includes end-to-end optimization. At IBM Research, we have already produced many tools that facilitate optimizing programs. Some of these tools are becoming part of our product line, such as Jinsight (www.research.IBM.com/jinsight), which is now part of the WebSphere Studio.

WSDJ: WHAT IS JINSIGHT?

DY: It's a program visualization tool that allows you to visualize the running of Java programs. Originally many customers downloaded this application from alphaWorks (see www.alphaworks.ibm.com, where many IBM Research technologies can be downloaded), but it is now part of WebSphere Studio. You run an application in test mode and record it. Then you can replay it, visualizing the program execution. You can see your stack frame, objects getting created, what objects call what other objects, when objects are garbage collected, etc. You get multiple views, the ability to look at summary views, and the ability to zoom in on individual method invocations. It's great for debugging and optimizing programs. You can see how much time is spent in different parts of the program. It has great features for helping detect memory leaks.

This research continues to evolve. The newest thing we've done is what we call "drive-by debugging." We're able to attach to a running program and say "give me a little snippet of what's going on.'' This allows you to use visualization on running programs without the overhead of recording the entire program. Another area is debugging distributed programs. Let's say that you have multiple WebSphere servers talking to each other, you have different JVMs, right? Can we correlate the activities in the two JVMs, as one makes a call (say, using RMI) to the other? We can visualize the flow and trace the entire distributed application.

WSDJ: LET'S TALK ABOUT ECLIPSE.

DY: Eclipse is the new basis for IBM tool development. WebSphere Studio is built on top of Eclipse. Eclipse itself is a Java-based platform for building tools.

WSDJ: IBM OWNS ECLIPSE AND PUTS IT OUT TO THE WORLD?

DY: IBM developed Eclipse, and it is open source, www.eclipse.org. IBM put it out to the world for both the Linux and Windows environments, and the open-source community really likes it.

WSDJ: IS JUST IBM USING ECLIPSE?

DY: All IBM tools are being built on top of Eclipse, but so are many ISV tools. Look at eclipse.org and you will see some of the founding members, including companies like Rational Software and TogetherSoft. I believe one of the key problems developers face today is having an integrated way of managing the development process. This requires that all your development tools are able to talk to one another. Eclipse doesn't completely solve this problem, but it is a strong start.

Say you're building an application using JavaScript, servlets, JSPs, EJBs, etc. You need a common way of organizing these artifacts and checking consistency between them, and common layouts in your tools. When looking at a particular artifact, you want the right context-sensitive editor to come up. If you have monitoring tools that do runtime monitoring, then you also want a common repository for storing runtime information and correlating these artifacts. Eclipse enables that.

At IBM Research, we are trying to take this even farther. Our vision is to allow you to move from requirements gathering to design to development to debugging to testing to maintenance, all using common artifacts and integrated tools. Underlying the system should be an intelligent analyzer that helps guide you through the process and that offers suggestions or help when you get stuck. It should facilitate reuse of development artifacts, including requirements and design documents.

WSDJ: IT SOUNDS INCREDIBLE. WHAT ELSE ARE YOU DOING WITH ECLIPSE?

DY: We have lots and lots of projects based upon Eclipse. I've already mentioned an Eclipse tool that helps find errors in source code. We also have a research project called Stellation, which we are making available under eclipse.org, together with Rational Software. Its focus is on how to facilitate large development projects. It does this by providing support for team-based development by using hierarchical branching – this allows better team-oriented change isolation. You have different teams working on different sections of the code base. These teams themselves can organize into subteams, where each subteam gets a specific part of the code to work on. You can continue to subdivide the project as much as you want. Locking and change notification mechanisms and version control are based upon this hierarchy. Stellation also provides better change control by managing versioning at a finer granularity (e.g., versioning of declarations and statements in a module, not just versioning of entire files). Another of its goals is to better facilitate understanding of programs by providing multidimensional views of the program artifacts.

Another Eclipse project at IBM Research is making it easy for nonexpert programmers to build Web-based applications. This means providing a high-level visual model for building applications. It allows a template builder to build templates and end users to customize these templates for their needs. It hides the complexity of J2EE, but still allows you to leverage the WebSphere platform.

We are also working on an Eclipse tool that facilitates the development of applications that can run on many pervasive devices. Often this can be a very time-consuming process; it requires customizing the application for each device, as you need to tailor the look-and-feel for each device. Our programming environment is intended to make it much easier to reuse most of the application development artifacts across devices. It works by having the developer group logical items together and specify the logical flow of the graphical user interface. The system automatically generates a customized user interface for each device. If this generated GUI is not good enough, you can always fine-tune it for the most dominant devices you need to support. But it dramatically reduces the development time of supporting many different form factors.

