SECURITY • PERFORMANCE • WEB SERVICES

WebSphere® DEVELOPER'S JOURNAL

The World's Leading Independent WebSphere Developer Resource

WebSphereDevelopersJournal.com

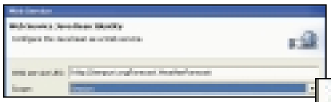
FEBRUARY VOLUME 2 ISSUE 2

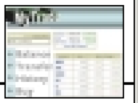
EXCLUSIVE INTERVIEW...

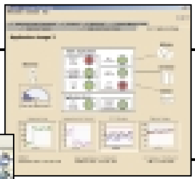
A Conversation with IBM's Suchitra Joshi

THE COMPETITIVE TECHNOLOGY LABS' ROAD SHOW BRINGS ITS MESSAGE TO THE WORLD

INTERVIEWED BY JACK MARTIN PAGE 24







DISPLAY UNTIL APRIL 30, 2003
$8.99US $9.99CAN
0 2>
0 09281 03422 3

SYS-CON MEDIA

Companies Must Deliver to Survive in Tough Times

BY JACK MARTIN

AAfter reading Jim Martin's closing piece last month I wondered if we were living on the same planet. He envisions a huge upturn in tech spending – a return to the good old days. I see massive consolidation, with the weakest players going out of business and the companies with good, serviceable products either just getting by or being acquired.

The handwriting is on the wall. All one must do to see the future is look at what is happening to the companies who actually buy enterprise technology. For example, Burger King and McDonald's have rolled prices back to the 1970s for their flagship products (which are also their most profitable), the Whopper and the Big Mac. Get them while they last for 99 cents each, all day, every day. This is a new burger war in which the two top competitors are selling their products below cost. Don't expect any big capital spending out of these two.

Or if you prefer to look up at the sky, you will see that every American-based airline is going broke flying their jets. I know JetBlue is reporting a profit in this market, but they are also flying a brand-new fleet of planes that haven't begun to require real maintenance and subsequent interruption of cash flow, nor do they have any pension expenses to pay out yet. If you were to add in these expenses, which will be unavoidable for JetBlue in the future, you will find that they are doing no better than any other airline.

Or you can come back down to earth and look at the automotive manufacturing sector. It is now common practice to retail new cars with zero-interest financing directly to the consumer, who may never pay the loan off due to repossession or accidents beyond his or her control. There isn't any profit at all in making small or medium-size cars; all the real money is in SUVs, and people are starting to understand that paying $50,000 for a truck with fancy seats and rugs – and that can instead be purchased for $22,000 – is pretty stupid. Yikes! There goes the automotive industry down the hopper. If we actually go to war and oil prices spike significantly – all of a sudden it will cost a working guy $100 plus per week to buy gas for his oversize pickup truck – there goes the resale value and up go the repossessions. Not exactly a formula for making profits.

We are quickly approaching a period of deflation. Not a single soul reading this editorial has lived through such economic times on a national scale. Deflation is a really interesting economic concept. It is a period where prices are constantly falling and assets are worth less and less. This may at first blush seem like a really great deal for the consumer, but it isn't. Because as prices continue to fall, so do wages. Every homeowner in America paying on a mortgage does so based on the simple expectation that the house will be worth more in the future. As anyone will tell you, over the long term real estate can only go up. Wrong: until the expansion of New York City hit high gear right after the Civil War, real estate was viewed as a terrible investment, as it was never really worth much of anything as long as there was so much of it. Take a close look around. Most of the available land in America is undeveloped, with not even a single structure on it.

One can currently move to North or South Dakota and get a house or even an entire town for next to nothing. Montana, Wyoming, and Idaho all have similar pricing opportunities for the 21st-century homesteader. Guess what? Real estate prices continue to fall in these and other areas. This is deflation applied to real estate. Just think what it would be like if the entire economy started to behave like that.

The automation revolution is absolutely here to stay, but the near term is going to be rocky at best. The companies that are here to stay will be the ones who consistently deliver value – not promises – to their customers.

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

WebSphere® DEVELOPER'S JOURNAL

ADVISORY BOARD

Richard Arone, James Martin, Greg Williams

EDITOR-IN-CHIEF	JACK MARTIN
EDITORIAL DIRECTOR	JEREMY GEELAN
EXECUTIVE EDITOR	GAIL SCHULTZ
CONTRIBUTING EDITOR	PATTI MARTIN
SPECIAL PROJECTS EDITOR	RICHARD ARONE
MANAGING EDITOR	JEAN CASSIDY
EDITOR	NANCY VALENTINE
ASSOCIATE EDITORS	JAMIE MATUSOW JENNIFER STILLEY

WRITERS IN THIS ISSUE

James Carey, Brent Carlson, Gennaro Cuomo,
Catherine Diep, Greg Flurry,
Marcel Heijmans, Jack Martin, Mike Malloy,
Granville Miller, Ken Mitsumoto

