SYS-CON MEDIA

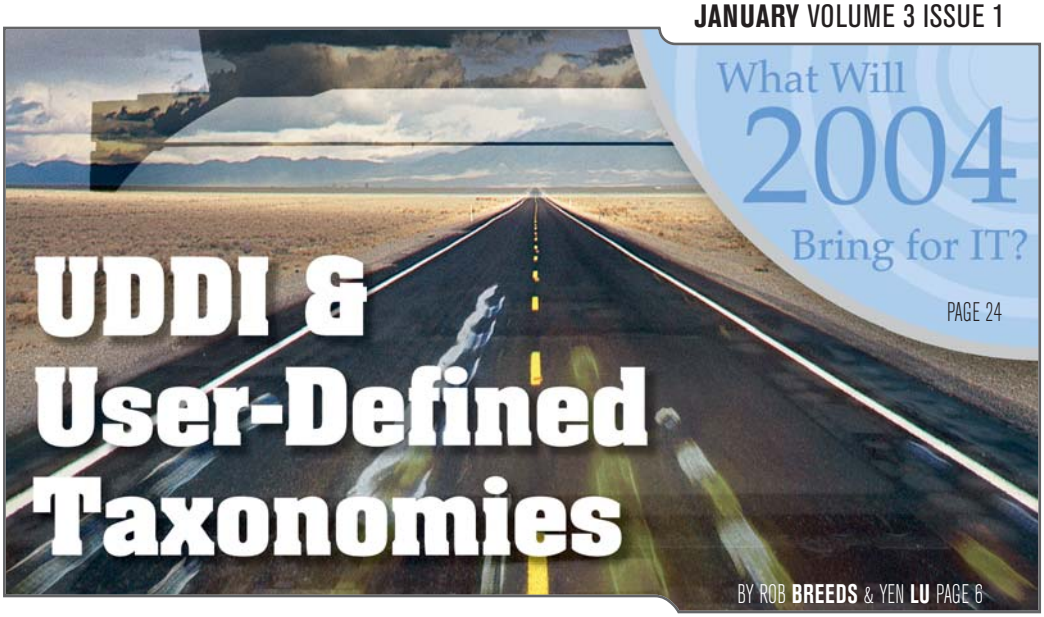
EXCLUSIVE: INDUSTRY EXPERTS LOOK AT THE COMING YEAR PAGE 24

WebSphere® JOURNAL

www.SYS-CON.com/WebSphere

The World's Leading Independent WebSphere Resource

JANUARY VOLUME 3 ISSUE 1



DISPLAY UNTIL MARCH 31, 2004
$8.99US $9.99CAN
01>
0 09281 03422 3

TODAY'S MISSION: INTEGRATION — FULL SPEED AHEAD

KEY PRACTICE SKILLS:

WEBSPHERE APPLICATION SERVER

WEBSPHERE PORTAL SERVER

WEBSPHERE MQ

MQSERIES WORKFLOW

WEBSPHERE MQ INTEGRATOR

WEBSPHERE BUSINESS INTEGRATOR & CROSSWORLDS

WEBSPHERE COMMERCE SUITE

WEBSPHERE STUDIO APPLICATION DEVELOPER

VISUALAGE FOR JAVA

WEB SERVICES

DB2 & INFORMIX

PLANNING YOUR WEBSPHERE, WEB SERVICES OR PORTAL STRATEGY?

UPGRADING YOUR IT INFRASTRUCTURE?

NEED A JUMP START ON WEBSPHERE 5.0 AND WEBSPHERE PORTAL 4.1?

Prolifics' Jump Start Solutions will...

Educate. As 1 of 5 WebSphere service providers retained by IBM, Prolifics' Consultants are top-ranked by IBM in delivering WebSphere Solutions. We give your team an edge by transferring our extensive knowledge and experience.

Enable. We prepare your e-infrastructure for ease of application integration, migration and deployment. We recognize risks and delays that could hinder your project timeline and budget.

Empower. Providing your team with best practices, documentation and guidelines, we identify the best strategy leading to your must-meet business goals.

PROLIFICS' JUMP START SERVICES MENU:

- **Jump Start:** Extend your enterprise with WebSphere Portal 4.1
- **Jump Start:** Build, publish, manage, optimize with WebSphere 5.0
- **Jump Start:** Get started with Site Analyzer
- **Jump Start:** Integrate WebSphere MQ into a J2EE environment
- **Jump Start:** Web services – build to integrate

Call 800.675.5419 or 212.267.7722

http://www.prolifics.com

Prolifics

IBM Business Partner

Solving Business Problems with Technology Solutions since 1978
Expert Delivery of WebSphere, Integration and Business Transformation Solutions

For a FREE white paper on how to Integrate Enterprise Resources Using Web Services Technologies
Go to: http://www.prolifics.com/webservices

e-business

PRIORITY CODE: DJ0702JS

Forget something?

Post-launch is NOT the time to be verifying web applications.

The wild blue yonder of operational monitoring and management is extremely unforgiving. Which means that going live with the monitoring software you used in development is a great way to go dead—quickly! You simply can't support operations if your staff is drowning in details provided by development profiling tools and debuggers. **Let NetIQ cover your apps...with AppManager.**

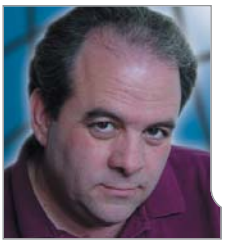
AppManager—the industry's easiest-to-use Systems Management suite—is a proven management system for monitoring J2EE application servers, databases, operating systems and even end-user response time. NetIQ's AppManager monitors ALL application components—not just your server. **NetIQ. Nobody does UNIX better. Nobody.**

Visit us at www.netiq.com/solutions/web to learn how we can help you address the challenges of your operational monitoring and management.

netiq
Work Smarter.

How to Deal with IT Outsourcing? Fulfill Society's Unmet Needs

BY JACK MARTIN



The whole point of my recent series of editorials has been to get people thinking about what has happened to the job market for IT developers and administrators – and where it is going next. Over the past two months I have written about what I have witnessed at countless dot coms and development companies around the country and where I see the IT work market going.

I've made some people angry. Some of them have even called me names or said I am out of touch, but others have agreed with me. I know that some of my examples went to the extreme, but that was the whole point; the current situation with offshore outsourcing has gone to the extreme. The entire market is now swinging to a new extreme of sending work outside of the country. This is not a good thing.

The reality of human behavior is that people tend to move in very distinct patterns, with large, seemly disassociated groups thinking that what they are doing and thinking is unique – when in reality their behaviors and thoughts are commonly held beliefs and attitudes that are part of mass movements.

Currently we have a classic labor-versus-management conflict bubbling up with a very special twist. Business in its most simple terms is a three-legged stool composed of labor, capital, and management. No business can exist without the three components.

Labor, when applied with no thought or direction, is of little or no value. All one ends up with is, at best, subsistence living and basic survival.

Capital is a very tricky concept for most people because they mistakenly think that capital has power. Nothing could be further from the truth. Capital has no kinetic energy; it possesses only potential energy. It can be transformed to kinetic energy only by the application of labor.

Management applied intelligently to labor and capital – with a unifying direction toward a specific goal – is what drives profits (or losses, depending on how well the labor and management have fulfilled society's unmet needs with their collective outputs). The winners are showered with profits and an increased standard of living, and the losers experience the reverse effect.

When the United States started shedding factory jobs in the 1960s no one seemed to care, except of course the factory workers who where thrown out of work permanently. This time I am not using an extreme example. Think about it for a moment: Do you drive a car or watch a television made in the United States? You probably don't own an American-made car, as I don't know even one code writer or system administrator who does. As far as televisions go, there hasn't been a mass-produced television made in the United States in over 30 years. Do you eat the All-American hamburger? Did you know that the United States has become the world's largest importer of beef?

It's getting pretty hard to find anyone in this country who drives a Chevy and watches an old black-and-white television while eating some Kansas-raised beef.

The reason nobody ever cared is that these seemly low-paying jobs were being replaced by much higher-paying replacement jobs. They told us it was good for the economy and that knowledge-based jobs in the service sector were the road to independence and a standard of living that could only go up.

Guess what? They were wrong. Nobody ever anticipated third-world countries coming into the U.S. economy and taking high-paying jobs from college-educated Americans. This phenomenon is the single largest unintentional consequence of the rise of the Internet. It seems that the new replacement jobs are only lower-paying, downward-mobility jobs.

A national debate is brewing, but don't look to your favorite politicians for the answer. You know more about what's going on than they do.

Information technology is here to stay and will continue to embed itself into our society. The only answer is innovation and the development of new products through the use of technology to fulfill unmet human needs.

ABOUT THE AUTHOR... Jack Martin, editor-in-chief of ***WebSphere 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. Jack is coauthor of the upcoming book, ***Understanding WebSphere***, from Prentice Hall. **E-MAIL...** jack@sys-con.com

WebSphere® JOURNAL

ADVISORY BOARD

Richard Arone, David Caddis, Mike Curwen, Devi Gupta, Lloyd Hagemo, Barbara Johnson, Shannon Lynd, James Martin, Doug Stephen, Victor Valle

EDITOR-IN-CHIEF	JACK MARTIN
EXECUTIVE EDITOR	GAIL SCHULTZ
CONTRIBUTING EDITOR	PATTI MARTIN
PRODUCT REVIEW EDITOR	JAY JOHNSON
SPECIAL PROJECTS EDITOR	RICHARD ARONE
MANAGING EDITOR	JEAN CASSIDY
EDITOR	NANCY VALENTINE
ASSOCIATE EDITORS	JAMIE MATUSOW JENNIFER VAN WINCKEL

WRITERS IN THIS ISSUE

Kulvir S. Bhogal, Rob Breeds, Dave Danielson, Mark Dixon, Lloyd Hagemo, Ravi Kalidindi, Kwang S. Kang, Yen Lu, Jack Martin, Jay Parikh

SUBSCRIPTIONS

For subscriptions and requests for bulk orders, please send your letters to Subscription Department.
SUBSCRIBE@SYS-CON.COM
Cover Price: $8.99/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

PRESIDENT AND CEO	FUAT KIRCAALI
PRESIDENT, SYS-CON EVENTS	GRISHA DAVIDA
GROUP PUBLISHER	JEREMY GEELAN
TECHNICAL DIRECTOR	ALAN WILLIAMSON
SENIOR VP, SALES & MARKETING	CARMEN GONZALEZ
VP, INFORMATION SYSTEMS	ROBERT DIAMOND
VP, SALES & MARKETING	MILES SILVERMAN
PRODUCTION CONSULTANT	JIM MORGAN
FINANCIAL ANALYST	JOAN LAROSE
ACCOUNTS RECEIVABLE	CHARLOTTE LOPEZ
ACCOUNTS PAYABLE	BETTY WHITE
ADVERTISING DIRECTOR	ROBYN FORMA
DIRECTOR OF SALES AND MARKETING	MEGAN MUSSA
ADVERTISING SALES MANAGER	ALISA CATALANO
ASSOCIATE SALES MANAGERS	CARRIE GEBERT KRISTIN KUHNLE BETH JONES
CONFERENCE MANAGER	MICHAEL LYNCH
NATIONAL SALES MANAGER	SEAN RAMAN
LEAD DESIGNER	LOUIS F. CUFFARI
ART DIRECTOR	ALEX BOTERO
ASSOCIATE ART DIRECTORS	RICHARD SILVERBERG TAMI BEATTY
WEB DESIGNERS	STEPHEN KILMURRAY CHRISTOPHER CROCE
CONTENT EDITOR	LIN GOETZ
CIRCULATION SERVICE COORDINATORS	SHELIA DICKERSON EDNA EARLE RUSSELL LINDA LIPTON
CUSTOMER RELATIONS MANAGER	BRUNI STAROPOLI

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® Journal (ISSN# 1535-6914) is published monthly (12 times a year).
Postmaster send address changes to:
WebSphere Journal, SYS-CON Publications, Inc.
135 Chestnut Ridge Road, Montvale, NJ 07645

WORLDWIDE NEWSSTAND DISTRIBUTION
CURTIS CIRCULATION COMPANY, NEW MILFORD, NJ

FOR LIST RENTAL INFORMATION:
KEVIN COLLOPY: 845 731-2684, KEVIN.COLLOPY@EDITHROMAN.COM
FRANK CIPOLLA: 845 731-3832, FRANK.CIPOLLA@EPOSTDIRECT.COM

Two years without a vacation.
The application's up. It's down.
It's up. It's down.

I'm to blame. Steve's to blame.
Someone's always to blame.

Not any more.

Get Wily.™

Enterprise Java
Application Management
1 888 GET WILY
www.wilytech.com

wily technology

UDDI & User-Defined Taxonomies

Part I: Create and test a user-defined taxonomy in WebSphere Studio

BY ROB **BREEDS** & YEN **LU**

UDDI (Universal Description, Discovery, and Integration) plays the pivotal role of matchmaking between service provider and service requester. Sophisticated publish and inquiry capabilities allow providers to describe their offerings, and seekers to locate them. The quality and depth of a service description is critical to how easily that service can be found by interested parties. Likewise, result sets from find operations can be unmanageable if search criteria are imprecise.

ABOUT THE AUTHOR

Rob Breeds is a staff software engineer at IBM Hursley Laboratories, UK, and is lead developer for user interfaces and tooling in IBM WebSphere UDDI Registry. He has worked on development projects since WebSphere 2.0. Rob has over 20 years of experience in software development and has worked in technical sales, marketing, and publishing. Prior to joining IBM, Rob was a TPF mainframe developer at Galileo International.

E-MAIL
rob_breeds
@uk.ibm.com

Services can be described in UDDI in terms of contact information (White Pages), classification (Yellow Pages), and invocation details (Green Pages). Support for taxonomic classification of all entities in UDDI helps users find and register Web services in a standard way. Default taxonomies such as NAICS, UNSPSC, and ISO 3166 in UDDI v2-compliant registries enable classifications in broad terms such as industry type and geographic location. These, however, may not be sufficiently precise for describing a given Web service – and waiting for the underlying standardized taxonomies to change may not be an option. Also, organizations may need to classify entities in accordance with their own schemes to meet internal standards, e.g., corporate standards. These issues can be addressed through the use of user-defined taxonomies.

The IBM WebSphere UDDI Registry in WebSphere 5.0.2 and WebSphere Studio 5.1 (Application and Site Developer editions) and above provides support for user-defined taxonomies. Users can create their own taxonomies and make them available for use with the registry. This article will show you how to create and test a user-defined taxonomy in the Unit Test environment in IBM WebSphere Studio 5.1.

User-Defined UDDI Taxonomies in the WebSphere UDDI Registry

UDDI registries use taxonomies to classify entities in a consistent manner. In order to do this, taxonomies are typically "checked," that is, the category value assigned to a particular UDDI entity must be one of a set of valid values. The WebSphere UDDI Registry has several "internal" taxonomies, and with the tools presented in this article you can easily add your own.

A taxonomy data file contains the valid values, along with additional data that identifies the taxonomy the value belongs to, a description of the value, and the parent category of the value. The "#" character is the default delimiter.

The first few lines of the example data file show the required format:

```
procclas#10.00#Create Customer & Business Value#10.00
procclas#10.05#Understand and Choose Value#10.00
procclas#10.10#Create and Extract Value#10.00
procclas#10.15#Communicate and Deliver Value#10.00
procclas#15.00#Manage Relationships#15.00
```

Each row represents a separate category in the taxonomy. The first value in the row, "procclas", is the taxonomy name. This can be 1-8 characters, and is the same for every row of data in the taxonomy.

The second value, up to 32 characters in length, represents the unique code within the taxonomy that is used to check that keyedReference key values are valid.

To describe a taxonomy value, the third field takes up to 128 characters of text. UDDI does not require the description for publishing entities, but it is a useful hint about the meaning of the key value. For example, "15.00" doesn't mean much, but with "Manage Relationships" it's clearer.

The last value in the row is the parent code. Like the code value, it must be 1–8 characters in length. The parent code allows you to give the taxonomy hierarchical structure using categories and subcategories. For example, consider a food types taxonomy in which one of the values is:

```
foodtype#fruit#Fruits#food
```

You could define "citrus" as being a subcategory of "fruit", and "lemon" as a subcategory of "citrus" by defining two new values as:

```
foodtype#citrus#Citrus Fruits#fruit
foodtype#lemon#Lemon#citrus
```

The parent code value of "fruit" is a reference to the previous code value of "fruit". WebSphere Studio Web Services Explorer and the user console in the WebSphere UDDI Registry both use the parent code value to help render a tree view of taxonomies.

The taxonomy data file should be in UTF-8 encoding, without the "byte order mark" header. The next section explains how to create a taxonomy data file.

Create a User-Defined UDDI Taxonomy in WebSphere Studio

Let's create a taxonomy to describe a Web service that acts as a customer address book for a fictitious corporation. The corporation already has a taxonomy data file, process.tax (available for download from www.sys-con.com/websphere/sourcec.cfm), that incorporates its corporate standards.