I should mention that a good part of our research agenda is still focused on Java itself. For example, we have developed a JVM written entirely in Java called the Jikes Research Virtual Machine, or simply Jikes RVM. You can download from the open source section of www.ibm.com/developerworks. It is a great environment to experiment with the Java language and programming environment and execution environment. Many academic institutions are using it for their Java research. Much of our own research in optimizing compilation is done on this platform. We have written some of the leading papers on dynamic and adaptive compilation, which decides how much to optimize a program at run-time, trading off how much is to be gained by the optimization against how much time it takes to compile it.

"Our vision is to allow you to move from requirements gathering to design to development to debugging to testing to maintenance, all using common artifacts and integrated tools."

WSDJ: IF YOU WERE RUNNING A COMPANY THAT HAD WEBSPHERE INSTALLED, WHAT KIND OF NEW SKILLS WOULD YOU WANT YOUR PEOPLE TO BE BUILDING NOW SO THEY WOULD BE EFFECTIVE IN THE WORKPLACE 12 TO 24 MONTHS DOWN THE ROAD?

DY: Web services is certainly one of them. XML skills are extremely important. I would also think WebSphere Portal Server because it is more important than ever to start thinking about the way information is delivered to the end user.

Volume licensing that works

IBM Passport Advantage

BY JON MAURER

Most software vendors include volume licensing programs in their repertoire of products and services. IBM does it, Microsoft does it – and so do others. However, volume licensing programs are not all created equal. Some are seemingly designed to tip the scales strongly in favor of the software vendor while squeezing the customer.

ABOUT THE AUTHOR

Jon Maurer is program director of the IBM Passport Advantage volume licensing program and is responsible for the design and worldwide deployment of the program. Prior to joining IBM, Jon worked at Digital Equipment Corporation on the design and deployment of many worldwide software licensing and business practice programs.

FOR MORE INFORMATION
contact the Passport Advantage Hotline at pahotlin@us.ibm.com

Passport Advantage is IBM's volume licensing program and is designed with the customer in mind. Software vendors should be allowed to profit from a volume licensing program. But the customer should win, too. Unfortunately, that's not always the case. For example, some vendors' programs require their customers to upgrade with every new release of popular software. Does this benefit the customer – or does it just create the equivalent of an annuity-like revenue stream for the vendor? How does it benefit you? Well, it doesn't.

While not all volume licensing programs are created with customers in mind, Passport Advantage is one that was designed to benefit them as much as IBM. Here's how:

- ***More choice:*** Passport Advantage includes all IBM distributed software – WebSphere, Data Management, Lotus, and Tivoli. In addition, you can leverage Passport Advantage across all distributed platforms – Intel, Sun Solaris, HP-UX, Linux, and more. This can increase your return on investment because it applies to so many products, solutions, and platforms.
- ***More opportunities to earn discounts:*** Passport Advantage discounts your volume purchases over time, but you can also obtain even greater discounts based on the size of individual transactions.
- ***Passport Advantage Online:*** This is not just a download service! It's a set of full-featured online tools for Passport Advantage customers and resellers. This allows you and/or your IBM Business Partner to manage your assets online.
- ***Passport Advantage lets you choose how to conduct business with IBM:*** You can do it directly, or you can go through your Business Partner.
- ***What Passport Advantage doesn't make you do:***
  - *Upgrade:* Do it when it works best for you.
  - *Minimum purchase requirements:* None.
  - *License renewals:* Passport Advantage provides a perpetual license, meaning your license remains in effect perpetually – you're not forced "to rent" your software license.
  - *Forecast your purchases:* Passport Advantage doesn't require you to predict the number of licenses you'll be purchasing within a specific time period.

It's About ROI – Yours, Not Theirs

When choosing a software vendor, customers are increasingly basing their decisions on business as well as product-function considerations. This means you're looking for a vendor who is easy to work with, helps in managing software assets, offers lower costs associated with software licensing, and provides software maintenance. In short, you're looking for a reasonable return on your software investment.

You expect to be able to acquire software licenses, and new versions and releases of software incorporating the latest technological advances. These should be backed up by high-quality, responsive technical support – all within a single, flexible, and easy-to-understand program.

Further, you expect to pay a price determined by your total investment in the vendor's products, worldwide and over time, while being able to maintain relationships with Business Partners of your choice. You also know that the rate of change in software development continues to increase, and you want the ability to protect and enhance your investment without continuous incremental expense for frequent upgrades. And you don't want anyone telling you when you have to install upgraded software – that's a business decision for your organization to make.