SUBSCRIPTIONS

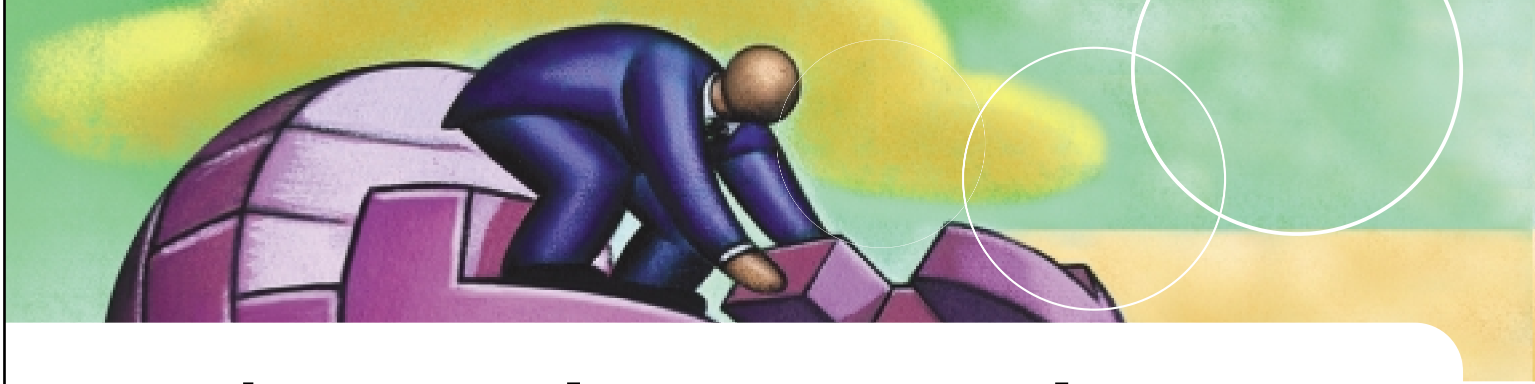
For subscriptions and requests for bulk orders,
please send your letters to Subscription Department.
SUBSCRIBE@SYS-CON.COM
Cover Price: $8.99/Issue Domestic: $149/YR (12 Issues)
Canada/Mexico: $169/YR Overseas: $179/YR
(U.S. Banks or Money Orders)
Back issues: $15 U.S. $20 All others

PRESIDENT AND CEO	FUAT A. KIRCAALI
COO/CFO	MARK HARABEDIAN
SENIOR VP, SALES & MARKETING	CARMEN GONZALEZ
VP, INFORMATION SYSTEMS	ROBERT DIAMOND
VP, SALES & MARKETING	MILES SILVERMAN
PRESIDENT, SYS-CON EVENTS	GRISHA DAVIDA
PRODUCTION CONSULTANT	JIM MORGAN
FINANCIAL ANALYST	JOAN LAROSE
ACCOUNTS RECEIVABLE	KERRI VON ACHEN
ACCOUNTS PAYABLE	BETTY WHITE
ADVERTISING ACCOUNT MANAGERS	MEGAN RING-MUSSA ROBYN FORMA
ASSOCIATE SALES MANAGERS	CARRIE GEBERT ALISA CATALANO KRISTIN KUHNLE
CONFERENCE MANAGER	MICHAEL LYNCH
ART DIRECTOR	ALEX BOTERO
ASSOCIATE ART DIRECTORS	LOUIS F. CUFFARI RICHARD SILVERBERG TAMI BEATTY
WEB DESIGNERS	STEPHEN KILMURRAY CHRISTOPHER CROCE
CONTENT EDITOR	LIN GOETZ
CUSTOMER SERVICE REPRESENTATIVE	MARGIE DOWNS
CUSTOMER RELATIONS MANAGER	RACHEL MCGOURAN

EDITORIAL OFFICES

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

SYS-CON MEDIA



Web Services Development with WSAD 5.0

New version offers improved development capabilities

– BY GREG FLURRY –

WebSphere Studio Application Developer (WSAD) version 5.0 is the latest version of IBM's J2EE e-business application development tool. WSAD supports all phases of Web service development: the initial development of components such as JavaBeans or Enterprise JavaBeans, the transformation of those components into Web services, the testing of the Web services, and the publication of the Web services in a UDDI-compliant registry. The focus of this article is Web services development and testing.

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

E-MAIL
flurry@us.ibm.com

In a previous article, "Web Services Development with WebSphere Studio Application Developer (*WSDJ* Vol. 1, issue 3)," I discussed Web services development in WSAD version 4.0.2. In this article I examine the improved Web services development capabilities of WSAD version 5.0.

This article creates a weather forecasting Web service using the same JavaBeans and resource files as the earlier article. I assume you are familiar with the basics of developing Java components in WSAD and will offer details only on the Web services–related aspects of WSAD version 5.0.

Getting Started

We'll assume that an existing Web project contains the JavaBeans and resources to be used for the weather forecast Web service (the source code for this article and my earlier article can be downloaded from www.sys-con.com/websphere/sourcec.cfm). Figure 1 shows the J2EE Navigator view for the WF Web project, which contains the five JavaBeans that contribute to the Web service:

- ***WeatherForecast:*** The Web service "business logic;" which accesses the National Weather Service (NWS), extracts forecast information, and formats the information appropriately
- ***States:*** Holds the states covered by the NWS
- ***Cities:*** Holds the cities for a particular state covered by the NWS
- ***Forecast:*** Holds the forecast for one period for a particular city and state combination
- ***Forecasts:*** Holds the forecasts for several periods for a particular city and state combination

Listing 1 shows the signature of the States bean, the Cities bean, the Forecast bean, and the Forecasts bean. See my earlier article or the source code for more detail. The three beans for the most part are standard data beans. The Cities bean and the Forecasts bean have "add" methods used by the WeatherForecast bean to insert information.

Listing 2 shows the signature of the WeatherForecast bean. Note that the getStates()method returns a States bean, the getCities()method returns a Cities bean, and the getForecast() method returns a Forecasts bean.

Creating a Web Service from the Bean

We will use the WSAD Web Service wizard to create the Web service from the WeatherForecast bean. To create the Web service, select the WeatherForecast bean and then right-click and select New>Other. You will see the dialog shown in Figure 2. You must select Web Services in the left panel and Web Service in the right panel. Now hit Next to go to the next dialog.

Now you will see the panel shown in Figure 3. We do not want to publish the Web service, so leave that box unchecked. We do want to generate a proxy to support testing, so check

that box. We do not want to test the generated proxy, so hit Next to go to the next panel.

In the resulting Deployment Settings panel, shown in Figure 4, you can choose the runtime protocol and target server type. At this time, the only protocol option is SOAP. There are a number of server options, including WebSphere versions 4 and 5. As indicated in the panel, the default protocol is SOAP and the default server is WebSphere v5, which is what we want, so simply leave the defaults selected. Hit Next to go to the next panel (not shown), which simply confirms that you've chosen the right JavaBean. Hit Next to go to the next panel.