Looking at process.tax, one category that this Web service might fit into is Manage Customer Relationships, with a key value of 15.05. However, it would be more precise to have a subcategory under this one to refer to the action of managing customer addresses. Let's add a new subcategory called Manage Customer Addresses, with a key value of 15.05.05. Before you begin, ensure that TCP/IP port 9080 or 7080 is free if you are using WebSphere Studio Application Developer or Site Developer editions, respectively. To create the taxonomy, perform the following steps:

1. Launch WebSphere Studio with a new workspace and note its location (<workspace directory>).
2. Select the main menu item Window > Preferences to open the Preferences dialog.
3. In the left pane of the Preferences dialog, expand the Workbench node and select the Editors node. In the resulting right pane, select the radio button labeled "Other", select UTF-8 in the adjacent dropdown menu, and then press OK.
4. From the main menu, select File > New > Project… to open the New Project wizard.
5. In the New Project wizard, select Simple in the left pane followed by Project in the right and press Next.
6. Enter "taxonomy" for the project name and press Finish.
7. Select the main menu item File > New > Other… to bring up the list of New wizards.
8. In the list of New wizards, select "Simple" followed by "File" in the left and right tree views, respectively, and press Next.
9. Select "taxonomy" in the tree view, enter "process1.tax" for the file name, and press Finish.
10. Open the process.tax file in a text editor and paste its contents to process1.tax in the right edit pane. Locate the line

```
procclas#15.05#Manage Customer Relationships#15.00
```

and add the new line

```
procclas#15.05.05#Manage Customer Addresses#15.05
```

after it. Save the file by pressing CTRL+S. The end result should look something like Figure 1.

Now Manage Customer Addresses is a subcategory of the more generic Manage Customer Relationships parent. The taxonomy data file is saved under <workspace directory>/taxonomy. You will now use the Unit Test UDDI Wizard to deploy a UDDI registry with this taxonomy:

1. Select main menu item File > New > Other… to open the list of New wizards. In the resulting dialog, select "Web Services" in the left pane, followed by Unit Test UDDI in the right, and then press Next to proceed to the Unit Test UDDI Registry Configuration page.
2. Select the "Deploy the Unit Test UDDI Registry" radio button and "Private UDDI Registry For WAS v5 with Cloudscape" in the Private UDDI Registry type dropdown and press Next to proceed to the Unit Test UDDI Registry with Cloudscape Configuration page.
3. Press the Add… button adjacent to the User Defined Categories table to bring up the Add a New Category dialog.
4. Enter "process" for the Name. For the File field, use the Browse… button to navigate to the directory <workspace>/taxonomy and pick the process1.tax file. Ensure that the "Checked" checkbox is selected. The final dialog and wizard page should look similar to Figure 2.
5. Press OK to dismiss the dialog and add the taxonomy to the User-defined categories table. Back in the wizard, press Finish.

"The quality and depth of a service description is critical to how easily that service can be found by interested parties"

The wizard will configure and run an instance of the WebSphere UDDI Registry in the WebSphere Test Environment and then launch the Web Services Explorer. The Web Services Explorer will show the details of the registry, including the user-defined taxonomy, in its "User Defined Categories" section. We will test the user-defined

ABOUT THE AUTHOR

Yen Lu is an advisory software developer at IBM Canada Ltd., where he is currently responsible for the architecture and development of the Web Services Explorer and the Unit Test UDDI Wizard in the IBM WebSphere Studio suite. Yen holds a MASc in electrical engineering from the University of Toronto.

E-MAIL

yenlu@ca.ibm.com

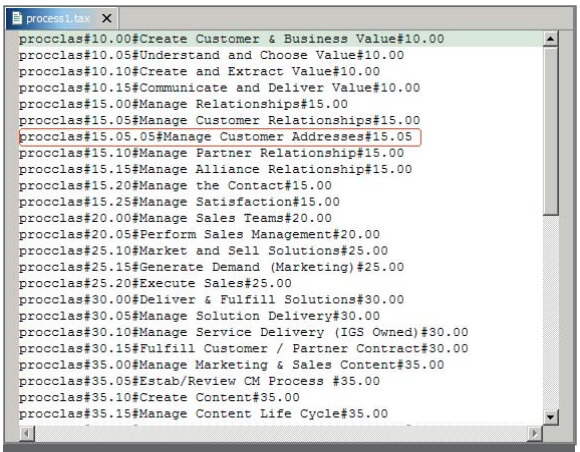


FIG. 1: MODIFIED TAXONOMY DATA WITH THE NEW SUBCATEGORY HIGHLIGHTED IN RED

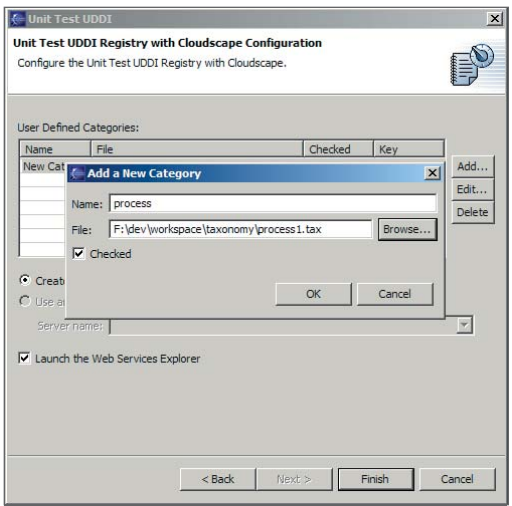


FIG. 2: ADD A NEW CATEGORY DIALOG AND UNIT TEST UDDI REGISTRY WITH THE CLOUDSCAPE CONFIGURATION WIZARD PAGE

taxonomy using the Web Services Explorer in the next section.

Test the User-Defined UDDI Taxonomy in WebSphere Studio

WebSphere Studio provides a tool called the Web Services Explorer for working with UDDI registries. That provides functions for UDDI publication and discovery. You can use the tool to publish the customer address book Web service with the new Manage Customer Addresses subcategory from the process taxonomy and later find it using this subcategory. Before you can begin, you need to deploy the Web service.

1. Select main menu item File > New > Other… to bring up the list of New wizards and then select Web in the left pane, followed by Dynamic Web Project in the right pane, and then press Next to proceed to the Dynamic Web Project page.
2. Enter "AddressWP" in the project name field and then press Finish. You do not need to switch to the Web perspective when prompted to do so.
3. In the Navigator view, Expand AddressWP, select the JavaSource folder, and then select main menu item File > Import… to bring up the list of Import wizards. Select "File system" as the input source and then press Next to open the File System Import wizard.
4. Use the Browse… button to navigate to and then select the directory <WebSphere Studio install directory>/wstools/eclipse/plugins/com.ibm.etools.webservice_5.1.0/samples/JavaBean_example. Back in the wizard, expand the "JavaBean_example" node in the left pane, select the checkbox beside the "webservice" node and press Finish to import the customer address book Web service's JavaBean code and dismiss the wizard.
5. Back in the Navigator view, expand AddressWP's JavaSource folder, followed by its webservice and addressbook children. Right-click on AddressBook.java and select Web Services > Deploy as Web Service to launch the Deploy as Web Service wizard.
6. In the wizard, press the Finish button to deploy the customer address book Web service.

The Web service is now deployed and ready to be published to a UDDI registry. As required by UDDI, you must first publish a business entity to own this service.

1. Scroll down the Web Services Explorer's Actions pane to the Publish link near the bottom. Click on the Publish link to load the Publish Form into the Actions pane.
2. Enter any user ID and password. *Note:* You must use the same ID and password to modify or unpublish previously published items.
3. Enter "My Business" in the Name field and press Go to publish the business. The Status pane will display a message reflecting the outcome of the operation. It should state that the operation was successful and the Actions pane should load the details of the business.
4. Scroll down to the bottom of the Actions pane and click on the Publish Service link to load the Publish Service form.
5. Select Advanced for the Publication format.
6. Click on the Browse… link adjacent to the WSDL URL field to open the WSDL Browser. In the browser, select AddressWP as the Web Project and ensure that the WSDL URL dropdown points to http://localhost:<port>/AddressWP/wsdl/webservice/addressbook/AddressBook.wsdl and then press Go to transfer the WSDL URL selection to the form and dismiss the dialog.

"Defining your own taxonomies gives you a better vocabulary to describe any UDDI entity and offers users a richer set of search criteria"

Optimize J2EE.

The J2EE revolution is here to increase performance, value, and lower IT costs.

It's a solution from Mercury Interactive that makes your whole J2EE applications ecosystem work right.

It's the fastest way to find the toughest J2EE problems from the development to the live applications. Even pinpoints root cause and the line of source code.

It's about more than delivering J2EE applications. It's about delivering applications that work and yield real business value.

It's the Business Technology Optimization revolution.

And Mercury is the only one bringing you a revolutionary solution for J2EE.

Download our free white paper, "**Diagnosing J2EE Performance Problems Throughout the Application Lifecycle**,"

at **www.mercuryinteractive.com/optimizej2ee**

Get Optimized.™

MERCURY INTERACTIVE

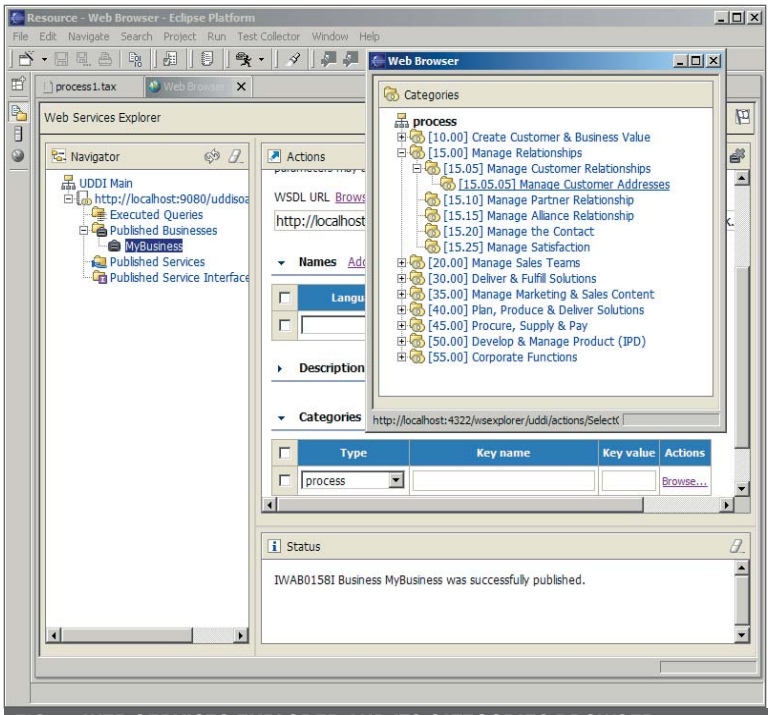


FIG. 3: WEB SERVICES EXPLORER AND ITS CATEGORIES BROWSER

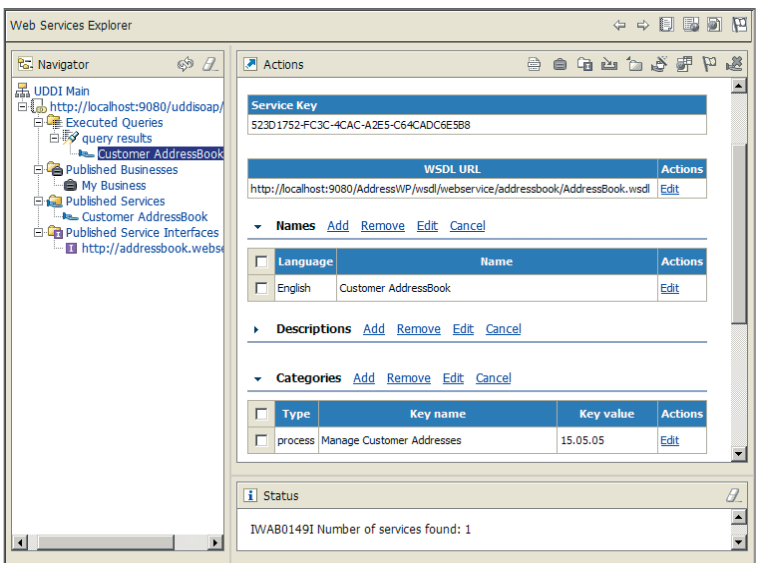


FIG. 4: WEB SERVICES EXPLORER SHOWING THE CUSTOMER ADDRESSBOOK DISCOVERED USING A USER-DEFINED TAXONOMY

7. Click the Add link in the Names section of the form to add a new name for the UDDI service. The name will appear in the Names section inside a table. There are two columns: Language and Name. Enter "Customer AddressBook" in the Name field.
8. Scroll to the bottom of the form and click on the Add link beside the Categories label to add a category to this service. The category added to the Categories section of the form consists of a Type (taxonomy), Key name, and Key value. For a known taxonomy, there is a Browse… action to display a user-friendly tree view from which a particular category or subcategory can be chosen.
9. In the newly added category, change the type to "process" and click on the Browse… link to open the Categories browser for the process taxonomy.
10. The Categories browser displays categories and subcategories sorted according to their key value. Expand the [15.00] Manage Relationships category, followed by the [15.05] Manage Customer Relationships subcategory, and then select the subcategory: [15.05.05] Manage Customer Addresses. Figure 3 shows the Web Services Explorer and the Categories browser with the subcategory underlined. After selecting the subcategory, the Categories browser should close and the key name and key value fields of the new category should match those of the selection. You can add multiple categories and subcategories.
11. Click Go to publish the service.

The Status pane should provide feedback on the result. If successful, you will see Manage Customer Addresses in the Categories table of the Service's details in the Actions pane. It should look similar to the Actions pane in Figure 4.

To complete verification of the user-defined taxonomy, perform a UDDI inquiry, including a category in the search criteria:

1. Select the registry node immediately underneath UDDI Main in the Navigator pane of the Web Services Explorer. The Publish form should load.
2. Select the third icon from the right of the Actions pane toolbar (a flashlight) to load the Find form.
3. The Find form will appear in the Actions pane. In the "Search for" dropdown, select Services and then choose the "Advanced" radio button for the Type of search.
4. Scroll down the form until you see the Categories heading. Click the "Add" link to add a category for the search. The new category appears as a new row in the table inside the Categories section.
5. For the new category, change the type to "process" and click on the Browse… link in the Actions pane. Follow Step 10 of the publish procedure to select the Manage Customer Addresses subcategory and then press Go at the bottom of the form.

When the search operation completes, the Status pane will inform you that one service was discovered. This should be the Customer AddressBook service. Figure 4 shows the expected results.

Conclusion

Defining your own taxonomies gives you a better vocabulary to describe any UDDI entity and offers users a richer set of search criteria. In Part 1 of this series we set out to show how you can create taxonomies for UDDI registries and try them out in the WebSphere Studio test environment. In Part 2 we'll demonstrate how to deploy the same taxonomy in the IBM WebSphere UDDI Registry, and offer more useful tips on publishing and using taxonomies.

Resources

- *Universal Description, Discovery, and Integration:* www.uddi.org
- *North American Industry Classification System:* www.census.gov/epcd/www/naics.html
- *United Nations Standard Products and Services Code:* www.unspsc.org
- *ISO 3166 Maintenance agency:* www.iso.ch/iso/en/prods-services/iso3166ma/index.html

For the sake of performance

Caching EJB Home Objects

BY KULVIR SINGH BHOGAL & KWANG SIK KANG

ABOUT THE AUTHOR

Kwang Sik Kang, consultant for IBM's Software Services for WebSphere, has extensive experience in providing J2EE and messaging-based enterprise solutions.

E-MAIL
kkwang@us.ibm.com

ABOUT THE AUTHOR

Kulvir Singh Bhogal works as an IBM Software Services for WebSphere consultant, devising and implementing WebSphere-centric solutions at customer sites across the nation. He has over 50 patents pending in a myriad of technology areas.

E-MAIL
kbhogal@us.ibm.com

The Java Naming and Directory Interface (JNDI) is an Application Programming Interface (API) used to locate resources registered in the naming service of a Java 2 Platform Enterprise Edition (J2EE) server such as IBM WebSphere Application Server version 5. The JNDI lookup process is an expensive operation and should be performed judiciously.

Caching of EJB home objects is a recommended approach to minimizing performance costs. This article will show the performance benefits of implementing this practice in a J2EE-centric application.

The application will consist of a servlet interacting with an EJB object. For learning purposes, our exercise uses IBM WebSphere Studio Application Developer version 5.

The EJB of Interest

A simple stateless session EJB (2.0) will provide the necessary features to perform this exercise. The following implementation is intentionally simple, so as not to lose sight of our goal. (see Listing 1)

The code consists of the standard EJB methods, along with a simple custom business method named "square". The method returns the square value of the parameter.

Expose the business method by promoting it to the remote interface. Once the method has been promoted, generate the deploy and RMIC (Remote Method Invocation Compiler) code. It is a good practice to test the implementation; in this case, leverage the Universal Test Client (UTC) to unit-test the EJB method (see Figure 1).

Invoking Externally

The servlet code in this article will invoke the remote method of the newly created EJB. The first sample shows an inefficient noncaching approach (see Listing 2).

A JNDI Lookup

Figure 2 shows three invocations of our servlet. Notice that the EJB home is looked up three times. You should note that the time burden associated with these lookups can be exasperated further if our client application is sitting on a different physical server from the one that hosts our EJBs. In such a setup (which is quite common), the temporal burdens of the lookup process are augmented by the latency of the underlying network.