Are your expectations out of line? Not at all. But you need to shop around. Here are some things to look for:

- ***Does the program increase your options?*** Does it include an entire array of the software vendor's product line or is it limited to just a few products? Does it cover all of your distributed software platforms, or just Intel? Does it let you choose when to install upgrades, or does it force you to upgrade whenever a new release is issued?
- ***What happens if your participation in maintenance lapses?*** Will you have to buy a new license?
- ***What kind of upgrade commitment do you have to make?*** And when do you have to make it?
- ***Is there a required length of time for which you must commit to maintenance?***
- ***Is there a minimum purchase required before you start realizing the benefits of maintenance?***

As you answer these questions and compare volume licensing programs, think about who is getting the better return on investment – you or the software vendor?

What Is Passport Advantage?

Passport Advantage is a simple yet comprehensive program that covers software license acquisition and maintenance options all under a single, common set of agreements, processes, and tools. It features specific solutions designed to match the way different businesses acquire software and maintenance services. You can order a single platform or many platforms; one product or an entire suite; for one location or a network of offices around the world. Passport Advantage offers consistent worldwide features and pricing, simplified acquisition, and multilingual licensing. Whether yours is a small company, a centralized or decentralized department, or a member of the Fortune 500, there's a Passport Advantage solution to match the way your organization manages software and services.

The Passport Advantage program consists of a worldwide general acquisition program that allows customers to receive discounts for IBM software purchases based on their purchase volume over time and the size of individual transactions – all combined to give you lower prices. There is also an option to acquire groups of products to be deployed across an enterprise on a per-user basis.

Passport Advantage also includes Software Maintenance, which provides complete upgrade and cross-platform migration coverage for all commercially available IBM-distributed software. With Software Maintenance coverage, you can upgrade to new releases and new versions as your business needs dictate. Software Maintenance (including both subscription and support) is simple for your business to purchase and administer in one planning and budgeting cycle as part of your overall software acquisition.

ISSUE	IBM PASSPORT ADVANTAGE	OTHER VOLUME LICENSING PROGRAMS
What about your options?	You get more options. You can choose application solutions from any of IBM's distributed software brands—WebSphere, Data Management, Lotus, or Tivoli. Your software will run on all distributed platforms. And, you get to choose when to install new releases.	You get fewer options. You are limited in your selection of products from among their total catalog of software solutions. And, you may be forced to upgrade as new versions of software are released.
What if your participation in maintenance lapses?	Your license is perpetual—you can use the licensed software indefinitely. Software Maintenance can be reinstated, at a higher price than if you had continuously renewed, but without being required to purchase a new license.	A penalty may be imposed if your maintenance lapses. You could be required to buy a new license plus maintenance.
Is an upgrade commitment required?	No. You choose when to install new software releases.	From the time of your initial license purchase, you may be required to upgrade as new versions become available. You may have no choice.
Is there a required length of time for which you must commit to maintenance?	Only until your next anniversary. Coverage is optional after that.	Some programs require you to make a 2- to 3-year commitment.
Is there a minimum purchase?	No. You start realizing your Software maintenance benefits immediately.	Yes. Most require you to make several purchases before your maintenance benefit kicks in.

TABLE 1: PASSPORT ADVANTAGE

Software Subscription ensures that you have access to the latest software releases without paying additional fees. This means you never have to budget for upgrades because they're included with your subscription.

Technical Support helps keep your users up and running wherever they're working in the world. In fact, Technical Support includes several layers of support and, depending on the type of problem you're facing, may even provide you or your IBM Business Partner with a direct link to IBM developers. This is how IBM makes sure you have the tech support you need. It's your way of getting an increased return on your IBM investment in a total software solution.

With all this, you also get a personalized Web site where you can view your account history and proofs of entitlement. More on this later. In short, there's no easier way to lower software acquisition and administrative costs, ease migration between platforms, boost productivity, and increase profits. And there is no easier way to do business with any software vendor.

How Passport Advantage Works

GENERAL PROGRAM OFFERING

Each Passport Advantage acquisition is evaluated against two criteria: first, the size of that transaction; and second, the customer relationship as measured by the amount of current business you're doing under Passport Advantage. You receive a Relationship Suggested Volume Price (RSVP) level based on your initial acquisition of licenses and Software Maintenance under the program. That RSVP level is reviewed during each anniversary period to determine if the accumulated acquisitions during that period qualify your business for an improvement in the RSVP level and, therefore, a better price. In addition, for customers who achieve an RSVP Level of "D" or above, each acquisition transaction, depending on its size, may qualify for an even better price for that specific transaction. This is known as the Transaction SVP level, or TSVP. The larger the transaction, the better the TSVP. If eligible, you will always receive the better of the RSVP or the TSVP for your transaction.