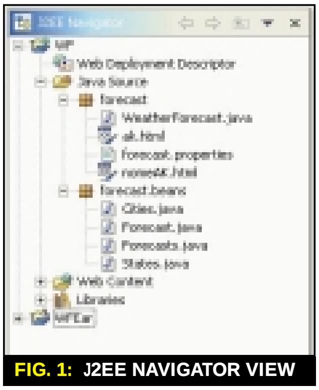


FIG. 1: J2EE NAVIGATOR VIEW

You should now see the Java Bean Identity panel, shown in Figure 5. If this were more than just an exercise, you would also want to ensure that the Web service URI is unique, but that is not necessary in this case. This panel allows the developer to use HTTP-like processing scope for a Web service, i.e., request, session, or application scope. The WeatherForecast bean is designed to operate with session scope, so you must change the Scope field to "Session". Setting the scope to Session ensures a new instance for each user, allowing caching of information from the NWS to work properly.

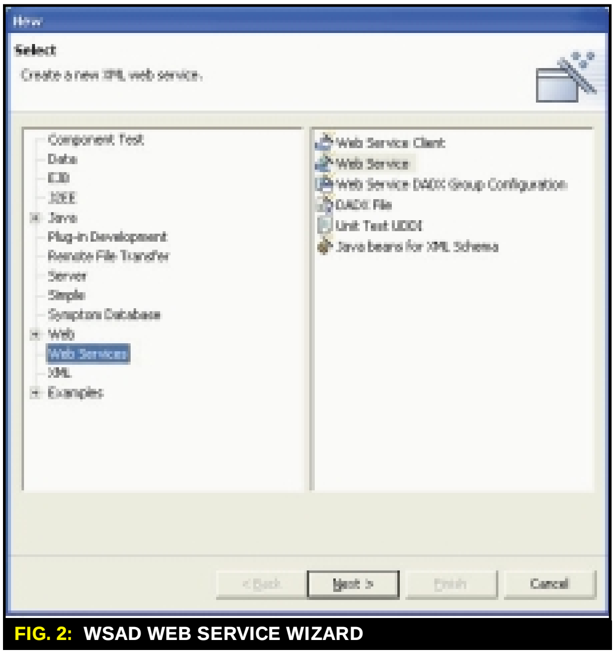
The WSDL document fields show new features in WSAD v5.0. The WSAD wizard now produces, by default, separate WSDL documents for the interface, binding, and the service WSDL elements. In fact, the wizard produces two different bindings, one for the protocol chosen in the previous step, in this case SOAP/HTTP, and one for a Java binding. This new default behavior enables advanced features beyond the scope of this article. If you want the wizard to generate a single WSDL document, simply use the same name in the interface, binding, and service documents. The default values of those fields are acceptable for this exercise, so select Next.

You will now see the JavaBean Methods panel, shown in Figure 6, which allows you to select which methods in a JavaBean to expose in the Web service. In this case, we want to expose all the methods, so no changes are needed. Notice that SOAP encoding is automatically selected for both the input strings and the output JavaBeans. This is exactly as desired, so select Next.

You will now see the Binding Proxy Generation panel, shown in Figure 7, which allows you to select which type of binding you want the generated proxy to use for the Web service. In this case, we want to use the SOAP binding, so no changes are needed.

Since we want to generate a SOAP-based Web service and the corresponding proxy, and we don't want to generate a test client for the proxy or publish information about the Web service to a UDDI registry, you can select Finish.

The Web Service wizard now generates the WSDL files describing the Web service, generates a SOAP deployment

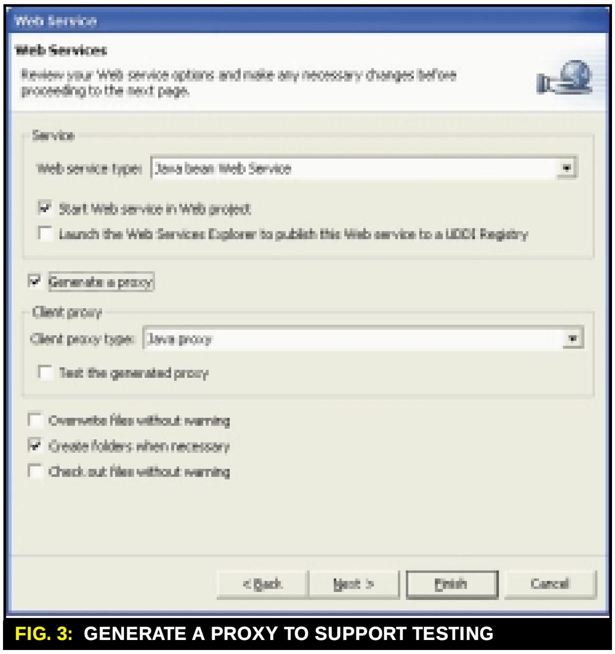


New

Select

Create a new XML web service.

FIG. 2: WSAD WEB SERVICE WIZARD



Web Service

Web Services

Review your Web service options and make any necessary changes before proceeding to the next page.

Service

Web service type: Java bean Web Service

☑ Start Web service in Web project

□ Launch the Web Services Explorer to publish this Web service to a UDDI Registry

☑ Generate a proxy

Client proxy

Client proxy type: Java proxy

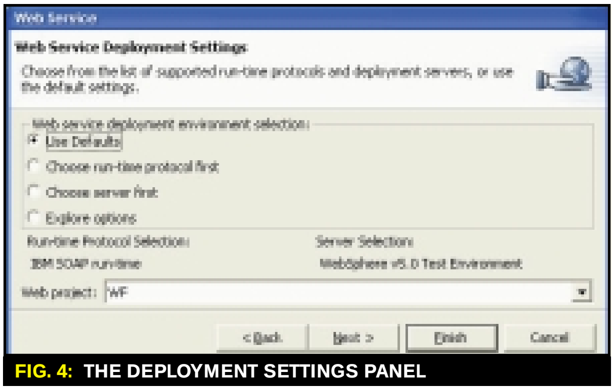
□ Test the generated proxy

□ Overwrite files without warning

☑ Create folders when necessary

□ Check out files without warning

FIG. 3: GENERATE A PROXY TO SUPPORT TESTING



Web Service Deployment Settings

Choose from the list of supported run-time protocols and deployment servers, or use the default settings.