Packing Our EJB Homes and Relocating to the Servlet init Method

We are now going to polarize our EJB lookup code to the init method for strategic purposes. You may recall that the init method of a servlet is called when the servlet class is first loaded. Depending on how we configure our servlet with our application server, we can opt to have the servlet class loaded either when our user first invokes the URL that corresponds to our servlet (think of a just in time–like approach) or when our server is first started.

By default, a servlet class will not be loaded upon startup of the server. You can change this by modifying the Web deployment descriptor (web.xml file). This can be done graphically, as shown in Figure 3 by checking the

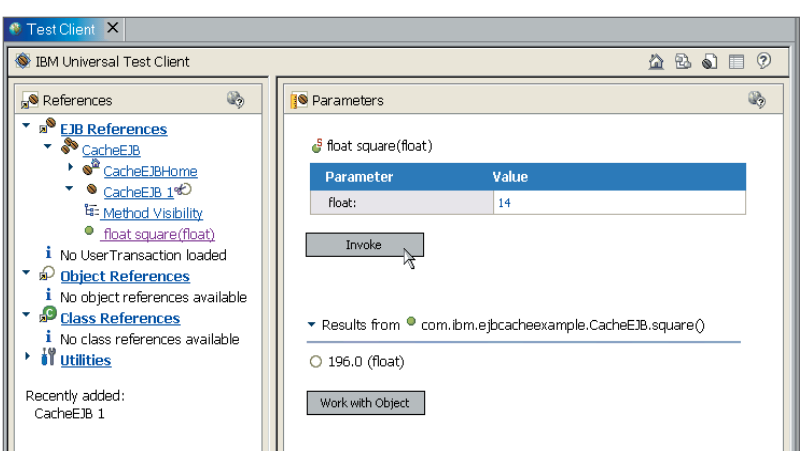


FIG. 1: USING THE UNIVERSAL TEST CLIENT (UTC) TO UNIT-TEST THE EJB METHOD

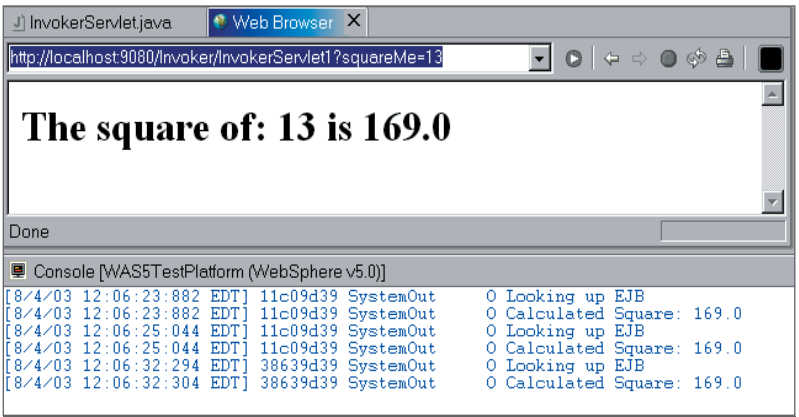


FIG. 2: IN THE NONCACHING APPROACH, EACH TIME THE SERVLET IS INVOKED, THE EJB HOME METHOD IS LOOKED UP

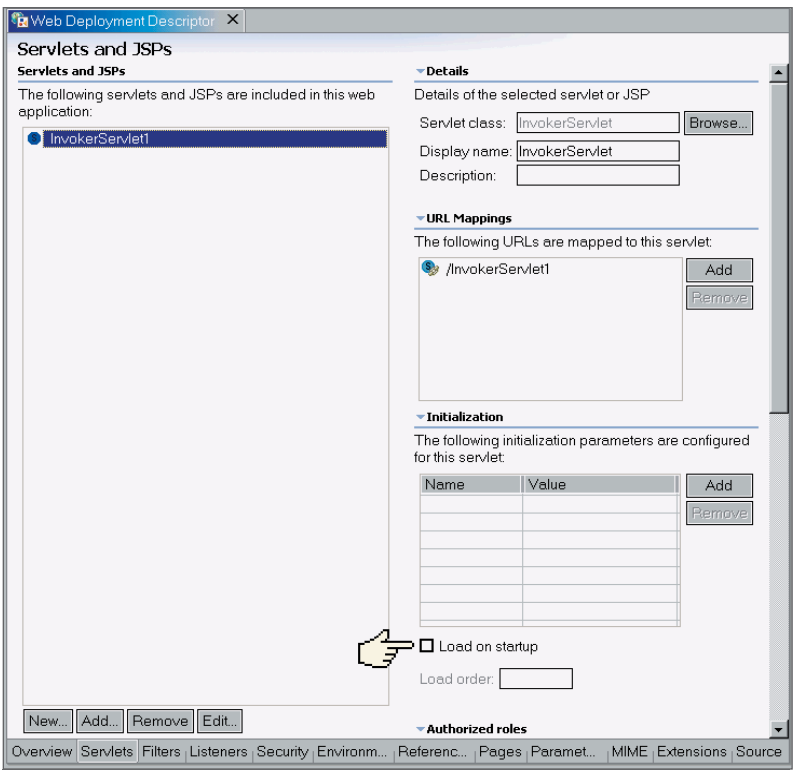


FIG. 3: MODIFYING THE WEB DEPLOYMENT DESCRIPTOR (WEB.XML FILE)

"Load on startup" option for the servlet.

Listing 3 reflects the changes necessary to implement the caching logic. Pay particular attention to the EJB home lookup code in the servlet init method of our servlet.

Now that our lookup code is in the init method, the lookup will occur only once, regardless of the number of invocations of our servlet. This can be shown by a simple inspection of our server console (see Figure 4), which shows our servlet being invoked three times. Notice the EJB home is looked up only once and is cached for future use.

Conclusion

In this article, we've studied the practice of caching EJB homes. This technique can speed up the use of EJBs in your applications. As the lookup process of EJB homes can be a time-consuming one, it makes good sense to try to store or cache the homes of the EJBs looked up, thereby looking up a given EJB's home only once.

As you have seen in this article, in a Web application using Java servlets, the init method serves as an ideal location for performing the caching of EJB home objects. However, it is a good possibility that you are dealing with a more complex setup in which multiple servlets are trying to access the same EJBs. In such a setup, you should consider leveraging a singleton design pattern. We'll do this in an upcoming sequel to this article.

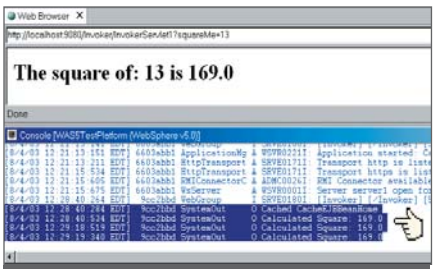


FIG. 4: ALTHOUGH THE SERVLET IS INVOKED THREE TIMES, THE EJB HOME METHOD IS LOOKED UP ONLY ONCE

LISTING 1

```
package com.ibm.simple.ejbcacheexample;
/**
 * Bean implementation class for Enterprise Bean: CacheEJB
 */
public class CacheEJBBean implements javax.ejb.SessionBean
{
   private javax.ejb.SessionContext mySessionCtx;
   /**
    * getSessionContext
    */
   public javax.ejb.SessionContext getSessionContext() {
          return mySessionCtx;
   }
   /**
    * setSessionContext
    */
   public void setSessionContext(javax.ejb.SessionContext
   ctx) {
          mySessionCtx = ctx;
   }
   /**
    * ejbCreate
    */
   public void ejbCreate() throws javax.ejb.CreateException
{
   }
   /**
    * ejbActivate
    */
   public void ejbActivate() {
   }
   /**
    * ejbPassivate
    */
   public void ejbPassivate() {
   }
   /**
    * ejbRemove
    */
   public void ejbRemove() {
   }

   /**
    * Simple method returns the square of a given float
    */

   public float square(float squareMe)
   {
          return squareMe*squareMe;
   }

}
```

LISTING 2

```
public void doGet(HttpServletRequest req, HttpServletResponse
resp)throws ServletException, IOException {
   resp.setContentType("text/html");
   PrintWriter out = resp.getWriter();
   out.println("<HTML>\n"
          + "<HEAD><TITLE>EJB Invoker</TITLE></HEAD>\n"
          + "<BODY>\n"
          + "<H1>The square of: ");
   String squareMe = req.getParameter("squareMe");
   out.println(squareMe);
   javax.naming.Context ctx = null;
   java.util.Hashtable env = new Hashtable();
   env.put(Context.INITIAL_CONTEXT_FACTORY,"com.ibm.web-
   sphere.naming.WsnInitialContextFactory");
   try {
          ctx = new InitialContext(env);
          Object homeObject =
   ctx.lookup("ejb/com/ibm/ejbcacheexample/CacheEJBHome");
          System.out.println("Looking up EJB");
          ejbHome =
                    (CacheEJBHome)
                    javax.rmi.PortableRemoteObject.narrow(
                              homeObject,
                              CacheEJBHome.class);
                    CacheEJB cacheEJB = ejbHome.create();
                    float whatToSquare =
                    Float.parseFloat(req.getParameter("square
                    Me"));
                    float squared =
                    cacheEJB.square(whatToSquare);
                    System.out.println("Calculated Square: "
                    + squared);
                    out.println("is " + squared);
          } catch (CreateException ce) {
                    ce.printStackTrace();
          } catch (NamingException ne) {
                    ne.printStackTrace();
          }
          out.println("</BODY></HTML>");
   }
```

LISTING 3

```
import java.io.IOException;
import java.io.PrintWriter;
import java.util.Hashtable;

import javax.ejb.CreateException;
import javax.naming.Context;
import javax.naming.InitialContext;
import javax.naming.NamingException;
import javax.servlet.ServletConfig;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.ibm.ejbcacheexample.CacheEJB;
import com.ibm.ejbcacheexample.CacheEJBHome;

/**
 * @version         1.0
 * @author
 */
public class InvokerServlet extends HttpServlet {
   /**
   * @see javax.servlet.http.HttpServlet#void
     (javax.servlet.http.HttpServletRequest,
     javax.servlet.http.HttpServletResponse)
   */
   CacheEJBHome ejbHome = null;
   public void init(ServletConfig config) {
          try {
                    super.init(config);
                    javax.naming.Context ctx = null;
                    java.util.Hashtable env = new
                    Hashtable();
                    env.put(
                              Context.INITIAL_CONTEXT_FACTO-
                              RY,

"com.ibm.websphere.naming.WsnInitialContextFactory");
                    try {

                              ctx = new InitialContext(env);
                              Object homeObject =
     ctx.lookup("ejb/com/ibm/ejbcacheexample/CacheEJBHome");
                              ejbHome =
                                        (CacheEJBHome)
                                        javax.rmi.PortableRem
                                        oteObject.narrow(
                                                  homeObject,

CacheEJBHome.class);
                              System.out.println("Cached
CacheEJBBeanHome");

                    } catch (Exception e) {
                              e.printStackTrace();
                    }
          } catch (ServletException se) {
                    se.printStackTrace();
          }
   }

   public void doGet(HttpServletRequest req,
   HttpServletResponse resp)
          throws ServletException, IOException {
          resp.setContentType("text/html");
          PrintWriter out = resp.getWriter();
          out.println(
                    "<HTML>\n"
                              + "<HEAD><TITLE>EJB
                              Invoker</TITLE></HEAD>\n"
                              + "<BODY>\n"
                              + "<H1>The square of: ");
          String squareMe = req.getParameter("squareMe");
          out.println(squareMe);

          javax.naming.Context ctx = null;
          java.util.Hashtable env = new Hashtable();
          env.put(
                    Context.INITIAL_CONTEXT_FACTORY,
                    "com.ibm.websphere.naming.WsnInitialConte
                    xtFactory");
          try {
                    CacheEJB cacheEJB = ejbHome.create();
                    float whatToSquare =
                    Float.parseFloat(req.getParameter("square
                    Me"));
                    float squared =
                    cacheEJB.square(whatToSquare);
                    System.out.println("Calculated Square: "
+ squared);
                    out.println("is " + squared);
          } catch (CreateException ce) {
                    ce.printStackTrace();
          }
          out.println("</BODY></HTML>");
   }

   /**
   * @see javax.servlet.http.HttpServlet#void
     (javax.servlet.http.HttpServletRequest,
     javax.servlet.http.HttpServletResponse)
   */
   public void doPost(HttpServletRequest req,
   HttpServletResponse resp)
          throws ServletException, IOException {
          doGet(req, resp);
   }

}
```

ROB **HIGH,**
IBM DISTINGUISHED ENGINEER AND CHIEF ARCHITECT FOR IBM WEBSPHERE APPLICATION SERVER

IBM's Rob High Talks WebSphere Security

Part 2: Layered approach thwarts attacks

INTERVIEWED BY JACK **MARTIN**

Jack Martin, editor-in-chief of *WebSphere Journal*, recently spoke with Rob High, IBM Distinguished Engineer and chief architect for IBM WebSphere Application Server. In this exclusive interview, High discusses the SMB market, computing as a utility, and WebSphere's approach to security.

WJ: Security has become a monstrous issue since September 11. Where do you see WebSphere security going now and how do you look to strengthen it?

RH: One of the key principles in securing the environment is to avoid limiting yourself to implementing security only on the perimeter of your data system. I think that one fundamental mistake that Microsoft has made in their security strategies is that they presume that they can protect at the boundary. The boundary is a sieve. There are a lot of entry points into the information system – and unless they capture every one of those, then they have left themselves vulnerable to attack.

What is essential to really protect the system is a layered approach in which you basically protect on the periphery, you protect the next tier, the next tier, and so on throughout your entire information system, so that if any vulnerabilities do come up and they do expose some portion of your computing base, then you are able to stop that at the next tier. If that turns out to be vulnerable, then you are able to stop it at the next tier. So, building in security policies on every component within the architecture of the application is an absolutely essential prerequisite to enabling the really robust security system. Limiting the complexity of the model is also a very important.

Vulnerabilities are introduced in large part when administrators don't adhere to basic processes. When the security model gets too complicated for them to understand, then they are basically going to stop performing procedures that they should be performing. Customers need a fairly uniform security model that spans all the resources of the applications in the IT data center.

We've got to remove security policy from the hands of the application developers. Especially if they are focused on the domain logic, they tend to be amateurish about their approach to security. Security policies change over time, and so policies may no longer be valid or sufficient.

So extracting security policies out of the application logic, externalizing them, and making them a declarative aspect of the application is another key principle to achieving effective protection of the information system.

WJ: Do you actually have groups that attack WebSphere and try to figure out how to get in?

RH: We have worked with ethical hackers to set up various security attacks and test them.

WJ: So you have organized attacks to try to anticipate what hackers might think of before they can actually do damage?

RH: That's exactly right. First of all, we leverage the J2EE security model, which introduces the entire security model for assigning security policies to an application component. J2EE also gives us the ability to assign security policies at every component in the application design. Second, we have built the WebSphere security infrastructure with plug points so that we can adapt in external third-party security providers. We did that specifically so that a single security provider can be employed to protect WebSphere resources, database resources, messaging resources, etc., across the entire enterprise infrastructure.

WJ: So if you had a rogue programmer working for your company, WebSphere security is constructed to detect that there might be something going wrong?

RH: Exactly. Because at best, programmers tend to be relegated to being responsible only for their piece of the application, so it somewhat limits the vulnerability to a particular area of isolation.

WJ: So "Mr. X" can't gain access to multinational companies because he is writing

ABOUT ROB HIGH

Rob High is an IBM Distinguished Engineer and the chief architect for the WebSphere Application Server product family. In his 26 years of programming experience he helped to define, and then later refine, the basic concepts of container-managed component technology, which is now intrinsic to the EJB specification and implemented by WebSphere and other J2EE application servers. Rob holds a bachelor's degree in computer and information science from the University of California at Santa Cruz.

code for their attendance system?

RH: Right. Even if he put some backdoor into the application code, when that application turns around and makes use of data from the data system, or generates messages posted through the messaging system, then those layers of the application design can catch those attempts.

WJ: And that stops it?

RH: That's right.

WJ: Does it alert the enterprise, or just stop it, or is it a combination?

RH: It's a combination. The middleware is going to enforce authorization policies. So if the policy does not allow for a rogue user to access the data of the application that it intends to use, then the data system is going to stop it because the user is not authorized.

WJ: So a rogue programmer could not write a piece of code to shut down an IT infrastructure 30 days from now?

RH: It becomes increasingly difficult. That's really the benefit of a layered security approach, because every resource manager takes responsibility for the resources they host, and independently applies access policies to those resources.

It's pushing the enforcement of the security policy down into the middleware, based on deployment policies, that allows the security administrator to have a whole lot of control over what is and is not being allowed. It takes security out of the secret protected space the programmer into oftentimes have their own auditing mechanisms. So for example, Tivoli Access Manager is a pluggable provider of WebSphere security. It audits authorization checks every time WebSphere calls it and surfaces those audit entries back up to a security administrator. Then a customer can also plug into the intrusion-detection systems, which look for patterns of use that represent a rogue attack on the system.

WJ: WebSphere – Express has been rolling for about a year now as IBM pushes deeper into the SMB market. That really translates into smaller and smaller companies – companies that don't even know what zSeries are – they are now are more of a Windows, and maybe a Linux, operation. How do you see that playing out with Linux and Windows?

RH: Our basic strategy toward reaching the SMB market space is primarily through ISVs that are already creating applications for that space.