Your additional locations or sites anywhere in the world can enroll under the same agreement so your business can leverage its combined acquisition power, including Software Maintenance renewals. Software Maintenance is included with licenses for one or two anniversaries, depending on the option chosen.

CEO PRODUCT CATEGORY ACQUISITIONS

CEO Product Categories are the premier offering for organizations that implement IBM distributed software solutions for their entire enterprise. If your business is willing to make an enterprise-wide commitment to IBM distributed software you'll benefit not only from a lower total cost of licensing and Software Maintenance, but also from ease of deployment, migration, and the management of technology changes. You must place an initial order for at least 500 users and the order must include at least one CEO Product Category that will be deployed across your enterprise. You may also choose additional solutions-based CEO Product Categories, again with a 500-user minimum for each, as well as individual products from the general Passport Advantage eligible product list.

Your additional locations or sites anywhere in the world can enroll under the same agreement. Additional Product Categories may be acquired for 500 users or more on an as-needed basis.

Purchasing CEO Product Categories is the best way for your business to standardize on IBM distributed software, manage your software assets, and reduce costs for software and Software Maintenance.

Take a Closer Look

Earlier in this article, I discussed customer expectations and presented some things you should look for in a volume licensing program. Let's look at how Passport Advantage stacks up against the competition (see Table 1).

Does Passport Advantage increase your options? Yes. Under Passport Advantage you actually enjoy more options. You can choose solutions from any of IBM's distributed software brands. Other vendors' programs offer fewer options and limit your selection of products from among their total catalog of software solutions – and they may only run on one platform. Also, Passport Advantage has no upgrade requirement, so you decide when to install new releases.

What if your maintenance participation lapses? With Passport Advantage, your license is perpetual. If Software Maintenance lapses, you simply reinstate it, at a higher price than if you had continuously renewed, but without having to purchase a new license. Other volume licensing programs may require you to acquire a new license as well as to reinstate maintenance.

What kind of upgrade commitment is required under Passport Advantage? Simply stated, none. Passport Advantage gives you the option to upgrade but it's never required. Other vendors' programs may require you to upgrade as new releases become available – in other words, you may have no choice, and that can get expensive.

Is there a required length of time for which you must commit to maintenance? With Passport Advantage, you get Software Maintenance for one year (or two if you choose that option). Software Maintenance is optional after that. Other vendors often require a two- or three-year commitment.

Does Passport Advantage allow you to combine your purchases of multiple IBM software brands and products to leverage your pricing? Yes. Other vendors don't give you this option, limiting you to only a subset of products within their catalog.

What is the minimum purchase required by Passport Advantage before you start realizing the benefits of maintenance? None. You realize the benefits of Software Maintenance from the beginning of your participation. Other volume licensing programs may require that you make several purchases before your maintenance benefit kicks in.

With Passport Advantage, you get volume licensing, upgrades, and both technical and how-to support. It's a total solution that ensures you're always covered when you install new software or upgrade to newer versions.

What's the Benefit?

When you shop for a volume licensing program, be sure you understand the program and who it benefits. Also consider these additional benefits of the Passport Advantage program – whether you're already a Passport Advantage customer, a potential customer, or an IBM Business Partner that sells or is thinking about selling Passport Advantage products:

- Lowers acquisition and administrative costs, lowers the cost of cross-platform migration and software upgrades, and includes new versions as well as new releases.
- Provides a single agreement for all IBM customers – one set of price levels that all customers can qualify for so all businesses receive attractive pricing.
- Offers customers who continue to make purchases under the program an opportunity to qualify for price levels for each transaction (TSVP) that are better than their established RSVP level. The larger the transactions, the better the TSVP.
- Provides a way for you to improve your RSVP levels between anniversary dates based on a periodic review of recent purchases.
- Includes automatic annual recalculation of your RSVP based on previous 12-month purchase history.
- Includes an agreement and enrollment forms that do not need to be signed, unless needed to fulfill local legal requirements.
- Provides no limits on the number of designated IT technical staff that can contact technical support for help.
- Includes 24x7 technical support for emergency calls ("Severity 1") at no extra charge. You're also provided with access to IBM technical voice support during normal business hours for help with your installations, "how-to," and code-related questions.
- Offers a secure Business Partner extranet site. The Worldwide Pricebook & Configurator tool allows Business Partners to view their customers' details, run what-if analyses, or load a previously saved configuration or part list. It's a quick and easy way for Business Partners to access their customers' information.
- Offers a secure Passport Advantage customer extranet site that allows you to view, download, and/or access your purchase his-

tory and proofs of entitlement, modify your contact information, handle software maintenance renewals, make additional acquisitions, and download the very latest releases and versions of products covered by your Software Maintenance as soon as the products are generally available.