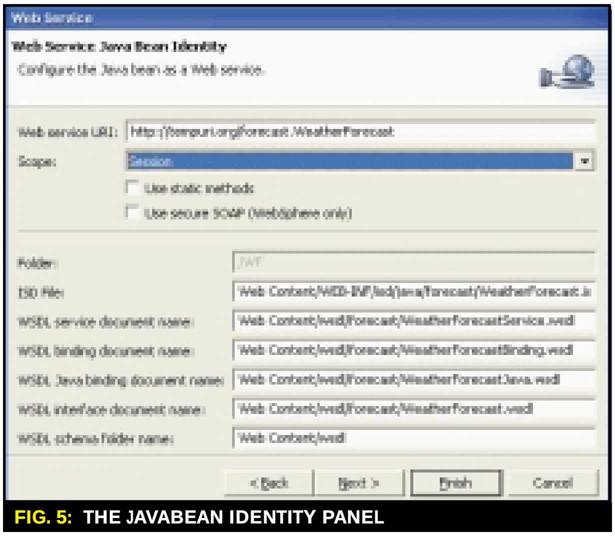
Web service deployment environment selection:
- ● Use Defaults
- ○ Choose run-time protocol first
- ○ Choose server first
- ○ Explore options

Run-time Protocol Selection: IBM SOAP run-time

Server Selection: WebSphere v5.0 Test Environment

Web project: WF

FIG. 4: THE DEPLOYMENT SETTINGS PANEL



Web Service Java Bean Identity

Configure the Java bean as a Web service.

Web service URI: http://tempuri.org/forecast.WeatherForecast

Scope: Session

- □ Use static methods
- □ Use secure SOAP (WebSphere only)

Folder: /WF

ISD File: Web Content/WEB-INF/isd/java/forecast/WeatherForecast.is

WSDL service document name: Web Content/wsdl/Forecast/WeatherForecastService.wsdl

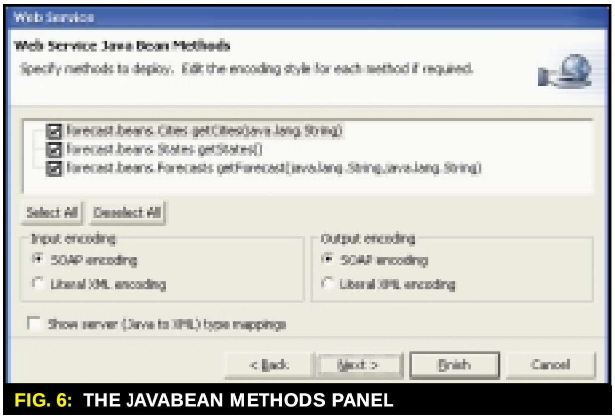
WSDL binding document name: Web Content/wsdl/Forecast/WeatherForecastBinding.wsdl

WSDL Java binding document name: Web Content/wsdl/Forecast/WeatherForecastJava.wsdl

WSDL interface document name: Web Content/wsdl/Forecast/WeatherForecast.wsdl

WSDL schema folder name: Web Content/wsdl

FIG. 5: THE JAVABEAN IDENTITY PANEL



Web Service Java Bean Methods

Specify methods to deploy. Edit the encoding style for each method if required.

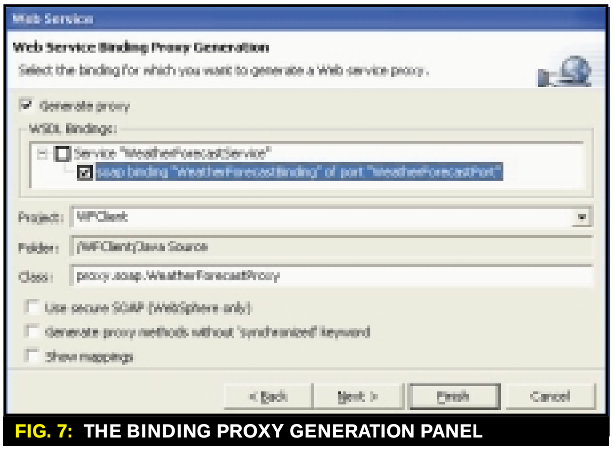
- ☑ forecast.beans.Cities getCities(java.lang.String)
- ☑ forecast.beans.States getStates()
- ☑ forecast.beans.Forecasts getForecast(java.lang.String,java.lang.String)

Input encoding
- ● SOAP encoding
- ○ Literal XML encoding

Output encoding
- ● SOAP encoding
- ○ Literal XML encoding

□ Show server (Java to XML) type mappings

FIG. 6: THE JAVABEAN METHODS PANEL



Web Service Binding Proxy Generation

Select the binding for which you want to generate a Web service proxy.

☑ Generate proxy

WSDL Bindings:
- □ Service "WeatherForecastService"
  - ☑ soap binding "WeatherForecastBinding" of port "WeatherForecastPort"

Project: WFClient

Folder: /WFClient/Java Source

Class: proxy.soap.WeatherForecastProxy

- □ Use secure SOAP (WebSphere only)
- □ Generate proxy methods without 'synchronized' keyword
- □ Show mappings

FIG. 7: THE BINDING PROXY GENERATION PANEL

descriptor for deploying the Web service in the WebSphere Application Server, deploys the Web service in the WebSphere Test Environment provided by WSAD, and starts the WebSphere Test Environment to make the Web service available for use.

Artifacts Generated by WSAD

Figure 8 shows the server-side artifacts generated by the WSAD Web services wizard. You can see the WSDL documents and associated schema documents generated from the WeatherForecast bean as part of creating the Web service. The WSDL is placed in the Web Content folder in the WF Web project.