The software we are delivering targets the ISVs that are developing applications. They are just as sophisticated as large enterprise customers; the way they build and design applications is just as distributed and leverages as many resource types as anybody else's.

What the ISVs want to do, in turn, is deliver their applications to their customers without the burden of a complicated installation infrastructure. They want to deliver that as a transparent aspect of their own application product delivery, so that WebSphere essentially becomes an

"I think that one fundamental mistake that Microsoft has made in their security strategies is that they presume that they can protect at the boundary. The boundary is a sieve"

previously used and externally exposes it to the security administrator so that he or she can audit the policy.

WJ: So the administrator can see if this program is going to a database that it shouldn't be going to?

RH: The administrator can see if it's doing that or it's trying to attach itself to resources that shouldn't be authorized, or exposing access to rogue individuals who shouldn't have access, etc. Administrators can classify the roles and define security policies that are outside of the control of the program. So it takes security out of the hands of the programmer.

WJ: It doesn't matter who writes the code?

RH: It doesn't. The security infrastructures that we plug embedded runtime that is hidden by, but that enhances the value of, the application itself.

Customers want robust applications and an infrastructure that is lightweight – nonimpacting – yet that is qualitatively good enough to satisfy their needs. They also prefer that their applications not be bound to one platform. They want to write their applications and simply gain the benefit of the collaboration between WebSphere and the platforms they have installed.

WJ: Can you give us your view of computing as a utility?

RH: The utility model – from a WebSphere standpoint – means a couple of things. One, of course, is exactly everything that we have been talking about – that is, providing an opportunity for people to deploy an application on whatever platform the utility environment chooses to use.

They want to use a mixed environment and do that without suffering the consequences of having to migrate all their applications from one platform to another.

It also means being able to provision applications onto the WebSphere environment. So there has to be a remote provisioning capability for allowing an ISV to say, "I want a new instance of this application for this customer and to let them contribute to the administration of their own application." If the customer then wants to register users of their office environment for that application, they need to be able to do so remotely, and do so as a proxy to the hosting environment.

They certainly don't want a failure of some other customer's application instance to roll over and take up their own resources. They want to be assured that any business information is not somehow bleeding out to other customers in a hosting environment.

They want to be assured that they will get the response time they need, when they need it. So from that service standpoint, we build in server mechanisms that deploy an application with multiple versions. We are able to customize each of those versions to each individual customer to integrate with provisioning services. They, in turn, know how to provide remote administration and provide a remote portal for things like adding and registering users.

That still gives that central utility the opportunity to manage all those applications and understand how that aggregation of application instances is exploiting the available capacity of the hosting environment.

They can trade off between the service-level agreements they have formed with customers, including with customers that may be paying a different premium. And they can add capacity without ever having to bring down the applications – or vice versa, to retract capacity as demand is decreased.

WJ: Like for a retailer after the white sale in January?

RH: You got it. So, from the application server standpoint, that means that we leverage some of the things that are already supported in WAS 5.0 in terms of externalizing control of integration and regarding remote administration support. Both central IT organizations and remote customers can monitor the relative utilization and performance of various application servers.

So, as far as I am concerned, we have built a very solid foundation. We work closely with other members of the IBM middleware family – Tivoli, DB2, Rational, Lotus Collaboration Services technology. So you may have J2EE applications that are hosted on WebSphere making use of DB2, making use of MQ messages, integrating with user collaboration services, driving business activity events back to a set of dashboards, etc.

All of this puts demands on the infrastructure that need to be balanced against a set of service-level agreements. Tivoli will provide provisioning services for monitoring the system, and for driving back workload management directives to the various middlewares to ensure service-level objectives are being met. Each of the middleware systems need to understand where the boundaries lie between hosted customers, and use that understanding to ensure complete isolation between them – in terms of system utilization as well as data integrity and to avoid cascading failures.

WJ: Right.

RH: Then there's the huge variety of ISV middleware out there, with SAP, PeopleSoft, and Siebel basically providing infrastructures for executing their applications. So all of that needs to be aggregated and delivered as a capability to anybody who wants to get into utility computing.

"Building in security policies on every component within the architecture of the application is an absolutely essential prerequisite to enabling the really robust security system"

WJ: Where do you see the first practical application of grid computing taking place?

RH: Right in the data center.

WJ: What is that used for?

RH: Grid computing refers to a fairly broad spectrum of capacity concerns. Simple clustering is a form of grid computing in which we are redirecting and routing workload to process the cluster, based on utilization and capacity that may vary throughout the day.

Today, customers have to spend a fair amount of their resources and time just rebalancing their system, adding additional capacity, and renewing capacity. They regularly go back and rebalance their systems, based on what they have experienced, as well as what they anticipate in terms of capacity and utilization demand.

We're helping customers dynamically allocate server usage. We say, "Give me the hundred servers you already have deployed in your IT data center and rather than preallocate them to individual applications, instead pool them together, leverage that capacity across all of the applications you deploy in your center." And then we balance the capacity of the pool to the applications that need it, when they need it, based on real and projected demands.

MQSoftware™
SEE IT WORK.™

8,557 orders

flowed through your system flawlessly today.

1 didn't.

Unfortunately, one failed transaction a day

will cost you **$4.3 million** this year.

– If only you saw it sooner.

.:::.:.:: Real-time monitoring of all your critical business transactions.
:.:.:.:.:.:. Because finding out tomorrow might be too late.

MQSoftware
Q Nami!™

IBM Business Partner

To learn more about Q Nami! and the benefits it brings to your business, register to attend a free online Web seminar.
Visit **www.mqsoftware.com/qnami** for seminar dates, or call **800.998.2858** for details.

J2EE pattern frameworks provide template for flexible and modular architecture

Creating a Framework

BY LLOYD HAGEMO & RAVI KALIDINDI

Many patterns have been published for J2EE applications. By developing and connecting multiple patterns, developers can create a framework that improves the stability, performance, and scalability of their J2EE application architectures. Because the number of patterns continues to expand, it can be difficult for developers to select the best combination of patterns to create frameworks that optimize J2EE applications and fulfill specific IT or business requirements.

ABOUT THE AUTHOR

Lloyd Hagemo is a senior director for Candle Corporation's Application Infrastructure Management Group. He is responsible for WebSphere tools development. Lloyd has led the successful development of more than 20 products for the WebSphere environment, including operating system utilities, network performance and tuning products, WebSphere MQ configuration and management tools, and application integration solutions.

E-MAIL

lloyd_hagemo@candle.com

Similar to individual patterns, frameworks serve as development process templates that enable organizations to streamline development while ensuring high performance levels. The ability to create a solid blueprint is critical to J2EE application development success. Industry analyst Giga Information Group, Inc., reports that up to 75% of all application development projects fail due to inadequate planning, tools, training, or related factors.

This column includes a summary of common J2EE patterns to use as a guide when developing applications. Each pattern is typically implemented in each individual class. We have used the term "pattern" throughout the column to minimize any potential confusion between "pattern" and "class." The column also highlights four pattern frameworks designed to manage specific J2EE functions.

Summary of J2EE Patterns

To simplify the task of identifying appropriate patterns for IT and business requirements, a list of common J2EE patterns follows. The patterns are segmented according to the following industry-accepted and functionality-based categories created by Sun Microsystems, Inc.

- ***Presentation tier patterns:*** Contain patterns related to presentation components
- ***Business tier patterns:*** Contain patterns related to business components
- ***Data/integration tier patterns:*** Contain patterns related to integration or data resources

Table 1 summarizes the basic patterns that are used in the frameworks.

Connecting J2EE Patterns

Frameworks can be designed with various combinations of patterns to address application requirements. Following are four pattern frameworks for the most common types of applications used in J2EE environments:

1. Web container pattern framework
2. Command pattern framework
3. Session bean pattern framework
4. Entity bean pattern framework

The first two pattern frameworks are designed for simple Web applica-

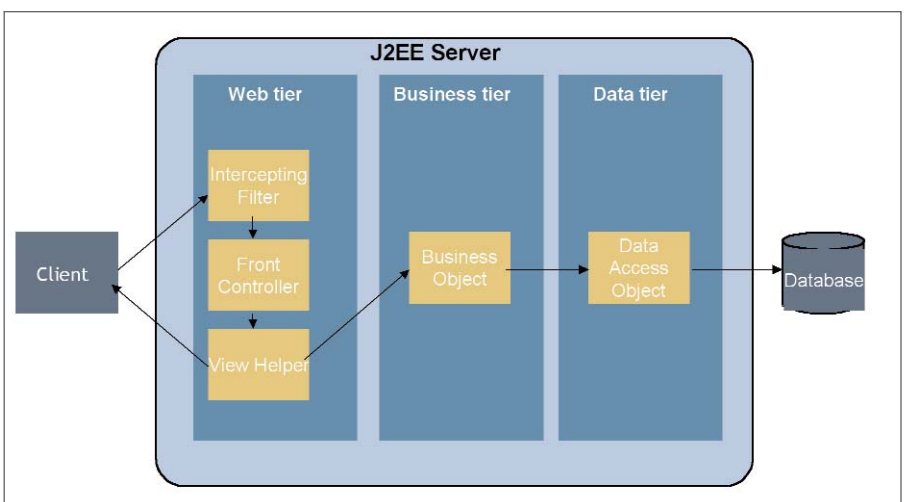


FIG. 1: THE WEB CONTAINER PATTERN FRAMEWORK

tions developed with servlets and JSPs deployed in a Web container. The third and fourth pattern frameworks are used in Web and EJB containers.

The Web Container Pattern Framework

This architecture is used for database access from a Web container. The Intercepting Filter is the first point of contact for all client requests. It is used to control access and authentication. Once requests are received and filtered using the Intercepting Filter, they are forwarded to the Front Controller, which is the central dispatcher for the application. The Front Controller dispatches the request to the appropriate View Helper. A View Helper is a set of helper functions used to collect specific data. The Composite View is used to build the response to the client, and is also responsible for calling the appropriate Business Object, which calls the Data Access Object pattern to get the related data. This framework is shown in Figure 1.

Figure 1 outlines how the Web Container pattern framework is used for an inventory requirement. If a client requests a list of inventory, the Intercepting Filter checks for authentication and forwards the request to the Inventory Controller (Front Controller), which in turn dispatches the request to the Inventory Helper (View Helper). The Inventory Helper can be a normal JavaBean or a bean that will talk to the Inventory Business Object to get the appropriate inventory data. The Inventory Business Object contains business logic to process the inventory and asks the Inventory Data Access Object (DAO) to get the data from the database. The Inventory DAO contains database communication and SQL code to get the inventory-related data. This simple example shows how you can apply this pattern framework easily for inventory functionality. The advantage is that the application architecture is well documented and can be easily understood.

The Command Pattern Framework

The second framework is similar to the Web Container pattern framework, but uses the Command pattern, as shown in Figure 2. The Command pattern provides decoupling, which can make it easier to add more functionality to an application in the future. The controller requests the command, gets the results, and forwards them to the Composite View.

If you apply this framework to the Inventory example, there would be an Inventory Command pattern that contains business logic to process the inventory request. The Inventory Command pattern calls the Inventory DAO to get the related data. When using the command it is an advantage to use the Command Helper and Command Factory patterns to call and create appropriate commands.

The Session Bean Pattern Framework

The third framework adds Session Bean patterns to the Web Container pattern framework, as shown in Figure 3. While this framework adds more complexity to the architecture, using this pattern-oriented architecture allows greater flexibility for adding features to the application. Session Bean patterns are used to simplify the J2EE container development when introducing session beans in an application. The Command pattern framework can also be used instead of the Web Container pattern framework, depending on flexibility and maintenance requirements. As a Web site becomes more complex and requires handling of multiple transactions, the introduction of a command factory provides the application developer with a single location to modify existing

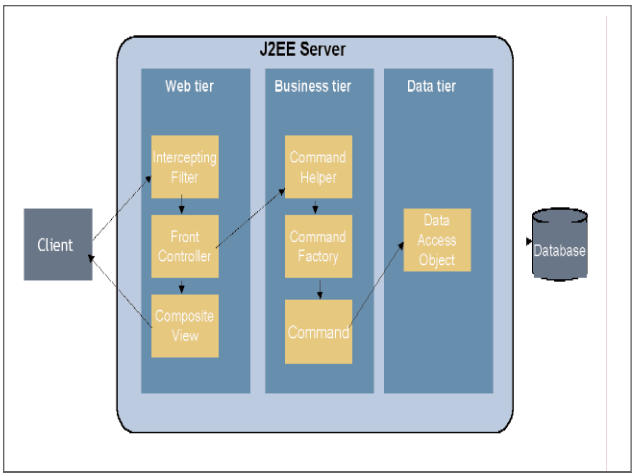


FIG. 2: THE COMMAND PATTERN FRAMEWORK

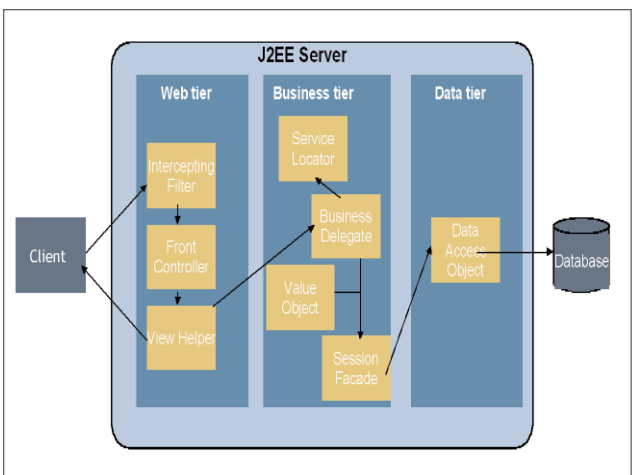


FIG. 3: THE SESSION BEAN PATTERN FRAMEWORK

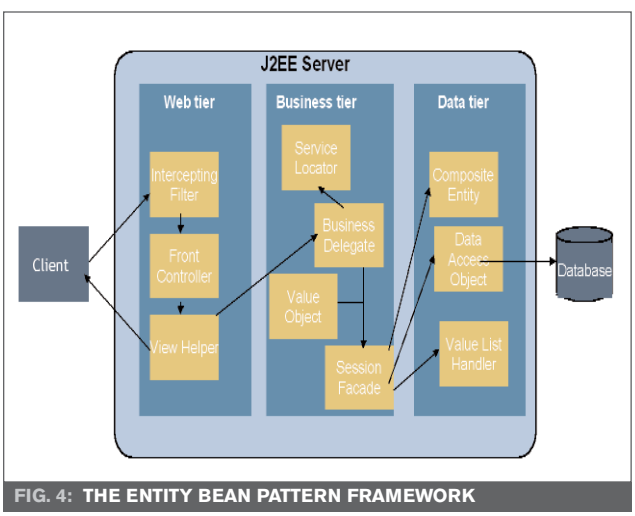
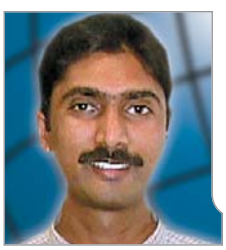


FIG. 4: THE ENTITY BEAN PATTERN FRAMEWORK



ABOUT THE AUTHOR

Ravi Kalidindi is a senior software engineer in Candle's Application Infrastructure Management Group. Ravi has worked with Java since its inception and has published several articles that focus on J2EE best practices.

E-MAIL
ravi_kalidindi
@candle.com

PRESENTATION TIER PATTERNS

Intercepting Filter	Allows handling of non-application specific services such as logging and authentication by encapsulating intercepting logic for requests and responses. An intercepting filter is typically implemented as a pluggable filter using servlet filters.
Front Controller	Encapsulates all request and response control logic and avoids duplicate control logic in multiple JavaServer Pages (JSPs). One can have multiple controllers depending on application functionality.
View Helper	Separates processing logic from JSPs; puts processing logic in a reusable helper class.
Composite View	Aggregates multiple subviews (e.g., header, footer, navigator, and body views) into a single composite view. It provides visibility into the View Helper pattern.

BUSINESS TIER PATTERNS

Business Delegate	Decouples presentation and business tiers; also encapsulates business tier access facility and hides those details from presentation-tier classes.
Business Object	Encapsulates business logic and business data.
Command	Provides a common interface to the clients and decouples presentation and business tiers.
Service Locator	Encapsulates Java Naming and Directory Interface (JNDI) lookup code to look up EJB homes and other JNDI-registered services. It can cache these services to avoid redundant JNDI calls.
Session Facade	Encapsulates core application business logic by calling multiple data sources or entity beans; also acts as an entity bean wrapper to avoid fine-grained method calls.
Value Object	Holds the data that needs to be exchanged between clients and remote EJBs. Without the value object, clients have to make multiple method calls to get the required data.
Value Object Assembler	Makes a composite value object by accessing different business components, which allows clients to get a composite value object in a single method call.
Value List Handler	Iterates through a large result set and gives the client a small set of results upon each request. It can cache the subset of a large result set to improve performance.

DATA/INTEGRATION TIER PATTERNS

Data Access Object	Encapsulates relational database management system access code and other data sources, such as object-oriented databases.
Composite entity	Avoids inter-entity bean communication and fine-grained entity beans. A composite entity bean represents a coarse-grained entity bean by having a plain Java object as the dependent object instead of another entity bean.