Software Maintenance: You're Always Up-to-Date

The benefits of Software Maintenance are included with all Passport Advantage license acquisitions. Passport Advantage Software Maintenance is the most cost-effective way for your organization to ensure that its users have the latest technology of the most current releases at their fingertips and can maintain the highest levels of productivity.

Passport Advantage Software Maintenance also makes it easy for you to put IBM's world-class software support to work in a cost-effective and efficient way. Passport Advantage Software Maintenance provides easy access to responsive, cross-platform software and technical support around the clock and around the world.

This program provides prompt responses to your questions by offering the convenience of IBM call-in numbers throughout the calling areas in which you conduct business.

Passport Advantage Online Helps You Manage It All

One of the strongest tools provided by IBM in your Passport Advantage toolbox is Passport Advantage Online. Whether you're a customer or a Business Partner, this service can help you manage your installed base of IBM software.

CUSTOMER SITE TOOLS

As a Passport Advantage customer, your company has its own secure Web site that puts your software license account information, subscription renewals, and software downloads at your fingertips. Here's a glimpse of some of the services available on the Web:

- ***Proof of Entitlement (POE):*** Confirms the products, quantities, and services you have ordered and are eligible to install, or that you can begin to use. There's no need to wait for the physical delivery of POE certificates, since you can deploy software licenses, install subscription media, and use support services upon viewing your POE certificates on the Web.
- ***Customer statements:*** Quick and easy access to your Passport Advantage purchase history.
- ***Software download:*** Ability to download and install new releases or upgrades of covered products upon commercial availability. You don't have to wait for media delivery or keep track of CDs; simply download your entitled code whenever you need it.
- ***Popular offerings catalog:*** Allows the purchase of the top-selling IBM software products. The catalog functionality includes search by product description, search by product group (e.g., Data Management), and search by product type (e.g., License with Software Maintenance). You have the choice of purchasing directly from IBM at SVP, or sending your order to a Business Partner.
- ***Subscription renewal:*** Details coverage that is due for renewal; allows you to purchase through a Business Partner or directly from IBM at the Suggested Volume Price (SVP). (This functionality is not yet available in all countries.)
- ***Change contact information:*** Allows you to review and update the program contacts on file for subscription media delivery, renewal notifications, POEs, and general enrollment information.

RESELLER SITE TOOLS

As a Passport Advantage reseller, your company has its own secure Passport Advantage Web site. The site puts your customers' license information, Software Maintenance renewal, and pricing information at your fingertips. Some of the services available on the Web are:

- ***Account Information Tool (AIT):*** Shows the customer's Software Maintenance renewal month, product licenses with active or pending Software Maintenance coverage, company contact information, and the sales order history by category/date.
- ***Worldwide Pricebook & Configurator:*** The single source for Passport Advantage customer, part number, and pricing information for automated configuration, pricing, and customer quote generation for distributed software worldwide. It allows you to view your customers' details, run what-if analyses, or load a previously saved configuration or part list. The tool provides online help for all screens. If you're a distributor or house account, you can view your cost of goods. All resellers can view their customers' RSVP and, if applicable, the TSVP associated with the transaction being configured.
- ***Passport Advantage fundamentals tutorial:*** A detailed guide to the program. It contains information about enhancements, including IBM support. The tutorial is voice enabled and available in six languages (English, Spanish, French, Italian, German, and Japanese). It allows the user to focus on a specific content area with multiple examples of customer scenarios for reference and contains a testing module where you can test your program knowledge.
- ***Comments and questions:*** Gives you the ability to submit a question or comment and receive an answer from the appropriate IBM Passport Advantage contact.

Summary

Not all volume licensing programs are created equal. When shopping for such a program, be sure you understand its benefits as well as whom the program is benefiting.

With Passport Advantage, you can be assured that the program benefits you and can help you realize a greater return on your IBM investment. It's the most flexible, cost-effective way for your organization to reap the benefits of volume pricing for new software, acquire new releases of the latest technology, and gain access to the technical support that will keep your business up and running. Passport Advantage can help lower acquisition and administrative costs because new releases of the latest technology – if you choose to install them – are included. Passport Advantage will facilitate migration to new platforms, boost productivity with its Software Maintenance feature, and increase profits through volume licensing discounts.