By default WSAD generates a three-part WSDL description of the Web service, supported by schema definition files. The “interface” WSDL file describes the abstract interface definition. The “binding” WSDL file describes the binding of that definition to a concrete transport; the “service” WSDL file describes the actual URI for the Web service. It is instructive to examine the WSDL generated by WSAD to see how the returned beans are described.

Listing 3 shows an excerpt from the interface file, WeatherForecast.wsdl. The most important element in the file is the WSDL portType element. This excerpt shows only the getForecast operation included in the portType. Note that the interface file imports schema definition files that define the structure of the JavaBeans used in the interface.

The schema definition corresponding to the Forecast JavaBean (shown in Listing 4) is quite straightforward, containing a single complex type definition. Note that it and all the other JavaBean definitions are placed in the namespace “http://beans.forecast/”, which corresponds to the name of the package in which the original beans are defined.

Listing 5 shows excerpts of the schema definition corresponding to the Forecasts JavaBean. It defines a complex type corresponding to an array of the Forecast JavaBean and a complex type corresponding to the Forecasts JavaBean itself, which references the array complex type. The other schema definition files are similar to these two.

Listing 6 shows excerpts from the binding WSDL file, WeatherForecastBinding.wsdl. The key element is the WSDL binding element. The binding file imports the interface file described above to pick up the definition of the portType for the type attribute of the WSDL binding. Since this is a SOAP/HTTP binding, extensibility elements identify it as a SOAP binding and define operations using SOAP encoding. As before, only the getForecast operation is shown.

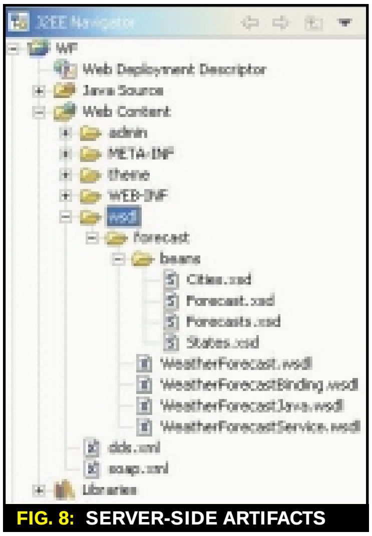
The service WSDL file, WeatherForecastService.wsdl (shown in Listing 7) imports the binding file to pick up the definition of the binding used in the binding attribute of the port element. Since it uses a SOAP/HTTP binding, the port element contains an extensibility element identifying the URI of the Web service. The key element in this file is the WSDL port element, a child of the WSDL service element.

WSAD 5.0 creates a new Web project, WFClient, to contain the artifacts generated (see Figure 9). The proxy, WeatherForecastProxy, is analogous to the WeatherForecast bean in the WF Web project. The new project also contains a set of “data” beans used by the proxy that are analogous to the data beans in the forecast.beans package of the WF Web project. These beans are generated solely from the WSDL without any knowledge of the implementation in the original package. This allows a client implementation to use the beans exactly as the original

beans would be used, but to be totally independent of the Web service implementation. Note in particular that the new beans have exactly the same package, as well as the same class names, as the original beans. This is a significant new feature of WSAD 5.0.

Listing 8 shows excerpts of the WSAD-generated proxy. Note that the signature of the proxy contains the same business methods as the Web service. Also note that the return values correspond to the client-side bean types generated by WSAD.

Listing 9 shows excerpts of the most complex of the client-side beans, forecast.beans. Forecasts. Note that it extends a special class, com.ibm.etools.xsd.bean.runtime. AnyType. This class allows generic handling of beans constructed from schema by WSAD. The other beans are constructed similarly. This capability is much improved in WSAD 5.0.



```
WF
  Web Deployment Descriptor
  Java Source
  Web Content
    admin
    META-INF
    theme
    WEB-INF
    wsdl
      forecast
        beans
          Cities.xsd
          Forecast.xsd
          Forecasts.xsd
          States.xsd
        WeatherForecast.wsdl
        WeatherForecastBinding.wsdl
        WeatherForecastJava.wsdl
        WeatherForecastService.wsdl
    dds.xml
    soap.xml
  Libraries
```

FIG. 8: SERVER-SIDE ARTIFACTS

Testing the Web Service and Proxy

TestWF (shown in Listing 10) is a simple test program for the Web service; it uses the proxy generated by the WSAD Web Service wizard. You should create the program in the default package in the WFClient project. You must add the variable "MAILJAR" to the Java Build Path property for the project to successfully compile and run the program.

You can run the program with the Web service configured to access the NWS over the Internet or to run against some "canned data" used when the Web service does not have access to the Internet (access is controlled by the OFFLINE property in the forecast.properties file). The test program calls all three operations on the Web service. The test program produces the following result when configured to actually contact the NWS.

```
The first state: AK
The first city: Anchorage
The first forecast: Wednesday night - Low 21,
20% chance of precipitation.
```

Using the Web Service

Now we will create a client of the Web service using only the WSDL, as would be done by finding the WSDL for a Web service in a UDDI registry. Navigate to and select the WeatherForecast-Service.wsdl file in the WF Web project. Right-click on it and select New>Other. In the resulting panel (see Figure 2), select Web Services on the left side and Web Service Client on the right and then select Next. You will see the Web Service Client wizard shown in Figure 10. Select Next. The resulting panel allows you to verify that you have selected the correct WSDL document. Select Next.

Now you will again see the Binding Proxy Generation panel shown in Figure 7, which allows you to select which type of binding you want for the Web service. In this case, we want the SOAP binding and proxy, so no changes are needed. However, we want to create a new client project, so in the Project field, enter "WFClientB" and then select Finish.

Figure 11 shows the results produced by WSAD. Notice that the results are identical to those produced above for the WFClient project used in testing the Web service (see Figure 9). This gives extra assurance that testing of the Web service and client artifacts by the provider will use the same artifacts used by the consumer of the Web service. This symmetry is new in WSAD 5.0.