TABLE 1: AN APPLICATION'S REQUIREMENTS FOR AVAILABILITY IMPACT THE DESIGN, PROCESSES, AND COST OF OWNERSHIP

features and introduce new functions. This allows features to be incrementally added to Web sites with little effect on the existing function.

The Business Delegate, Session Facade, Service Locator, and Value Object patterns are introduced in this framework. The View Helper pattern calls the Business Delegate pattern to delegate the request to the Session Facade pattern. The Business Delegate pattern is responsible for looking up the Session Facade pattern using the Service Locator pattern. The data is transferred from the Business Delegate pattern to the Session Facade pattern using the Value Object pattern. The Session Facade pattern calls the DAO to get the data from the persistent store. Applying this framework to the Inventory example, there will be Inventory Delegate, Inventory Facade, Service Locator, and Inventory Value Object pattern.

The Session Bean pattern framework takes full advantage of the capabilities provided by Web and J2EE containers. It provides a set of common functions that relieve the programmer of the responsibility of looking up and calling specific EJBs. Instead, this pattern framework includes a set of commonly used design patterns that make it easier to develop EJB-based solutions.

The Entity Bean Pattern Framework

This framework (see Figure 4) adds entity bean patterns to the Session Bean pattern framework. The Entity Beans pattern can be used, as shown here, when choosing entity beans in an application. The general recommendation for entity beans is to use container-managed entity beans. This process removes the SQL from the Java code and places it in the XML definitions. The coding of SQL in XML takes time to learn and understand. This pattern also includes the Value List Handler pattern, which is very important when dealing with large database records.

We introduce the Composite Entity pattern and the Value List Handler pattern in this framework to improve the performance of the application. In this framework, the Session Facade pattern calls composite entities and/or DAOs. When dealing with large database results, the framework uses the Value List Handler pattern to send a small set of processed results iteratively to the client. The implementation of the Value List Handler framework is done using a normal Java class that contains the SQL. For example, when inventory functionality is implemented, you can have Inventory Entity and All Inventory Handler patterns. The Inventory Entity pattern acts as a Composite Entity pattern and the All Inventory Handler pattern acts as a Value List Handler pattern to deal with a large inventory search.

Conclusion

Pattern frameworks enable developers to achieve highly functional, modular architectures for their applications. They also enable developers to implement pattern-oriented architectures that clearly identify where requirements can be implemented. We have presented four types of frameworks by connecting different patterns depending on two common types of J2EE applications. We encourage developers to leverage the power of pattern frameworks by taking the examples presented here as a base and extending them as needed, or by building new pattern frameworks that meet unique application requirements.

What Will 2004 Bring for IT?

Looking Ahead

WebSphere Journal exclusive: Industry experts look at the coming year

A sampling of industry experts offer their thoughts on what the coming year will bring for the IT industry in general – and for WebSphere in particular.

WALTER HURST
CTO
WAKESOFT

- ***On offshoring IT:*** There will be a "levelling off" in offshoring as companies determine through trial and error where it works and where it doesn't. For new system development, there will be backlash as enterprises get frustrated with failed projects and realize that it is the same as outsourcing with all the related problems. Large onshore SIs (Accenture, etc.) will have the expertise to get it right, and large enterprises will go to those trusted vendors (who will pass on some of the savings because they've been threatened by customers going directly to offshore companies).

- ***On standards:*** Consolidation will continue in middleware, with the platforms being IBM versus Oracle versus Microsoft versus whoever acquires BEA. The marketplace won't be pure J2EE versus .NET, but corporate competition between the big companies with those companies doing whatever it takes to compete (including continued diminishing relevance of JCP). When competing with Microsoft (which is the market that everybody wants), you have to adopt Microsoft's successful tactics (build first, standardize later).

- ***On service-oriented architecture:*** Large enterprises will like solutions delivered as application service providers (SalesForce versus Siebel). This will go hand in hand with their adoption of service-oriented architectures that allow them to customize and extend the ASP solutions for specific needs.

DAVID CADDIS
VICE PRESIDENT, APPLICATION INFRASTRUCTURE MANAGEMENT GROUP
CANDLE CORPORATION

Many organizations have not realized the value of their IBM WebSphere investments because they have not developed the right blueprints for success. Architecting J2EE to support industrial-strength business requirements is a complex undertaking. An effective runtime environment must optimize each system within an infrastructure and address how the interrelated technologies impact one another. When architecting e-business environments, the performance of the entire infrastructure is greater than the sum of its parts. Many J2EE blueprints have addressed only front-end application considerations, which is like designing a car without including a radiator to prevent the engine from overheating.

Failure to design for performance, scalability, and flexibility across the entire enterprise has caused many applications to either perform poorly or remain stalled in preproduction. To mitigate these problems, 2004 will mark a refocus on architecting enterprise Java applications and the underlying transaction processes that serve as the highways for business operations. A comprehensive J2EE blueprint will enable organizations to sidestep performance hurdles and quickly deploy new business applications.

SUNNY GUPTA
SENIOR DIRECTOR, J2EE PRODUCTS
MERCURY INTERACTIVE

Many IT organizations have made significant investments in building new applications, integrating existing applications, and customizing package applications leveraging the WebSphere platform. However, less than 20 percent of these organizations are getting what they need from these applications – 99 percent availability – to provide value to the business. Almost half the time, availability and performance issues get noticed only when customers call to complain that they cannot

complete a business process. In other words, once revenue and productivity have already been lost.

In the year ahead, we believe that more IT organizations will adopt a business technology optimization (BTO) approach to maximize the business value of their WebSphere applications. With BTO, companies can focus resources on application delivery and application management to optimize the quality, performance, and availability of their WebSphere applications, get more out of their existing investments, and control IT costs.

NICOLAS JABBOUR
CTO
PROLIFICS

WebSphere is becoming a commodity, as are most of the J2EE application servers. Their value is no longer based upon what they do but instead upon how they do things. I suggest that support for the latest and greatest J2EE standard will be taken for granted. Instead, the enhancements that complement the J2EE standards, and the overall enrichment of the application server with value-add offerings (e.g., Rational Suite, WSAD, Portal, Pervasive, etc.) will make the significant difference. In fact, we have seen this played out in history before, and it was successful for both IBM and Microsoft. When you own the infrastructure you can sell more value-add software. Once that value-add software becomes recognized for its own merits the process reverses and the value-add software drives more infrastructure software sales. IBM's model with the legacy operating systems (VM, MVS, etc.) and Microsoft's model with MS Office are brilliant examples of this.

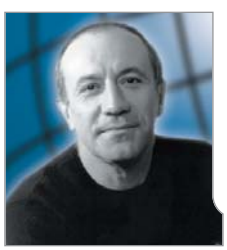
Some other predictions:

- The application server space will evolve into an integration server space. JMS, JCA, and the inclusion of the Web services standard into the J2EE spec are a few examples of this evolution.
- Coming soon are the days when application servers will be distributed with the operating systems; just follow the history.

KURT ZIEGLER
SENIOR VP, PRODUCT MANAGEMENT AND MARKETING
DIRIG SOFTWARE

In 2004, change will be an ever-increasing factor for production WebSphere environments. WebSphere professionals are still struggling to see the performance and impact of the various components affecting the service level along the paths of their business applications as they traverse the complex of heterogeneous, distributed, multivendor computing technologies to serve the end users.

But this year, WebSphere professionals need to find more effective ways to deal with the faster rate of change, in addition to establishing and maintaining application- and transaction-specific service levels. This will help them identify and monitor the transaction-specific paths, enabling them to detect and proactively eliminate availability exposures and friction, as well as provide sufficient application and environmental forensics to ensure rapid problem resolution or circumvention. More important, these new approaches will enable IT management to set business-oriented priorities for service levels and the use of resources.



ADAM KOLAWA
CEO/CHAIRMAN
PARASOFT

Within the next year, I expect that many companies using outsourcing will start to abandon their outsourcing efforts because they are not providing the anticipated level of cost reduction. This change will come as companies realize that the quality of code developed offshore is not improving and they begin to understand that it will cost them just as much to fix the outsourced code as it would cost them to have their own developers produce it from scratch. However, outsourcing will prove effective for some companies, and those companies will continue to outsource.

WebSphere Journal Advertiser Index...

COMPANY	URL	PHONE	PG
BOWSTREET	WWW.BOWSTREET.COM/EASY	978-863-1500	15
CYANEA	WWW.CYANEA.COM/WSDJ/NOMOREMADNESS/HTML	877-CYANEA8	11
EDGE EAST 2004	WWW.SYS-CON.COM	201-802-3069	41-56
H&W COMPUTER SYSTEMS	WWW.HWCS.COM/WSD1.ASP	800-338-6692	60
KENETIKS	WWW.KENETIKS.COM	888-KENETIKS	29
MERCURY INTERACTIVE	WWW.MERCURYINTERACTIVE.COM/OPTIMIZEJ2EE	800-837-8911	9
MQSOFTWARE	WWW.MQSOFTWARE.COM/QNAMI	800-998-2858	19-21
NETIQ	WWW.NETIQ.COM/SOLUTIONS/WEB	408-856-3000	3
PROLIFICS	WWW.PROLIFICS.COM/WEBSERVICES	800-675-5419	2
QUEST SOFTWARE	HTTP://JAVA.QUEST.COM/JPROBE.WDJ	800-663-4723	59
LINUXWORLD CONFERENCE & EXPO	WWW.LINUXWORLDEXPO.COM	800-657-1474	30-31
VERSATA	WWW.VERSATA.COM/BUSINESSLOGICDESIGNER		57
WILY TECHNOLOGY	WWW.WILYTECH.COM	888-GET-WILY	5

WebSphere Journal Coming Next Month...

INTERVIEW
Craig Hayman Discusses IBM Pervasive Computing Products and Success Stories
INTERVIEWED BY JACK **MARTIN**

GUI DEVELOPMENT
Using Java Web Start and WebSphere Studio to Create Rich Client Apps
BY JOE **WINCHESTER**

EAI
Improving Productivity with Enterprise Application Management
BY DANIEL **MAGID**

ILLUMINATING WEBSPHERE
Security: Incorporating the Power of Attack Trees
BY MICHAEL S. **PALLOS**

Top five rules for Notes/Domino developers moving to WebSphere

Writing Professional-Quality Java Code

BY MARK **DIXON**

ABOUT THE AUTHOR

Mark Dixon is cofounder and CTO of Teamstudio. He is responsible for all aspects of product strategy, architecture, and development, including management of the development, quality assurance, and documentation teams. Mark has architected the full suite of Teamstudio products. Prior to joining Teamstudio, he was a senior consultant at Ives & Company. He earned his master's degree in mathematics from Cambridge University, England, and followed it with postgraduate research in applied probability.

E-MAIL

Mark_Dixon@Teamstudio.com

It's no revolutionary statement to say that learning to program in a new language involves more than just learning the syntax. It does resonate, however, with those of you who have just been through learning to code in a new language, are currently learning a new language, or are thinking about investigating a new language. You're probably still poring over how you might have done this piece or that piece of the application differently, or you're wondering how to tackle a particular part of your program.

For Notes/Domino developers, the challenge isn't any more or less. However, given the perceived instability of a long-term career as a Notes developer, you've got added pressure from the questions of what you'll do next and how quickly you can become a pro.

For some of you, it means you're looking at making the transition from Domino to WebSphere because IBM has been your mainstay for Lotus Notes and Domino. You may have even downloaded the open source Eclipse IDE and started writing some Java code. Good for you! This article will hopefully reinforce what you're learning along the way.

What I'll do here is outline the top five rules you can follow to avoid the common rookie coding mistakes and produce professional-quality code from Day One.

Before we dive into the rules, I want to make a note about Java and LotusScript. They have a lot in common, but there are a few areas where their behavior differs significantly. Throughout this article, whenever you see "LotusScript Tip," you'll find an area that may be particularly troublesome for you if you are an experienced LotusScript developer. Although a strong LotusScript background will prepare you well for getting up to speed with Java, it's important to note these potential sandtraps.

Now, on with the top five rules for Notes and Domino developers moving to WebSphere. Counting down from No. 5:

- ***Rule 5: Make all fields private and use get/set methods.*** This is sound object-oriented design and applies as much to LotusScript as it does to Java. However, it is a key rule and it bears repeating.

Each class is a combination of two elements: the interface that is exposed to callers of the class – and the implementation – the actual code that performs the functions of the class. One of your goals when designing a class should be to keep these two as separate as possible. The more of your implementation you expose, the harder it is to maintain your class going forward. Java helps with this by allowing you to mark fields and methods within a class as private. Class members that are private are not visible to any other class and thus can be freely changed as your class evolves. Any members that are not private, even if they have limited access, such as protected or package members, can still be accessed by other classes and are part of your class's interface.

Any field you create that isn't private is thus part of your class's interface and cannot be changed without potentially having an impact on other classes. This greatly limits your options for enhancing your class over time. By making the field private and providing access via get/set functions, you leave the door open for changing, or even completely replacing, your internal data representation.

Of course, the accessor functions still impose some limitations on you – for example, you can't change the return type of a get method without breaking clients. For that reason you should avoid the habit of automatically providing accessor functions for all private fields. If you can have an object manipulate its fields itself, rather than expose them, then you allow maximum flexibility for enhancing the implementation. However, this is not always possible – and when you do have to expose data held within an object, accessor methods are the way to go.

LotusScript Tip: This is equivalent to the use of properties in LotusScript. Java doesn't have language support for properties, but

instead uses a naming convention. The set function for a property named "myProperty" would be called setMyProperty; the get function would be called either getMyProperty, or, if the property is Boolean, isMyProperty. As with LotusScript, you don't have to provide both get and set methods for a property and, although properties will often correspond to private fields in the class, there is no reason why they have to do so.

Example
Bad:

```
public class Employee {
  // Bad example –don't use public
  // fields
  public String firstName;
  public String lastName;
}
```

Good:

```
public class Employee {
  // Good example – fields are
  // private and are exposed via
  // methods
     private String firstName;
     private String lastName;

  public String getFirstName() {
    return firstName;
  }

  public void setFirstName(String
  firstName) {
    this.firstName = firstName;
  }

  // similarly for lastName
}
```

- ***Rule 4: Use equals() rather than == to compare objects.*** What does the following Java code snippet print?

```
String s = "test";
String t = "TEST";

if (s.toUpperCase() == t) {
  System.out.println("Match");
}
else {
  System.out.println("No Match");
}
```

"Match", right? Wrong. In Java, the equality operator, ==, compares objects rather than values. Both t and s.toUpperCase() are string objects with the value "TEST", but they are distinct objects and thus == reports that they are not equal.

You can fix the problem by training yourself to ask – every time you write in Java – whether you really mean object equality. If you don't, then you should use the equals function instead, which performs a value comparison.

LotusScript Tip: It is particularly easy to make this mistake with strings in Java because LotusScript overloads the equality operator to automatically perform string comparisons when appropriate.

Corrected code:

```
String s = "test";
String t = "TEST";

if (s.toUpperCase().equals(t)) {
  System.out.println("Match");
}
else {
  System.out.println("No Match");
}
```

- ***Rule 3: Always use break with switch statements.*** The switch statement in Java is similar to LotusScript's Select Case, but with one key difference. Java's case clauses don't allow you to list multiple values like you can in LotusScript; instead, program execution flows automatically from one case to the next until you exit the switch with a break, continue, return, or throw statement. So, for example, these two samples of code are equivalent.

Java:

```
switch (i) {
  case 3:
  case 4:
    System.out.println("3 or 4");
    break;
  default:
    System.out.println("Not 3 or
     4");
}
```

LotusScript:

```
Select Case I
  Case 3, 4
    Print "3 or 4"
  Case Else
    Print "Not 3 or 4"
End Select
```

That's a reasonable alternative, but what does the following code print when i is 3?

```
switch (i) {
  case 3:
    System.out.println("3");
  case 4:
    System.out.println("4");
}
```

If you guessed "3", then look again. There's no break statement at the end of the "case 3" clause, so execution will continue straight on to the next case, producing both a "3" and a "4". This is a very common coding error, and there's nothing the Java compiler can do to prevent it because much existing code relies on this behavior. You should discipline yourself to always end every case clause with a break, continue, return, or throw statement. In the rare cases in which you actually want to handle multiple cases at the same time (this is called "fallthrough" and was demonstrated in the first example), you should clearly mark the code to show that the fallthrough is intentional. For example:

```
switch (i) {
  case 3: // Fallthrough
  case 4:
    System.out.println("3 or 4");
    break;
  default:
    System.out.println("Not 3 or
4");
}
```

- ***Rule 2: Always use braces for the bodies of conditional and loop statements.*** The body of a loop or conditional statement in Java can be either a single statement or a block.

Example of single statement:

```
if (totalErrors > 0)
  System.out.println(totalErrors +
    " errors found");
```

Example of block – note the curly braces:

```
if (totalErrors > 0) {
  System.out.println(totalErrors +
    "errors found");
}
```

I strongly recommend that you always use the block version, even if the block contains only a single statement. To see why, let's look at an example. Suppose a maintenance programmer needs to add an extra statement to the body of the statement. Without the braces, it is very easy to be misled by the indentation and write:

```
if (totalErrors > 0)
  System.out.println(totalErrors +
    " errors found");
  runFailed = true;
```

Remember that the Java compiler completely ignores indentation and treats only a single statement as the body of the if statement. So to the compiler, this looks like:

```
if (totalErrors > 0)
  System.out.println(totalErrors +
    " errors found");
runFailed = true;
```

It's impossible to make the mistake with the braces there. Any extra code goes inside the braces and that's the only place it can go. Sure, typing in those braces every time takes a couple of extra keystrokes and an extra line or two in the source code, but that's a small price to pay for clearly written code.