As a Passport Advantage customer or reseller, not only will you leverage your investment in IBM technology, you'll have access to a wealth of IBM cross-brand and cross-platform solutions that meet your total business needs.

WebSphere Developer's Journal Advertiser Index...

WebSphere Developer's Journal Coming Next Month...

MISSION CRITICAL APPLICATIONS: MANAGING FOR SUCCESS
The introduction of pervasive e-business practices has elevated the complexity of managing IT for business success

THE MESSAGE DRIVEN BEAN IN WAS 4.0
Creating a pseudo MDB that can be integrated into your WebSphere application

LIFE SCIENCES WEB SERVICES
Use of a UDDI registry to publish and discover your services

AN INTERVIEW WITH MIKE COLLEARY
Vice President of North America Channel Sales, IBM software

BEHIND THE SCENES OF WAS 5.0
An up-close look at the latest features in the new release

WEB SERVICES IN A PERVASIVE COMPUTING ENVIRONMENT
Pervasive devices like mobile phones, PDAs, and pagers far outnumber computers, and the trend is accelerating

IBM TO BUILD EMERGENCY RESPONSE SYSTEM

(Washington, D.C.) – IBM has been selected to build a public safety data communications network for the Washington, D.C. region. The first interoperable wireless system to span multi-state government jurisdictions, the network will enable officials from more than 40 local, state, and federal agencies to communicate with each other in real time.

The Capital Wireless Integrated Network (CapWIN) is also designed to provide firefighters, police, transportation officials, and other authorized emergency personnel with wireless access to multiple government data sources during critical incidents. Improved access to information will help these "first responders" and public safety officials make vital public safety–related decisions.

The heart of CapWIN is IBM's First-Responder Interoperability Solution. It harnesses a cluster of IBM eServer pSeries UNIX servers running IBM WebSphere software for message routing and browser communications, and relies on a Global Directory developed by IBM to serve as a bridge connecting the disparate addressing systems in use by the various government agencies. The IBM solution offers security features that meet or exceed FBI standards for mobile data communications.
www.ibm.com

COMMERCEQUEST LEVERAGES IBM WEBSPHERE

(Tampa, FL) – CommerceQuest, an IBM Business Partner and certified solutions provider, has introduced a suite of comprehensive products designed specifically to complement IBM's WebSphere software. The CommerceQuest Suite for IBM WebSphere acts as an enabling engine to create additional connectivity to enhance WebSphere software capabilities, enable rapid customer implementations, and provide lasting synergies between WebSphere, IBM mainframe systems, WebSphere MQ, and other existing systems.

This product suite allows IBM and CommerceQuest customers to decrease the time, risk, and cost of their WebSphere implementations by minimizing the integration requirements necessary to unlock the value of their existing data and applications as XML Web services interfaces to support new WebSphere-powered solutions.
www.commerce quest.com

STELLENT DELIVERS PORTLETS FOR PORTAL 4.1

(Eden Prairie, MN) – Stellent, Inc., a global provider of content management solutions, has announced the availability of Stellent portlets updated to support the IBM WebSphere Portal Version 4.1.

Stellent's portlets use JavaServer Pages (JSP) tag libraries, Java 2 Enterprise Edition (J2EE) JavaBeans and J2EE Enterprise JavaBeans (EJBs) to provide a flexible and rapidly deployable environment for IBM WebSphere Portal customers. The Stellent tag libraries simplify content management integration within WebSphere, allowing developers to quickly deploy content-rich portals and Web sites.

The Stellent portlets are currently available through the Stellent Web site at www.stellent.com/ibm.

SITRAKA PERFORMASURE NOW SHIPPING

(Toronto) – Sitraka, a leader in J2EE performance assurance, has announced the release of Sitraka PerformaSure 1.6, a transaction-centric performance diagnosis tool for analyzing distributed J2EE applications for IBM WebSphere Application Server 4.0. Companies using IBM WebSphere can now use PerformaSure to detect and diagnose performance problems in their J2EE applications.

PerformaSure provides J2EE performance teams with critical investigative insight into how J2EE applications, application servers, and databases interact to serve end-user requests in transactional systems. E-business performance teams working with WebSphere now have access to Sitraka's complete performance assurance solution, which includes PerformaSure and Sitraka JProbe, the premier performance tuning toolkit for Java.
www.sitraka.com

IPEDO INTRODUCES XML VIEWS INTEGRATION TECHNOLOGY

(Boston) – Ipedo, a provider of software products for managing and delivering dynamic content over the Internet, has introduced Ipedo XML Views, an integration technology that unifies information management for portals, custom Web applications, and Web services initiatives. Integrated into its Ipedo XML Database and Ipedo Integration Manager, XML Views expands the reach of corporate information resources while reducing the complexity.