You can copy the above test program into the default package in the WFClientB project. As before, you must add the variable MAILJAR to the Java Build Path property for the project. When you run the program, you will get identical results, assuming the same configuration.

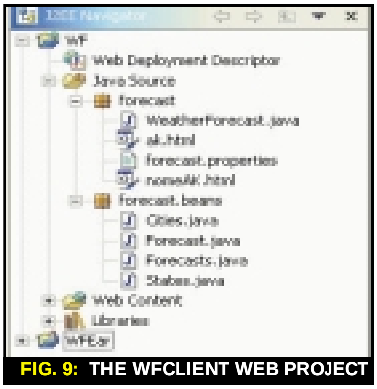
```
The first state: AK
The first city: Anchorage
The first forecast: Wednesday night - Low 21,
20% chance of precipitation.
```

Summary

This article showed that WebSphere Studio Application Developer version 5 Web service tools have improved the handling of somewhat complex parameters and return values over version 4. See the Readme file installed with WSAD and also the Mapping properties topic in the WSAD help information for limitations on the handling of simple and complex data types. If you find something WSAD cannot handle, you can always create your own serializers and deserializers; WSAD allows you to specify the (de)serializers in the Java to XML Mappings and XML to Java Mappings panels in the Web Service wizard, but that is beyond the scope of this article.

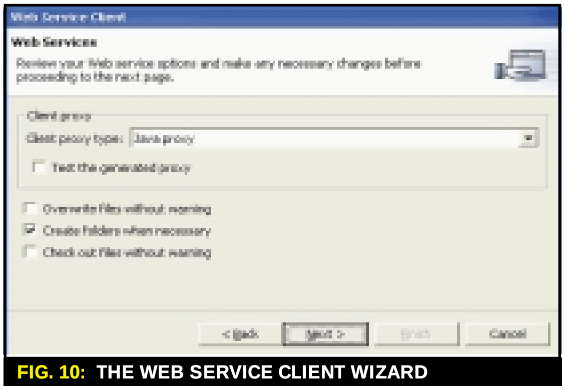
Reference

- *The Web Services Zone at IBM developerWorks:* www-106.ibm.com/developerworks/webservices/



```
WF
  Web Deployment Descriptor
  Java Source
    forecast
      WeatherForecast.java
      ak.html
      forecast.properties
      nomsAK.html
    forecast.beans
      Cities.java
      Forecast.java
      Forecasts.java
      States.java
  Web Content
  Libraries
WFEar
```

FIG. 9: THE WFCLIENT WEB PROJECT



Web Services

Review your Web service options and make any necessary changes before proceeding to the next page.

Client proxy

Client proxy type: Java proxy

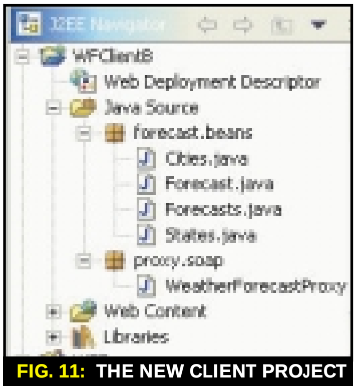
□ Test the generated proxy

□ Overwrite files without warning

☑ Create folders when necessary

□ Check out files without warning

FIG. 10: THE WEB SERVICE CLIENT WIZARD



```
WFClientB
  Web Deployment Descriptor
  Java Source
    forecast.beans
      Cities.java
      Forecast.java
      Forecasts.java
      States.java
    proxy.soap
      WeatherForecastProxy
  Web Content
  Libraries
```

FIG. 11: THE NEW CLIENT PROJECT

LISTING 1: THE SIGNATURE OF THE STATES BEAN

```
package forecast.beans;
 public class States {
      public String[] getStates() {...}
 }

package forecast.beans;
 public class Cities {

     public void setState(String state) {...}
     public String getState() {...}

     public void setCities(String[] cities) {...}
     public String[] getCities() {...}

     public void addCity(String city) {...}
 }

package forecast.beans;
public class Forecast {

     public void setPeriod(String p) {...}
     public String getPeriod() {...}

     public void setForecast(String f) {...}
     public String getForecast() {...}
 }

package forecast.beans;
 public class Forecasts {

     public void setLocation(String l) {...}
     public String getLocation() {...}

     public void setIssued(String i) {...}
     public String getIssued() {...}

     public void setForecasts(Forecast[] f) {...}
     public Forecast[] getForecasts() {...}

     public void addForecast(Forecast f) {...}
 }
```

LISTING 2: THE SIGNATURE OF THE WEATHERFORECAST BEAN

```
package forecast;
 import forecast.beans.*;
 public class WeatherForecast {
      public States getStates() {...}
      public Cities getCities(String state) {...}

     public Forecasts getForecast(String state, String city)
{...}
 }
```

LISTING 3: EXCERPT FROM WEATHERFORECAST.WSDL

```
<?xml version="1.0" encoding="UTF-8"?>
 <definitions name="WeatherForecast"
     targetNamespace="http://forecast.wsdl/WeatherForecast/"
     ...
     xmlns:xsd1="http://beans.forecast/">
     <import location="beans/Cities.xsd"
     namespace="http://beans.forecast/"/>
     <import location="beans/Forecasts.xsd"
     namespace="http://beans.forecast/"/>
     <import location="beans/Forecast.xsd"
     namespace="http://beans.forecast/"/>
     <import location="beans/States.xsd"
     namespace="http://beans.forecast/"/>
     <message name="getForecastRequest">
         <part name="state" type="xsd:string"/>
         <part name="city" type="xsd:string"/>
     </message>

<message name="getForecastResponse">
<part name="result" type="xsd1:Forecasts"/>
     </message>
     ...
     <portType name="WeatherForecast">
         <operation name="getForecast" parameterOrder="state
            city">
             <input message="tns:getForecastRequest"
               name="getForecastRequest"/>
             <output message="tns:getForecastResponse"
               name="getForecastResponse"/>
         </operation>
         ...
     </portType>
 </definitions>
```