LotusScript Tip: Why wasn't this a problem in LotusScript? LotusScript essentially always uses the block version of the if statement. If you think of the braces as corresponding to the "Then" and "End If" parts of a LotusScript if statement then you won't go wrong.

• ***Rule 1: Use the standard Java naming conventions.*** This is the single most important thing you can do make your Java code look right. Most Java code out there uses the naming conventions established by Sun from the earliest days of Java, and following the same set of rules will help your code fit in.

Naming conventions can be something of a religious issue for programmers, but the Sun convention is reasonable and so well-established that there is no reason to deviate from it.

LotusScript Tip: Identifiers (that is, names of classes, fields, variables, etc.) are not case sensitive in LotusScript. You can declare a variable as "count" and refer to it as "COUNT" and LotusScript won't complain. In Java, all identifiers are case sensitive and so "count" and "COUNT" would be two totally unrelated variables. *Bonus tip:* just because the Java compiler can easily remember the difference between "count", "Count", and "COUNT", it doesn't follow that someone reading your code can. You should avoid using identifier names that differ only in case. Table 1 shows the Java naming conventions.

"Naming conventions can be something of a religious issue for programmers, but the Sun convention is reasonable and so well-established that there is no reason to deviate from it"

ELEMENT	CONVENTION
Package	Use lowercase letters and digits only. For any package that is – or may ever be – exposed publicly, you should prefix the package name with the reversed elements of your company's domain name. For example, all code that we write here at Teamstudio is in a package with a name of the form com.teamstudio.xxx.
Class	The first letter of each word or acronym in the name, including the first, should be capitalized. Good names are FileInputStream, UrlStreamReader. Bad names include XMLDTDReader (XmlDtdReader is preferred to prevent the acronyms running into each other) and file_input_stream (use capitalization rather than underscores to separate words).
Interface	As a class. A few authors prefix interfaces with a leading "I", but that is unusual. Good: Serializable. Bad: serializable. Questionable: ISerializable.
Method	The first letter of each word or acronym in the name should be capitalized, except for the very first letter. The leading lowercase letter helps to distinguish between methods and constructors. Good: getInputStream(), getXmlDtd(). Bad: get_input_stream(), GetInputStream()
Field, Local Variable, Parameter	As a method. A few authors add leading or trailing underscores to distinguish between local variables and fields, but this is unusual and against the spirit of using readable names, so I advise against this.
Constant	Constant names should consist of only uppercase letters and digits to clearly distinguish them from fields. If the name consists of more than one word, use underscores to separate the words. Good: MAX_SIZE; Bad: maxSize.

TABLE 1: JAVA NAMING CONVENTIONS

Conclusion

This article has been a whirlwind tour through many of the techniques that experienced Java developers use to produce great code. These five rules are a starting point for those of you dabbling with WebSphere and Java – or those looking for a refresher – to help ensure your Java coding standards are clear and concise, and that your Java code is professional grade and reliable. Certainly, this isn't an exhaustive list; there are other guidelines to consider, but if you start with these, then you'll be on your way to writing like a pro.

Resources

- Campione, M., et al. (2000). *The Java Tutorial: A Short Course on the Basics (3rd Edition)*. Addison-Wesley.
- Arnold, K., et al. (2000). *The Java Programming Language (3rd Edition)*. Addison-Wesley.

LINUXWORLD® CONFERENCE & EXPO

Where Open Minds Meet

Keynote Speakers

Jack L. Messman
Chairman of the Board & CEO, Novell, Inc.
Wednesday, January 21
9:15 am – 10:00 am

Dave Dargo
Vice President, Linux Program Office, Oracle
Wednesday, January 21
11:45 am – 12:30 pm

Tom Killalea
Vice President of Infrastructure, Amazon.com
Wednesday, January 21
2:45 pm – 3:30 pm

Sam Greenblatt
Senior Vice President and Chief Architect, Computer Associates Intl.
Thursday, January 22
11:30 am – 12:15 am

Ross A. Mauri
General Manager, e-Business on demand, IBM Systems Group
Thursday, January 22
1:45 pm – 2:30 pm

CONFERENCE: January 20 – 23, 2004

EXPO: January 21 – 23, 2004

The Javits Center: New York, NY

LinuxWorld Conference & Expo is the world's leading and most comprehensive event focusing on Linux and open source solutions. This January, discover how companies across the globe have achieved higher profits and increased their productivity by utilizing Linux, the fastest growing operating system in the world.

- Get in-depth coverage on the latest Linux and open source developments and learn about innovative product solutions from the industry's top companies.
- Network with the leaders in the open source movement and discuss with your peers how to best leverage the technology for your organization.
- Participate in LinuxWorld's world-class education program and benefit from interactive training in the all-new Hands-on Labs!
- Attend the 2nd annual Linux Financial Summit and hear how Wall Street firms are leveraging Linux to achieve a lower total cost of ownership. (Sponsored by BEA)
- Discover real world practical solutions and find answers to today's most critical IT issues.

Cornerstone Sponsor

ORACLE

Platinum Sponsors

Computer Associates · IBM · intel · hp invent · Novell · Sun microsystems · suse simply change

REGISTER ONLINE WITH REFERENCE CODE: A-WDD

www.linuxworldexpo.com

IDG WORLD EXPO

Akamai teams with WebSphere to offer on-demand edge platform

At the Edge

BY JAY PARIKH

ABOUT THE AUTHOR

Jay Parikh is director of Engineering for Akamai's globally distributed computing service, EdgeComputing. He drives product direction and provides coordination and continuity in Engineering and across other organizations at Akamai. In addition to EdgeComputing, Jay has supported other Akamai customer-facing services, including Akamai's flagship content delivery service, EdgeSuite; Edge Side Includes (ESI), a markup language for dynamic assembly and delivery of Web applications at the edge; and FirstPoint, Akamai's global load-balancing service.

E-MAIL
jay@akamai.com

As more enterprises move their business to or enable their business on the Internet, Web applications have come into widespread use in many enterprise application infrastructures. The infrastructure to deliver these Web applications typically includes a wide range of technologies such as load balancers, HTTP Web servers, caching servers, messaging systems, transaction-processing monitors, application servers, and databases. A typical enterprise application infrastructure is shown in Figure 1.

As performance and geographic reach requirements expand, it becomes increasingly more difficult to scale the Web site infrastructure. IT managers must continually evaluate capacity plans to keep pace with the expected peak demand, and planning must consider events such as marketing promotions, news events, and other events that inevitably create more planning uncertainty. Errors in planning can result in overloads that can crash sites or cause unacceptably slow response times, leading to lost revenue.

Pre-provisioning extra capacity as insurance against overload is financially unacceptable for most enterprises. Ideally, enterprises want the needed resources when – and only when – they are needed; they do not want to buy extra resources that sit idle when they are not needed. "On-demand" computing provides better utilization of computing resources and represents a model in which they are brought into service as needed.

On-Demand Application Platforms

Today, on-demand computing technologies have been integrated into centralized enterprise application platforms, and generally offer enterprises improved fault tolerance and better scalability, through the use of intelligent scheduling and load balancing of application workloads. However, for many of today's Web applications, end users are inherently spread across the Internet. Congestion and failures in this end-user environment are common and because of this, centralized enterprise applications can become unreliable and their performance can become unpredictable.

One solution for enterprises is to use a distributed application delivery platform at the edge of the Internet to ensure optimal application performance and reliability. Akamai is one example of an on-demand edge application platform. Deployed at the "edge" of the network – close to user access points – it consists of nearly 15,000 servers in over 1,100 networks around the world. Enterprises can deploy applications to this distributed platform in order to avoid service bottlenecks and failures, and at the same time provide on-demand scalability, global reach, and high performance for application users. Enterprises are not limited to serving static content from edge networks; several edge networks offer enhanced services. One such capability is Edge Side Includes (ESI), which is specifically designed for dynamic content assembly.

In order to extend the enterprise J2EE programming platform to the edge network, IBM and Akamai have integrated and deployed WebSphere Application Server (WAS) version 5.0 onto the edge servers. Enterprises can execute J2EE Web tier applications in an on-demand WebSphere environment, known as Akamai EdgeComputing powered by WebSphere, and consume Internet computing resources on a pay-per-use basis.

This widely distributed application environment, shown in Figure 2, consists of an end user typically using a browser; the enterprise (origin) running business logic, legacy systems, and databases; and the edge servers running an embeddable WAS server that supports the J2EE Web application programming model.

Developing Applications for an Edge Platform

The development model remains standard J2EE for edge applications and does not require the use of any proprietary APIs; it is the deployment model that changes. If your applications generally follow J2EE component programming best practices, adapting the existing application for the edge will be easier. Akamai's edge application platform extends the WebSphere application programming platform to enable the execution of J2EE Web tier application components – JSPs, servlets, tag libraries, and JavaBeans.

Development for an edge application platform still relies on standard J2EE development tools and best practices in developing applications, but you must architect your edge-enabled application as two cooperating sub-applications: an edge-side application and an enterprise-side application.

Presentation Components on the Edge

The presentation components are the most common application components to deploy to the edge. These components access enterprise data via the Java Web services client model. Typically, the Web application will be developed using a framework based on the Model-View-Controller (MVC) architecture.

Jakarta's Struts framework, an open source framework for building Web applications based on the MVC pattern, is well suited for edge deployment. Struts provides its own Controller component via a servlet of class ActionServlet. This servlet is configured by defining mappings, ActionMappings, between the requested URI and the class name of an Action class. For the View, Struts can leverage JSPs and provides a very flexible and powerful set of JSP tag libraries that allow you to build the View portion of an MVC application without embedding Java code directly into the JSP itself. The Model component is commonly represented by JavaBeans. These Model JavaBeans may be self-contained or represent facades for other components such as JDBC or EJBs.

The View and Controller components of a Struts application are good candidates for distribution to the edge network. These components execute on the edge servers and can interact with Model components (EJBs) running at the enterprise. Depending on the functionality of your application, the extent to which these applications can move onto the edge application platform will vary. The edge View and Controller components are bundled, along with other Java classes, into a Web application archive (WAR) and deployed onto the edge server network.

Normally in an enterprise environment the Web tier will use RMI (Remote Method Invocation) to communicate with the business tier. For example, a servlet may use RMI-IIOP to call an EJB. However, with J2EE application components running in a distributed WAN environment, EJBs will remain running at the enterprise and are made accessible to the edge application via Web services interfaces. A session bean facade can be used to expose the necessary business logic as a Web service to the edge application. The edge application makes a Web service call back to the enterprise to invoke the appropriate logic through a session bean facade, perhaps made visible through a servlet running on the enterprise application servers. The edge application can use JAX-RPC to call the Web service at the enterprise via SOAP/HTTP(S). The edge application platform enables an edge application to cache the results of SOAP/HTTP(S) requests to optimize interactions with the enterprise.

In addition to Web services as a communication channel, other standard communication protocols are supported, including the following:

- ***HTTP(S):*** An edge application can make HTTP(S) requests to the central enterprise, using the HttpURLConnection class, for example, to request content or data objects. Any content fragments can be fetched and used to build the end-user response. Data objects such as XML files can be fetched and then transformed (XSLT) or parsed (JAXP) to create the end-user response. HTTP responses from the enterprise can be cached on the edge, so content or data objects can be persisted on the edge across user requests to further reduce the load on the enterprise.
- ***JDBC:*** Akamai provides a Type 3 JDBC driver that allows edge applications to tunnel JDBC queries to the enterprise via HTTP(S). If an application has already been developed using JDBC for data transaction with a database, the JDBC/HTTP mechanism will make it easier to adapt an application for the edge. The edge application can still use JDBC and benefit from the relational data form. Further, JDBC query responses can be configured to be cached on the edge servers.
- ***RMI:*** WebSphere applications running at the edge can use RMI-IIOP tunneled over HTTP(S) to communicate back to the enterprise. In this configuration, a servlet runs at the enterprise, intercepts the RMI requests from the edge, and translates the requests into the appropriate RMI method calls to the business tier. Since RMI calls are set to be uncacheable in the edge application environment, it is recommended that you use Web ser-

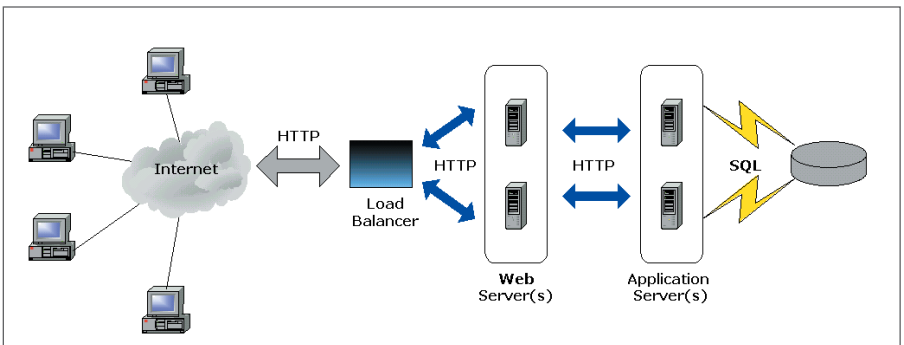


FIG. 1: A TYPICAL ENTERPRISE APPLICATION INFRASTRUCTURE

DATA ACCESS AND UPDATE FREQUENCY	Read-only access Low update frequency	Read/write access Low update frequency	Read/write access High update frequency	Write access Low/high frequency
BEST PRACTICE	Local DB on edge (Cloudscape)	Web service from edge to origin – cache at edge	If read access can be cached and preserve consistency needs, use Web service from edge to origin	Tunnel from edge to origin (Akamai edge to origin protocol optimizations)

TABLE 1: BEST PRACTICES FOR DIFFERENT DATA ACCESS REQUIREMENTS

vices over RMI to achieve better performance.

Data Access on the Edge

Any of these HTTP-based protocols are useful interfaces that allow your applications to bridge from the edge to the enterprise, but it is still important to avoid excessive communication for edge-origin requests, as end-user latency and origin utilization will increase. Since there is an absolute cost for every round-trip from the edge to the enterprise, calls should be as effective as possible. Requests should be bundled, if possible, and edge caching should be used to store data and other objects across requests.

The EdgeComputing platform provides caching mechanisms to persist application data to minimize the interaction and load on the enterprise infrastructure on a request-by-request basis.

- ***ServletContext:*** Any data that can be used across users for a particular application can be stored in the application's ServletContext object.
- ***HttpSession:*** Akamai supports the standard HttpSession interface on the edge application platform. The platform provides replication for HttpSession objects across the edge server nodes. HttpSession objects are most commonly used to keep per-user state information, but keeping them small in size is important for performance reasons.
- ***HTTP cookies:*** An edge application can store some user-specific data in user cookies or URL tokens, but privacy and security concerns may prevent the use of cookies to store data in some situations.
- ***Object caching:*** As previously mentioned, an edge application can make requests for data or content (using HTTP, Web services, or JDBC) and these objects can be stored in the edge node's cache.

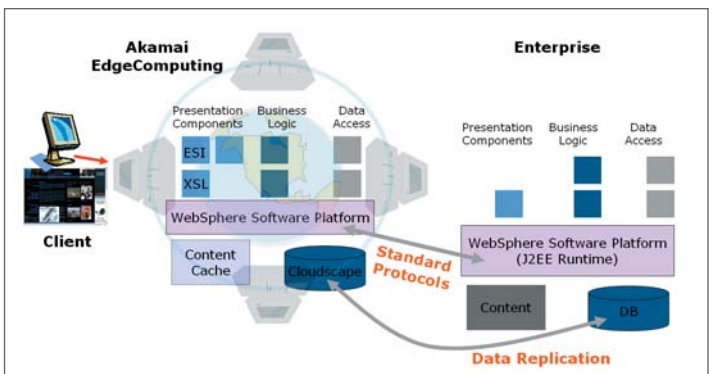


FIG. 2: A WIDELY DISTRIBUTED APPLICATION ENVIRONMENT

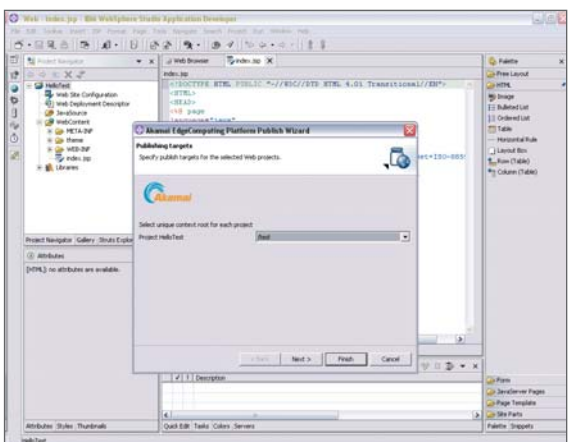


FIG. 3: THE GUI EXPORT WIZARD

Another powerful edge data capability employs IBM's 100% Pure Java Cloudscape DBMS to host infrequently changing, read-only data in an edge database. In this model, application data is exported into a JAR file (done by the enterprise application developer), and this "database in a JAR" file is bundled into the edge WAR file under the standard WEB-INF/lib directory. The application's data source has a fixed configuration corresponding to the location of the database within the JAR file, and the application and corresponding configuration are deployed in the WAR to the edge servers.