Ipedo XML Views are virtual, real-time snapshots of information from external XML and non-XML sources presented in a standard, common XML format to simplify integration tasks in applications. It can be deployed in conjunction with J2EE application servers such as IBM WebSphere.
www.ipedo.com

iWay Enterprise Integration Suite Leverages IBM WebSphere Software

(New York) – iWay Software, an Information Builders company and leader in middleware that accelerates e-business initiatives through the rapid integration of complex back-office systems, has announced that iWay has embedded IBM WebSphere software into its rapid back-office integration solution.

Five components of IBM WebSphere software are leveraged by the iWay software product line. The iWay Application Server is powered by IBM's WebSphere Application Server Advanced Edition; e-Commerce extensions from WebSphere Commerce Suite are included with the iWay Commerce Suite; MQSeries provides the foundation for iWay Messaging and Queuing; MQSeries Integrator has been embedded into iWay Enterprise Integrator; and VisualAge for Java has been extended to become the iWay Application Developer. Specialized iWay software components simplify the process of creating complete e-business solutions for enterprises with complex back-office environments.
www.iwaysoftware.com

WEBSPHERE USER GROUPS...

The Manhattan WebSphere User Group has held over 15 meetings since April 2000 and is led by Michael Mande (mmande@ websphere.org.) A full history of prior meetings and presentations is at www.websphere.org/ug/manhattan.html.

Immediate wealth that offers immediate benefits

J2EE Best Development Practices for Performance and Scalability

BY JIM O'HARA

I work as a Java developer for an insurance company located in New York City. With the rapid-fire change of pace in technology today, it seems that there really aren't enough hours in the day to keep abreast of the changes while remaining attentive to principles of good design methodology and best practices. I have personally found that a vital way to keep up-to-date with Java technology and coding methodologies is to frequently attend the user groups in the metropolitan New York area.

ABOUT THE AUTHOR

Jim O'Hara is a Java developer for a Fortune 400 insurance company in New York City. He specializes in the development of Web-based applications utilizing JSPs and servlets with WebSphere 4.0.

E-MAIL

jimohara@yahoo.com

One of the benefits I derive from being a member of a user group is that I get to hear pertinent information from notable authorities on a wide variety of topics. Many times there is an "aha" moment, and I am able to find a new solution to an old, vexing problem. One of the drawbacks in this day of hyperinformation and constantly changing technology is that the information I hear at a presentation may not be readily applied to the latest and greatest project that I'm working on. This was not the case with the presentation at the June Manhattan WebSphere User Group.

In that meeting, the Manhattan WebSphere User Group heard directly from Harvey Gunther, a senior IBM analyst with the IBM WebSphere Product Performance Group in Durham, NC, on best development practices. Best practices is a subject that always needs to be on everyone's radar screen. Harvey's presentation, "WebSphere J2EE Best Development Practices for Performance and Scalability," was an insightful and engaging talk on this very important topic in enterprise development (The presentation is available at www.web sphere.org/ug/manhattan.html.)

Harvey's discussion was not just a checklist of heuristics to follow. Instead, he carefully identified that best practices are based on measurable and quantifiable rationale. A resounding reminder on many of his slides was the phrase, "Plan and design to avoid problems!!!".

Quick Summary

There is a demonstrated performance benefit from utilizing best practices. Good application design for performance and scalability starts with certain fundamental practices and avoiding certain mistakes.

Best practices is not a litmus test or checklist for performance and scalability. It is a starting point for application design for performance and scalability. Some of the key points:

HTTP SESSION

- Avoid large HttpSession data.
- Make servlet state data application specific.
  - Java Heap is not unlimited.
  - HttpSession data has to go some where.
- Release HttpSessions when finished.
  - Explicitly invalidate HttpSession.

SERVLETS AND JSPS

- Minimize synchronization in servlets.
- Minimize use of Println s.
  - Indiscriminate use of println s slows performance.
- Don't use SingleThreadModel
- Avoid or minimize – jsp:include
  - *Jsp:include:* Elegant frameworks: used to build composite JSPs dynamically.
  - *Bad performance:* Each included JSP is a separate servlet.
- Avoid "Beans.instantiate": Empty Bean Tags.
  - Make sure that declared "jsp:use bean(s)" can be found in declared scope.
  - Enforce Model View Controller
- Don't create HttpSessions in JSPs by default.
  - Session is created by default.
  - Use the jsp tag <%@page ses sion="false"%> to disable the default state.