LISTING 4: THE SCHEMA DEFINITION CORRESPONDING TO THE FORECAST JAVABEAN

```
<?xml version="1.0" encoding="UTF-8"?>
 <schema attributeFormDefault="qualified"
     elementFormDefault="unqualified"
     targetNamespace="http://beans.forecast/"
     xmlns="http://www.w3.org/2001/XMLSchema"
      xmlns:xsd1="http://beans.forecast/">
     <complexType name="Forecast">
         <all>
             <element name="forecast" nillable="true"
               type="string"/>
             <element name="period" nillable="true"
               type="string"/>
         </all>
     </complexType>
 </schema>
```

LISTING 5: EXCERPTS OF THE SCHEMA DEFINITION CORRESPONDING TO THE FORECASTS JAVABEAN

```
<?xml version="1.0" encoding="UTF-8"?>
<schema attributeFormDefault="qualified"
     elementFormDefault="unqualified"
     targetNamespace="http://beans.forecast/"
     ...
     xmlns:xsd1="http://beans.forecast/">
     <include schemaLocation="Forecast.xsd"/>
     ...
     <complexType name="ArrayOfForecast">
         <complexContent>
             <restriction base="soapenc:Array">
                 <sequence/>
                 <attribute ref="soapenc:arrayType"
                    wsdl:arrayType="xsd1:Forecast[]"/>
             </restriction>
         </complexContent>
     </complexType>
     <complexType name="Forecasts">
         <all>
             <element name="location" nillable="true"
                type="string"/>
             <element name="forecasts" nillable="true"
                type="xsd1:ArrayOfForecast"/>
             <element name="issued" nillable="true"
                type="string"/>
         </all>
     </complexType>
 </schema
```

LISTING 6: EXCERPTS FROM THE BINDING FILE, WEATHERFORECASTBINDING.WSDL

```
<?xml version="1.0" encoding="UTF-8"?>
 <definitions name="WeatherForecastBinding"
targetNamespace="http://forecast.wsdl/WeatherForecastBinding/

xmlns:interface="http://forecast.wsdl/WeatherForecast/"
```

```
...
xmlns:tns="http://forecast.wsdl/WeatherForecastBinding/">
     <import location="WeatherForecast.wsdl"
        namespace="http://forecast.wsdl/WeatherForecast/"/>

   <binding name="WeatherForecastBinding"
      type="interface:WeatherForecast">
         <soap:binding style="rpc"transport="http:
         //schemas.xmlsoap.org/soap/http"/>
         <operation name="getForecast">
             <soap:operation soapAction="" style="rpc"/>
<input name="getForecastRequest">
                 <soap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://tempuri.org/forecast.WeatherForecast"
                     parts="state city" use="encoded"/>
             </input>
             <output name="getForecastResponse">
                 <soap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://tempuri.org/forecast.WeatherForecast"
  use="encoded"/>
             </output>
         </operation>
         ...
     </binding>
 </definitions>
```

LISTING 7: THE WSDL FILE, WEATHERFORECAST SERVICE.WSDL

```
<?xml version="1.0" encoding="UTF-8"?>
 <definitions name="WeatherForecastService"
targetNamespace="http://forecast.wsdl/WeatherForecastService/
"xmlns="http://schemas.xmlsoap.org/wsdl/"
 xmlns:binding="http://forecast.wsdl/WeatherForecastBinding/"
     xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
       xmlns:tns="http://forecast.wsdl/WeatherForecast
       Service/">
     <import location="WeatherForecastBinding.wsdl" name
       space="http://forecast.wsdl/WeatherForecastBinding/"/>
     <service name="WeatherForecastService">
         <port binding="binding:WeatherForecastBinding"
            name="WeatherForecastPort">
             <soap:address location="http://localhost:
             9080/WForecast/servlet/rpcrouter"/>
         </port>
     </service>
 </definitions>
```

LISTING 8: EXCERPTS OF THE WSAD-GENERATED PROXY

```
package proxy.soap;
 ...
 public class WeatherForecastProxy {
   public synchronized void setEndPoint(URL url) {...}
   public synchronized URL getEndPoint() {...}
   public synchronized forecast.beans.Cities
     getCities(java.lang.String state) {...}
   public synchronized forecast.beans.Forecasts

getForecast(java.lang.String state,
java.lang.String city) {...}

   public synchronized forecast.beans.States getStates()
{...}
 }
```

LISTING 9: EXCERPTS FROM FORECAST. BEANS.FORECASTS.

```
package forecast.beans;
import com.ibm.etools.xsd.bean.runtime.AnyType;
 ...

 public class Forecasts extends AnyType {
   public Forecasts() {

   addElement("location", java.lang.String.class);
   addElement("forecasts", Forecast[].class);
     addElement("issued", java.lang.String.class);

  }

   public String getLocation() {
     return (String)this.basicGet("location", 0);
   }
   public void setLocation(String location) {
     this.basicSet("location", 0, location);
   }

   public Forecast[] getForecasts() {
     return (Forecast[])this.basicGet("forecasts", 0);
   }
   public void setForecasts(Forecast[] forecasts) {
     this.basicSet("forecasts", 0, forecasts);
   }

   public String getIssued() {
     return (String)this.basicGet("issued", 0);
   }
   public void setIssued(String issued) {
     this.basicSet("issued", 0, issued);
   }
 }
```

LISTING 10: TESTWF, A SIMPLE TEST PROGRAM

```
public class TestWF {

     public static void main(String[] args) {
         try {
             proxy.soap.WeatherForecastProxy wf = new
               proxy.soap.WeatherForecastProxy();

             forecast.beans.States  states = wf.getStates();
             String[] a = states.getStates();
             System.out.println("The first state: "+ a[0]);

             forecast.beans.Cities cities =
                wf.getCities("AK");
             a = cities.getCities();
             System.out.println("The first city: "+ a[0]);

             forecast.beans.Forecasts forecasts =
                wf.getForecast("AK", "Nome");
             forecast.beans.Forecast[] fA =
                forecasts.getForecasts();
             System.out.println("The first forecast: "+
               fA[0].getPeriod()
                  + " - " + fA[0].getForecast());

         } catch (Exception ex) {
              System.out.println(ex.getMessage());
         }
     }
 }
```

Adding value to the business

Automating Business Processes with WSFL and BPEL4WS

BY GRANVILLE MILLER

As we strive to add more value to the business as computing professionals, the marriage of business processes and Web services provides opportunities to get closer to our business colleagues.