By using Cloudscape on the edge, even the Model components of an MVC application can be distributed onto the edge application platform. Edge applications can make use of JavaBeans and JDBC as Model components with Cloudscape as the DBMS to further reduce communication to the back-end enterprise systems.

One disadvantage of this model is that any time the application data changes, the application must be redeployed to the edge network. Future integration development between IBM and Akamai will enable Cloudscape to function as a database query cache at the edge of the network. This will enable dynamic caching of query results depending on the SQL statements issued by the edge application.

Table 1 outlines suggested best practices given the type of data access required by an application running on the edge.

Edge Application Examples

The following examples describe some applications modeled to run on EdgeComputing and illustrate the use of WebSphere Web services and Cloudscape in distributed edge applications.

- ***Product catalog:*** A product catalog browsing application can run almost entirely in the edge environment. Since most product catalogs consist of relatively static product data (not including inventory information), the edge application can utilize Cloudscape as the local DBMS. The data can be bundled into the edge WAR along with the catalog-browsing presentation components. Using this deployment model, it is feasible for the end user browsing interaction to be handled entirely by the edge application. When a user is ready to purchase any selected items, the edge application tunnels back to the enterprise for order processing.
- ***Marketing promotional contest:*** Say an enterprise wants to conduct a large-scale marketing promotion to give away a certain new product. Because of the uncertainty of the number of end users (contestants), an on-demand edge application is extremely beneficial to assuring a successful outcome. In this scenario, the application might have "random selection" logic to determine if an end user is a winner.

An application can be designed and developed to execute this logic on the edge, offloading the load from the enterprise. In addition, the corporate marketing team can implement various controls on how long the contest runs, how many products are given out, the rate at which they are disbursed, or other controls. The edge application executes the corresponding business logic entire-

ly on the edge and retrieves the control parameters from the enterprise via Web services calls.

Application Deployment to the Edge Platform

Once designed, developed, and tested, the edge subapplication is uploaded and provisioned on the edge network. Akamai has developed a provisioning mechanism that is available via two interfaces. The conventional interface involves using a browser-based GUI on the Akamai enterprise portal. A second interface involves uploading and provisioning via SOAP-based Web services that can be invoked programmatically.

IBM and Akamai have developed a plug-in for WebSphere Studio Application Developer (WSAD) in the form of a GUI export wizard, shown in Figure 3. This plug-in securely invokes the EdgeComputing provisioning Web service, and it allows developers to deploy their applications directly from the WSAD environment. In addition to the upload function, the plug-in is being enhanced to provide validation and a simulated unit edge test environment for the WSAD environment.

Summary

IBM and Akamai have developed a new deployment model for J2EE Web applications. The Akamai EdgeComputing powered by WebSphere service allows application components to migrate to the edge. By extending the WebSphere programming platform to run on the Akamai edge network, enterprises are able to attain higher levels of performance, reliability, and global reach and ultimately achieve scalability on demand. Akamai EdgeComputing powered by WebSphere helps enterprises respond to a growing need for "on-demand" computing and will help many enterprises adapt their business functions to the edge.

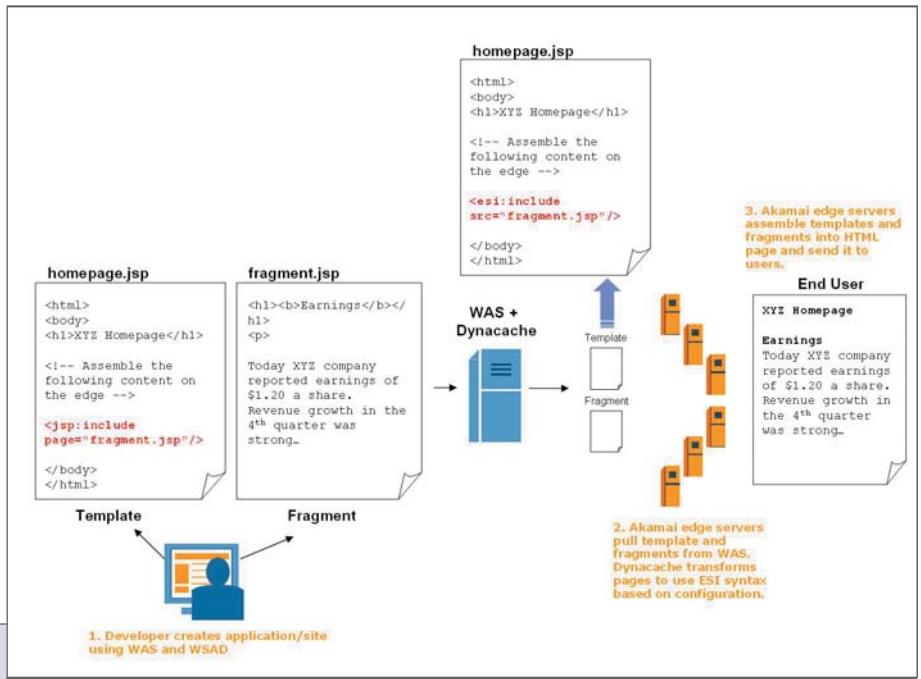


FIG. 4: THE ESI LOGIC

EDGE SIDE INCLUDES

Edge Side Includes (ESI) is a simple XML-like markup language that can be used to assemble dynamic, personalized Web pages at the edge of the Internet. Using ESI can dramatically improve the delivery of dynamic Web content. ESI content assembly is based on the fact that a dynamic Web page is composed of several smaller individual pieces, or fragments. Each of the individual fragments can have its own cacheability properties. An edge server processes the ESI logic and will fetch only noncacheable or expired fragments from the origin Web site. This model of edge page assembly can significantly reduce the load on the enterprise infrastructure, as well as the amount of data that must be transported to the edge.

IBM's WebSphere Dynamic Caching Service component, as part of WebSphere Application Server, supports the ESI specification and seamlessly integrates with Akamai's ESI processing capability. A page composed from a collection of servlets/JSPs (e.g., a portal-style page) can be assembled on the Akamai edge servers from either cached or uncacheable components. JSP/servlet results can be defined as "edgeable," using the WebSphere assembly tool or by directly editing the WebSphere Dynamic Caching Service configuration file, cachespec.xml. The ESI logic is illustrated in Figure 4.

Upon receiving an end-user request, an Akamai server contacts the enterprise WAS server. It adds an identification HTTP request header in the form of: Surrogate-Capability: akam="ESI/1.0". Upon processing this header, WAS returns content containing esi:include tags instead of a fully built Web page. The embedded ESI tags are processed by the Akamai edge server and the appropriate fragments are fetched (from the enterprise or from cache) to build the response to the end user.

The following code snippets show how a main, or template page, homepage.jsp, could call and insert the fragment, fragment.jsp.

homepage.jsp

```
<html>
<body>
<h1>XYZ Homepage</h1>
<!-- Assemble the following content on the edge -->
<jsp:include page="frag.jsp"/>
</body>
</html>
```

fragment.jsp

```
<h1><b>Earnings</b></h1>
<p>
Today XYZ company reported earnings of $1.20 a
share. Revenue growth in the 4th quarter was
strong…
```

WAS can invoke an Akamai Web services interface to force an invalidation of any templates or fragments. Additionally, IBM and Akamai are developing a mechanism for WebSphere cache configuration settings defined in the enterprise's WebSphere Dynamic Caching Service cachespec.xml to be automatically set in Akamai's customer cache configuration metadata.

WEBSPHERE **NEWS**

Dirig Software to Support WebSphere for z/OS and s/390

(Nashua, NH) – Dirig Software, a developer of adaptive application performance management solutions, has announced that its J2EE application performance management solution is available for IBM WebSphere for z/OS and S/390 environments.

The Dirig J2EE application performance management solution provides the ability to monitor and proactively manage every Java application function, connection point, and component to validate and ensure the integrity of each transaction request, execution, and completion.

The software auto-discovers, inventories, monitors, alerts, and/or responds while capturing the key application forensic detail to pinpoint the impact and root causes of slowdowns, failures, and outages. The component management ensures that the most critical pieces of an e-business solution are fully functional, including tracking usage and failure statistics on servlets, Enterprise JavaBeans, and connection pools. For resolving issues, the solution provides graphical tracking detail to identify problematic application code hotspots.
www.dirig.com

IBM, Akamai Team Up to Offer On-Demand Solution

(Somers, NY and Cambridge, MA) – IBM and Akamai Technologies, Inc., have announced new software that simplifies deployment of Java 2 Enterprise Edition (J2EE) Web applications built on IBM's WebSphere software across Akamai's global computing platform.

In May, IBM and Akamai announced a new joint solution – Akamai EdgeComputing powered by WebSphere – that combined Akamai's EdgeComputing Service – a utility-like computing service for J2EE application delivery – with IBM's market-leading WebSphere Internet infrastructure software. With the offering, WebSphere customers can use the Akamai global network to access and process Web applications fast and reliably, regardless of Internet failures.

The new software enables the Akamai EdgeComputing powered by WebSphere solution to make effective use of WebSphere Studio development tools, creating a single environment so developers and application architects can easily deploy J2EE applications. The developer codes an application, simply clicks on deploy, and the infrastructure and capacity of Akamai's global network are instantly available at the developer's fingertips.

Developers can download a technical preview of this technology – the EdgeComputing Toolkit for WebSphere Studio – for free from alphaWorks, IBM's resource for emerging technology (www.ibm.com/alphaworks). It will be generally available in the first quarter of 2004 through IBM Global Services as well as through Akamai's professional services group.
www.akamai.com/en/html/services/edgecomputing.html

Bowstreet Portlet Factory for WebSphere 5.7 Released

(Tewksbury, MA) – Bowstreet, a provider of development tools for adaptive J2EE applications, has announced the availability of Bowstreet Portlet Factory for WebSphere version 5.7, which has earned the prestigious Ready for IBM WebSphere Studio software validation. This new release also features the SAP Extension, allowing SAP customers to rapidly create customized portlets that expose SAP data and business processes to IBM WebSphere Portal.

The Ready for IBM WebSphere Studio software validation confirms Bowstreet Portlet Factory's seamless integration with WebSphere Studio. Bowstreet Portlet Factory provides developers with one powerful tool for all of their Java and J2EE development needs by delivering automation and rapid portlet development to WebSphere Studio, the company says.
www.bowstreet.com.

SCAPA TECHNOLOGIES LAUNCHES HTTP LOAD TESTING PRODUCT FOR WEBSPHERE AND ECLIPSE

(Edinburgh, UK) – Scapa Technologies, a supplier of innovative load and stress testing solutions for commercial software systems, has announced the forthcoming release of Scapa StressTest-Express, an affordable and highly intuitive load testing tool that analyzes the scalability, reliability, and performance of Web applications. Scapa StressTest-Express, based on the Eclipse platform, is designed to ensure a low-risk approach to building dynamic Web sites and establishing e-business systems. Scapa StressTest-Express provides a high level of integration and interoperability with IBM WebSphere Studio and has been validated through the Ready for IBM WebSphere Studio program.

As with IBM's WebSphere – Express product family, Scapa StressTest-Express is a low-cost, easy-to-use product designed, built, and priced to enable small- to medium-size companies to launch their e-business implementations confidently and ensure their investment in Web application technology as their businesses grow.
www.scapatech.com

Candle PathWAI Packages Accelerate J2EE and EAI Initiatives

(El Segundo, CA) – Candle Corporation, an enterprise infrastructure management provider, has announced six new PathWAI packages that accelerate IBM WebSphere Java 2 Enterprise Edition (J2EE) and enterprise application integration (EAI) initiatives.

Designed for builders, tuners, and testers of application infrastructure, the new packages empower application infrastructure development teams to rapidly identify and eliminate "early life-cycle" problems that cause operational performance issues and lead to deployment delays and cost overruns. These packages extend Candle's award-winning PathWAI portfolio of WebSphere application infrastructure management solutions.

The packages include PathWAI Performance Workshop for J2EE, PathWAI Tuning Workbench for J2EE, PathWAI Deployment for J2EE, PathWAI Performance Workshop for WebSphere Business Integration, PathWAI Tuning Workbench for WebSphere Business Integration, and PathWAI Deployment for WebSphere Business Integration.
www.candle.com.

SMARTS Unveils WebSphere Managment Solution

(White Plains, NY) – SMARTS, a provider of automated real-time business assurance solutions, has unveiled InCharge for Application Servers – WebSphere Edition. Further extending InCharge's ability to intelligently analyze and holistically manage mission-critical application services, the newest solution provides total real-time visibility into the IBM WebSphere environment – pinpointing critical problems affecting complex, distributed WebSphere-based applications by correlating application performance with the health of the surrounding application, server, and network infrastructure.

InCharge is the industry's only solution with the ability to automatically discover the WebSphere and network environment, and to conduct cross-domain correlation to isolate "Authentic Problems" impacting application performance, regardless of whether they originate in the network, the server, or WebSphere itself, the company says. With InCharge for Application Servers – WebSphere Edition, IT operations teams get the real-time actionable information they need in order to maximize the performance and availability of production applications and the business services they support.
www.smarts.com

TLCC Announces New Curriculum for WebSphere Developers

Many Notes/Domino developers have taken The Learning Continuum Company's Introduction to WebSphere 5 for Domino Developers course. Although this course is very popular, many developers with no Notes or Domino skills have also expressed an interest in learning WebSphere via TLCC's proven distance learning system.

A new course is now available for anyone who wants to learn WebSphere. Introduction to WebSphere Studio 5 is for any developer with a basic knowledge of Java. Developers simply download the Introduction to WebSphere Studio 5 to their home or office computer and learn how to develop J2EE applications using WebSphere Studio.

Installed as a plug-in to WebSphere Studio, this interactive course is completely integrated with WebSphere Studio. Lessons are viewed right in WebSphere Studio. Many live demonstrations and activities are included in the course. A Notes client is not required.

The retail price of this course is $599. For a limited time students can save $100 off the retail price of $599 and get a free CertFX practice certification exam. A free demonstration course is available to try out this unique way of learning WebSphere.
www.teachmewebsphere.com

NEW VERSION OF WEBSPHERE MICRO ENVIRONMENT TOOLKIT FOR PALM OS DEVELOPERS RELEASED

(Milpitas, CA) – palmOne, Inc. has announced the latest release of a developer toolkit that brings Java 2 Micro Edition support to Treo 600 smartphones and Tungsten handhelds, which significantly expands the capabilities of its product lines for business customers and developers. A beta of the new version of IBM's WebSphere Micro Environment Toolkit for Palm OS Developers, which includes a new high-performance runtime with support for the latest Java Community Process standards, has been released to developers through the palmOne PluggedIn developer Web site.

The toolkit enables developers to target, deploy, and execute their Java MIDlet applications on palmOne devices, directly alongside native Palm OS applications. The early-access release allows software developers to design and test applications that utilize the new standards support and other capabilities in advance of the end-user runtime release next spring. With support for both the Treo 600 smartphone and the Tungsten line of handhelds, compelling, standards-based applications and solutions can now be deployed in a consistent way to mobile workforces and consumers alike.

Developers can download the no-cost toolkit, both the beta of the MIDP 2.0 toolkit as well as the existing MIDP 1.0 toolkit, from www.palmone.com/java. The MIDP 2.0 runtime is scheduled to be available for both the Treo 600 smartphones and the Tungsten handheld devices next spring. The runtime will cost $5.99 for Treo 600 smartphone users.
www.palmone.com.

RSS Feed Now Available on WJ Home Page

The WJ home page (www.sys-con.com/websphere) now offers an RSS feed (www.sys-con.com/WebSphere/feed.rss) to help you keep current with the latest goings-on in the WebSphere community. Sign up today to start receiving timely updates of product announcements, news items, and other important happenings in the world of WebSphere.

Excerpted from Chapter 2:
The WebSphere Programming Model

Professional IBM WebSphere 5.0 Application Server

Written by IBM WebSphere experts, this book details how to develop, deploy, and manage enterprise applications for version 5.0 of IBM WebSphere Application Server. Over the course of the book, a large-scale e-commerce application is developed, demonstrating the use of WebSphere Application Developer Studio. The book also addresses other enterprise-level issues such as the functionality of the application server, deployment topology, and server administration. The following excerpt from Chapter 2 focuses on the WebSphere programming model.

We examine many aspects of the WebSphere programming model in this book. However, notwithstanding the breadth of coverage included in this book, the total WebSphere programming model is larger than we can hope to address between a single pair of covers. Fortunately, the majority of the WebSphere programming model is based on open standards, and so, a great deal of what we don't cover here can be learned from other readily available sources – the most important of which is the Java web site at: http://java.sun.com/. Other useful books on the general topic of J2EE programming include:

- *Professional Java Server Programming J2EE 1.3 Edition*, Wrox Press, ISBN: 1-86100-537-7
- *J2EE Design Patterns Applied*, Wrox Press, ISBN: 1-86100-528-8
- *Expert One-on-One: J2EE Design and Development*, Wrox Press, ISBN: 1-86100-784-1

To help put WebSphere in perspective, we provide an overview of the entire programming model in this chapter:

- We begin with a discussion of the basic models of computing supported by the WebSphere programming model.
- We then proceed with a discussion of the classic J2EE roles and some thoughts about additional roles that can contribute to the development and deployment of applications in your enterprise. This will include a discussion on how these roles play into the overall programming model process.
- We follow with an overview and introduction to the additional programming facilities introduced by WebSphere that go beyond the J2EE standard, including the web services programming model.