What I took away from this session was not just more coding to interpret and digest, but a philosophy of how and why to design and code in a certain fashion. Hearing from a speaker of Harvey's caliber is immediate wealth from which I can derive benefits.

Harvey mentioned that he is currently taking WebSphere 5.0 through its paces. Version 5.0 was expected to be available sometime this summer. Gee. More changing technology.

The talk Harvey gave was based on his September 7, 2000, document, "WebSphere Application Server Development: Best Practices for Performance and Scalability" (available at www -3.ibm.com/soft ware/webservers/appserv/perform ance.html).

Achieving More with Less in a Difficult Economy

The universal value of XML

BY ALEX EL HOMSI

Companies confront more challenges today than ever before. The business environment of the Internet-driven economy is more volatile, demanding, and complex. Given the events of the past year, businesses uncertain about the financial future have been reluctant to increase their workforce or spend capital. Pressures to improve profit margins have forced companies to produce more with fewer workers. The same pressures to improve productivity have forced companies to change the way they are managed and structured and how they use technology.

Not only is business constantly changing, technology is expanding and changing at an equally rapid rate. It has become increasingly difficult for IT organizations to keep pace with the day-to-day pressures, let alone anticipate the challenges they will be forced to address in the future. Yet hiring skilled IT resources or spending more money to address the challenges of today and tomorrow is not an option. Doing more with less becomes the imperative.

The reality of budget cuts and workforce reductions has dictated that IT organizations be constantly on the lookout for "faster, better, cheaper" ways of responding to ever-changing business needs. While most companies have been investing in technology on a project-by-project basis, many are just now realizing the importance of open standards to drive consolidation and consistency across the extended enterprise, effectively cutting costs. The fact is that a typical IT infrastructure is a broad mix of hardware, software, middleware, groupware, and homegrown applications. Dealing with this mix requires extensive skills and experience and many resources to make the different systems work together.

Among all of today's existing and emerging open standards, XML (eXtensible Markup Language) is definitely the one that can help IT organizations achieve more with less. XML shines by virtue of the simplicity and universality of the language. It's easy to learn, and there are an increasing number of software vendors that publish XML-based interfaces in the form of Web services, which promise an easier integration and interchange mechanism. In a perfect world, all content would be available as XML. Manipulating a single data format is much easier for developers than dealing with SQL, CICS, or even Lotus Notes data, which all require much more programming.

The availability of content as XML calls for a new XML-based component framework to become the foundation of every IT project. It's amazing that many IT organizations still don't enforce standardization and reusability in their development practices when it is clear that this is the only way to achieve significant productivity gains. Rather than reinventing the wheel and writing code, developers can be much more productive by assembling high-level, reusable building blocks that abstract architectural considerations. The popular and highly sophisticated Lego brick toys provide an excellent analogy. Who hasn't spent hours creatively assembling a standard set of Lego bricks into any form imaginable?

XML is the IT equivalent of the standard Lego shape enabling systems to work together. Most major technology companies are in the business of selling tools and services to build the bricks. Taking that route should be avoided because it's not adding real value to an organization's core business. The smarter approach is to look for vendors who specialize in providing the bricks themselves, and who have already done a great deal of the hard work, essentially leaving the user to configure and assemble the final results. Some vendors provide the bricks in combination with visual rapid application development tools, which further simplifies the user's efforts.

Using the bricks without a standard framework in place would be like buying material when the plans have yet to be defined. Understanding what kinds of bricks are needed to implement highly complex business applications is key. Business applications are designed almost exclusively to streamline business processes, which is the key driver behind all IT investments. Building business applications consistently across an IT organization requires a framework with the following components:

- An advanced business process management platform to visually model processes, the organization, and all content using XML
- A rapid application development environment to visually build Web applications by assembling XML components to enable user interaction with XML content
- Universal XML adapters that seamlessly enable bidirectional access to non-XML content and allow developers to work with XML regardless of the original content

Open standards are providing increased agility to both IT and business objectives. In a universal framework with the right building blocks in place, the promise of connecting IT initiatives directly to business objectives is real. The use of a standard language and a universal framework allows people and systems to keep pace with the complex and constantly changing business environment – which in today's world means being able to do more with less.

ABOUT THE AUTHOR... Alex El Homsi is the founder and CEO of Trilog Group, a software company that delivers open standards business process management and rapid J2EE and XML application development.

E-MAIL
alex@triloggroup.com