This chapter will also outline the APIs included in the WebSphere programming model. It will provide some perspective on the value of the functions intrinsic to the WebSphere programming model – hopefully offering you some new ways of thinking about your business applications.

We will, in this discussion, provide some insight on which APIs are considered strategic, which are supported across all of the WebSphere platforms and editions, and which are restricted. We will introduce the idea of privileged code to gain access to otherwise restricted elements of the programming model. By the time you've finished reading this chapter, you will have a base line on the WebSphere programming model, and be better prepared to understand the programming concepts and details presented through the rest of this book.

Models of E-Business Computing

WebSphere is a J2EE-compliant application server supporting the entire breadth of the J2EE specification. WebSphere version 5.0 is certified at the J2EE 1.3 level, and as such, J2EE is at the heart of the programming model. Given the platform portability premise of Java (and J2EE itself), there is a good chance that you will be able to port your conforming applications to WebSphere with little effort.

However, that only tells a part of the WebSphere story. As programmers, our needs for information computing are varied, dynamic, and growing. Most computing scenarios today have to address a variety of issues: end-user delivery channels, development methodologies, business-enablement approaches, legacy protection, and fulfillment goals. WebSphere has excellent capabilities for supporting many of these requirements. To understand these, it would be best to start with a basic understanding of the key models of computing and how WebSphere addresses each of these. Then, you can combine this knowledge to form a solution tai-

ABOUT THE BOOK

Professional IBM WebSphere 5.0 Application Server
by Rob High et al.
ISBN: 0-7645-4366-0
Publisher: John Wiley Publishing
List Price: $79.99

lored to your specific situation.

WebSphere provides support for four basic models of e-business computing:

- Multi-tier distributed computing
- Web-based computing
- Integrated enterprise computing
- Services-oriented computing

The fundamental structures of application design enable distributed computing for different business and organizational scenarios. Most applications will eventually exploit a combination of these models to solve the needs of the environment in which they will be used. The WebSphere programming model covers the full span of each of these models.

If you are already familiar with the fundamentals of distributed computing, component programming, shared and reusable parts, business process management, and service-oriented architecture then you might like to skip ahead a bit. More so, if you're already familiar with J2EE and the core premise for WebSphere, then skip this section – move on to The WebSphere Development Model section. If, on the other hand, you're not sure about these topics or a brief refresher might cement the issues in your mind then read on. Our view is that to best understand WebSphere you need to understand the types of computing models it is designed for.

Multi-Tier Distributed Computing

Multi-tier distributed computing is what motivated the development of many of the core server technologies within WebSphere. The idea of component-based programming is to define reusable and shared business logic in a middle-tier of a three-tier distributed application [see Figure 1].

The value of three-tiered distributed computing comes from first structuring the application with a clean separation between the logic elements (presentation, business, and data) and then leveraging the boundaries between these elements as potential distribution points in the application, allowing, for example, the presentation logic to be hosted on a client desktop, the business logic in a middle-tier, and the data logic on a traditional data centre.

Placing the presentation logic on the user's desktop has the benefit of enabling a rich interaction model with the end user. Placing the data logic in the traditional data center allows tight, centralized control over the data and information assets of the enterprise. Placing the business logic in a middle-tier allows for the exploitation of a variety of computing systems and better reuse and sharing of common computing facilities to reduce the cost of ownership that is commonly associated with expensive thick clients. It also means that you don't have to manage large quantities of application logic in the desktop environment.

The boundary between each tier represents a potential distribution point in the application, allowing each of the three parts to be hosted in different processes or on different computers. The J2EE programming model provides location transparency for the business logic, allowing you to write the same code irrespective of whether the business logic is hosted over the network or is in the same process. However, you shouldn't let this trick you into treating your business logic as though it is co-located in the same process – notwithstanding the tremendous improvements we've seen in networking technologies, communication latency will affect the cost of invoking your business logic and you should design your application keeping in mind that such boundaries can carry this expense.

Even if you don't distribute your application components across a network, you can still benefit from the boundary between the tiers. Every boundary that you are able to design into your application becomes an opportunity to "plug-replace" other components. Thus, even nominal distribution of these components can improve the ease of maintenance of your application. It also becomes an opportunity for the runtime to spread workload over more computing resources (we'll discuss this in detail in Chapter 12).

Through J2EE, WebSphere provides a formalized component architecture for business logic. This component technology has several key benefits to application development. Foremost, the component model provides a contract between the business logic and the underlying runtime. The runtime is able to manage the component. This ensures the optimal organization of component instances in memory, controlling the lifecycle and caching of state to achieve the highest levels of efficiency and integrity, protecting access to the components, and handling the complexities of communication, distribution, and addressing (see Figure 1).

Secondly, the component architecture allows the client programming model to conform to well-established rules, which implies that distributed components are shared components – that is, the same component can be used by many different applications simultaneously [see Figure 2].

Finally, because the J2EE component model is designed around object-oriented principles, you are able to do a better job of modeling your business in the application design. This will help improve communication between you and your business end users – you can cast your application artifacts in terminology that they recognize, with behavior that is consistent with the conceptual model of the business

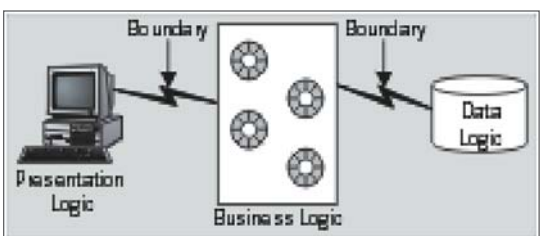


FIG. 1

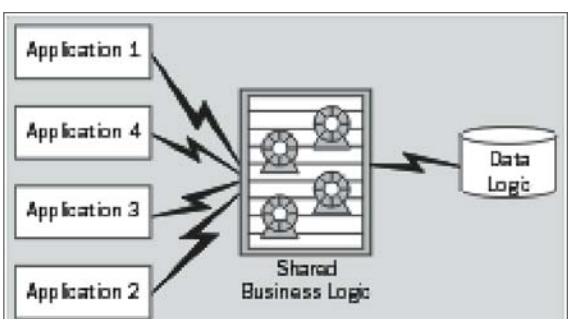


FIG. 2

they are trying to automate. We recommend that you exploit UML or some other standard modeling notation to define your basic business model design, and then use that model to generate the base J2EE artifacts in which you will implement that model. Rational Rose, for example, is one of several products that allow you to build a UML model, and then export that to a J2EE component implementation skeleton.

So, having a single, shared implementation and instantiation of business components in an environment where the developer doesn't have to deal with concurrency issues not only reduces the duplication of development effort, but also allows you to concentrate more on ensuring the correctness and completeness of your component implementation. It gives you more control over your business processes by ensuring that business entities are adhering to your business policies and practices – you don't have to worry that someone has an outdated or alternative implementation of your business model sitting on their desktop. We have all heard stories about loan officers giving out loans at the wrong interest rate with inadequate collateral because someone had out-of-date rate tables loaded in the spread-sheet. Shared business logic implementations help avoid these costly problems by encouraging fewer implementations of the business logic and therefore fewer opportunities for the business logic to get out of sync.

With the introduction of a component technology for business logic in a three-tiered distributed computing model, we immediately began to see design patterns for the composition of business objects. That is, businesses want to be able to form higher-level business concepts that aggregate previously independent elements of their business. Instead of a customer having multiple accounts, and therefore, multiple relationships with the business, the enterprise could bring together all the individual accounts for a given customer under a single customer object and thus form a better and more enhanced relationship with the business. The business logic could thus be structured in multiple tiers, ranging from general to more specific, all formulated under the same underlying component architecture.

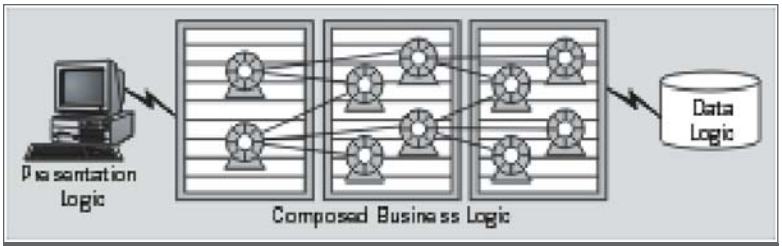


FIG. 3

"Many of the benefits of component-based programming come from the separation of application logic from concerns of information technology"

From this was born the idea of multi-tier distributed computing, where the middle-tier of business logic could be composed of an arbitrary number of intermediate tiers all running on a common WebSphere J2EE platform architecture and thus ensuring consistency of programming, deployment, and administration [see Figure 3].

We should reiterate here that the multi-tiered distributed computing nature of component-based programming for business logic enables the integration of distribution points within your application design. It also defines a contractual boundary to your business components that can be leveraged to share your business logic components within many different applications, which in turn may be distributed. However, you don't have to distribute the components of your application to benefit from component-based programming. In fact, distributing your components introduces coupling issues that are not normally found in centralized application designs.

In cases where you know the latency of distributed communication between two components cannot be tolerated in your application design, you can instrument your component with local interfaces that will defeat distribution, but still retain the benefits of component-based programming.

Many of the benefits of component-based programming come from the separation of application logic from concerns of information technology. In particular, the J2EE component model enables a single-level-store programming model, whereby the issues of when and how to load the persistent state of a component are removed from the client.

The component model allows the runtime to manage the component in the information system. This same principle of management applies to object identity, transaction and session management, security, versioning, clustering, workload balancing and failover, caching, and so on. In many cases, the runtime has a much better understanding of what is going on in the shared system than any one application can ever have, and thus can do a better job of managing the component and getting better performance and throughput in the information system. Since WebSphere is a commercially available product, you can acquire these benefits for much less than it would cost you to create the same capability on your own.

Versata's Rules

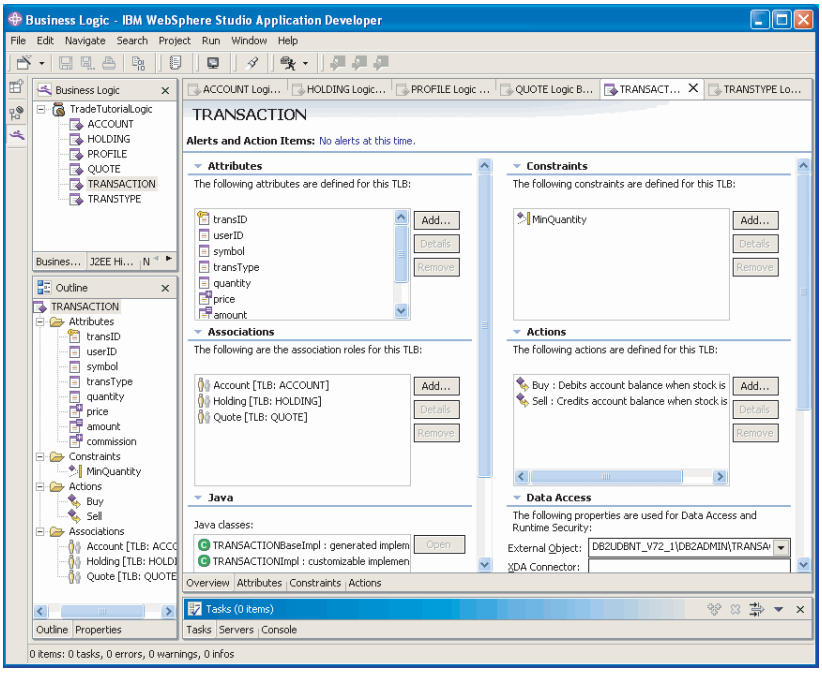
for

Simplifying WebSphere Studio Development:

RULE THREE

DEVELOPERS LIKE NEW SOFTWARE, ESPECIALLY **FREE** SOFTWARE.

J2EE can be a bear without Versata business logic software. As a plug-in for IBM's WebSphere Studio, Versata helps you develop and manage server-side business logic components, driving data for JSPs, HTML, Java, and Web services - making life a whole lot easier. Rather than program hundreds of lines of code, you simply specify your business logic as business rules: Versata does the rest.



versata™

To download a **free** copy of the Versata Business Logic Designer for WebSphere Studio, visit **www.versata.com/businesslogicdesigner**

Application Quality Management: Views You Can Use

BY DAVE DANIELSON

As the old saying goes, "Those who can, do; those who can't, teach." Well, there's a bit of such crass finger-pointing going on inside each corporate IT operations and development staff every day.

It goes without saying that whenever something bad happens with application performance, most users blame the network – and in the network it must be the router, right? But in today's application environment, the truth is not nearly as simple as this common but misguided statement. There are a myriad of moving parts that make up "the application," and finding the source or sources of performance problems is never that easy!

When a newly developed WebSphere application fails to perform in a distributed production environment, software developers and vendors alike tend to blame the network for the miserable performance. Unfortunately, the reality of troubleshooting application performance problems today is something that could be called "troubleshooting by committee" – where as many as 5 to 10 subject matter experts converge on the network and application infrastructure to poke and probe, often working for days to find the source of the performance problems.

Why does this happen? Don't we have enough tools? Haven't we bought all those products that promised to make life better over the past few years? The truth is that the typical enterprise has an operational organizational structure built around networks (Network Ops, etc.) and that the production side of the house has not completely embraced a deeper understanding of applications. So, if you are a developer of WebSphere applications, you need to embrace the reality that sooner or later you will be called into operational service to diagnose or handle application performance problems – even though, historically, these haven't been typical responsibilities of application developers.

The more compelling question for WebSphere developers is what to do about it. Despite the best efforts of developers, applications often don't perform as planned when deployed into a new production environment. There are many possible reasons for this, but again the key question is what to do about it. This situation is unlikely to change soon.

More companies today are learning that they must address this issue in a systemic manner. Application Quality Management (AQM) does just that. AQM is targeted at the fundamental building blocks of application performance.

- ***Process:*** Focusing on pre-production understanding or performance prior to production
- ***Tools:*** Required to monitor and manage performance in system testing and production
- ***Organizations:*** Application management groups tasked to deal with application performance issues in production

Fundamental to each of these is the ability to achieve detailed monitoring across the distributed application and to understand its behavior as a system.

There is a clear need for an embedded capability for IT to monitor WebSphere applications in context. Ideally, this would be a capability that would be implicit from development through deployment and that would allow network operations personnel to evaluate the "application footprint" in action. Of course, this was precisely the point of the initiatives a few years ago to embed quality-control telemetry into programs. These initiatives failed in most environments mainly because programmers hated building internal telemetry.

Now, however, a new class of management technology is available to monitor WebSphere applications in order to evaluate the impact of network and systems performance on application performance. While the current solution is to test the application in a networked environment, such tools allow developers to build test protocols directly into a tool that accompanies the application into production. Notably, this technology allows developers to predefine allowable limits that operations personnel can compare to actual metrics in order to finally answer the question of whether the network really is the problem.

AQM requires tools that can provide very deep insight into the WebSphere application environment, allowing developers to evaluate the performance of their code under production loads and then develop monitorable thresholds that can accompany the applications into the production environment. Another useful feature would be to provide highly abstracted and business-centric metrics that allow even nontechnical personnel to understand the dynamics of WebSphere application performance. Additionally, if such a tool can be bundled and shipped by ISVs along with applications as a built-in control panel for their applications, a new channel to the enterprise market could be supported in a productive manner.

IT managers and ISV management teams planning the deployment of new WebSphere applications may want to think about incorporating the concept of Application Quality Management to simplify the process of stabilizing their applications in production, as well as to significantly reduce finger-pointing when applications are deployed.

ABOUT THE AUTHOR... Dave Danielson is CEO of Altaworks (www.altaworks.com), where he is responsible for building the company's momentum and ensuring its ongoing position as the leading provider of Web enterprise performance management software solutions. Dave brings over 17 years of experience in leading-edge technology products and startup companies to Altaworks.

Introducing JProbe® 5.0
…now smarter and faster than ever

QUEST SOFTWARE™

For more info and a free eval, visit:

http://java.quest.com/jprobe/wdj

diagnoSys

Discover how you can fix your performance problems at H&W's free Web seminar series. Visit www.hwcs.com/wsd1.asp

You might not even know you have a leak, but that little drip can turn into a big deal if it's left unattended. Performance problems with your Java applications work the same way.

Hung threads, memory leaks, and faulty network connections can cause your application to perform poorly or even fail when it matters most – in production. Customers click away and profits plummet. But how do you find the problems if you don't even know they are there?

DiagnoSys is a complete software solution that allows you to analyze the performance of your Java applications through all phases of the life cycle. With true statistical correlation, in-depth analysis, and root cause identification, it helps your teams work together to find problems while they are still little.

The end result? With DiagnoSys, you build vital applications that optimize uptime and keep the bottom line healthy. No problem.

H&W

www.hwcs.com | 1.800.338.6692

Bridging the Enterprise Computing Gap Since 1979