ABOUT THE AUTHOR

Granville Miller has over 15 years of experience in the object-oriented community. He is coauthor of the Advanced Use Case Modeling series and *A Practical Guide to Extreme Programming*. Granville currently directs strategy for the TogetherSoft line of products.

E-MAIL

rmiller@togethersoft.com

The business process scripting languages, Web Services Flow Language (WSFL) and Business Process Execution Language (BPEL), provide a mechanism for integrating Web services to form cohesive business processes. The first step is to create models of your business. This is followed by a combination of development and business process engineering activities. Simulation of the business process can be used to "try out" the new process. Finally, J2EE technology can be used to roll out scalable, enterprise-level applications.

The real value behind any new technology comes from its ability to help us solve real problems. Technology provides its greatest value when it can be used to solve whole classes of problems across many diverse areas. One of the recent additions to our technology repertoire is Web services. The technology behind Web services is a set of XML-based access and description languages. However, the idea behind Web services is much more than description and access technologies. The intent is that they are coarse-grained services that provide tangible business value.

Originally, Web services were desired as a method to make business services available externally. Certainly Web services could be made available via a universal directory where people could discover and utilize these services over the Internet. However, Web services are more likely to be found internally as initial applications of this technology tend to focus on automating business processes.

Much of the automation of today's business processes can be found in a combination of systems. Some may be legacy systems, written using outdated software technology. Packaged solutions that provide interfaces to generic business services may automate another part of a business process. Finally, there may be areas of a business process for which existing systems cannot provide automation. These areas must be written as part of new business initiatives.

Web service technology can provide a ubiquitous interface to all of these systems. As a result, Web service technology provides the ultimate "glue" for integrating legacy systems, packaged solutions, and new technologies into solutions that automate or partially automate business processes. The concept of assembling a set of Web services to provide a solution that meets a business process need is called *Web service orchestration*.

Web service orchestration knits together a series of Web services to provide support for a business process. Each Web service provides support for an activity within the workflow. As each activity is completed, the flow provided moves the user to the next activity. A path through a set of activities is followed until the business process is completed. Multiple paths may reflect exceptions that can occur and the remedies for those exceptions if an activity is not completed successfully.

This sequence of activities can be depicted using a UML activity diagram (see Figure 1). In the activity diagram, the boxes represent the activities, while the links between the activities represent the flow between them. Activity diagrams are commonly used to represent flow between activities in business processes. Activity diagrams are a visual representation of a business process that is easy to understand and communicate.

Flow Languages

To make Web service orchestration easier, IBM created WSFL, which connects Web services to provide Web service orchestration. It is an XML-based language that provides the ability to describe workflows and to call Web services to orchestrate the workflow. The idea behind WSFL is that it is easier to build flows in a language specifically meant for this (especially when it is driven pictorially), rather than a high-level multipurpose language such as Java.

Of course, WSFL becomes Java

when it is deployed to WebSphere, but the implementation details are unimportant to those seeking to rapidly deliver business solutions in this fashion. Instead, WSFL can be used to quickly prototype the business process changes necessary to continually engineer business processes. New Web services can be written to fill out these new processes in high-level languages such as Java, supporting the newly engineered activities.

A succeeding language to WSFL is Business Process Execution Language for Web Services (BPEL4WS). This new flow language is a joint collaboration between IBM, BEA, and Microsoft. The basic ideas behind BPEL4WS are similar to those behind WSFL and the Microsoft flow language, XLANG. Though both WSFL and BPEL4WS are XML-based, their syntax is slightly different.

Flow Language Concepts

Let's look at a simple business process to understand workflow and flow languages. Let's consider a simple loan processing process. We can divide loan processing into the following activities:

- Submit loan request
- Validate credit references
- Evaluate loan request
- Book loan

There are many types of people or roles involved in this high-level business process. The applicant performs the first activity, submit loan request. The loan clerk validates the credit references and books the loan. The loan officer evaluates the loan request. Each of these activities would normally be divided into small subactivities. However, for the purposes of this example, let's assume that these are each atomic (nondecomposable) activities with simple succeed or fail results.

In WSFL, the roles are called service providers (partners in BPEL4WS). Service providers are the class of people responsible for performing the given activity. The activities in the business process are called activities in WSFL, and we always associate a role with each activity to show who performs it. The initial WSFL for the simple loan processing process is shown in Listing 1.

In our implement section, we would specify the name of the Web service that implements the activity.

This flow definition defines the service providers and the activities, but it does not define the flow of activities within the process. Control flow is determined by WSFL structures called control links and data flow is determined by data links. A control link specifies a source and a target activity and links the two activities together. The control flows from the source to the target, setting up a sequential flow of control across the activities.

```
<controlLink
source="submitLoanRequest"
target="validateCreditRef
erences" />
```

Data links show how data moves from activity to another. For example, the loan application is clearly used in all of the activities. The application is created in the first activity and is propagated to the next. The WSFL would look like:

```
<dataLink
source="submitLoanRequest"
target="validateCreditRef
erences">
    <map
sourceMessage="loanApplicati
on" targetMessage="
loanApplication" />
</dataLink>
```

Conditional control and data links allow multiple flows across the activities. This handles the many exception conditions that arise in normal business processes.

Implementation

There are many ways to invoke Web services using flow languages. The most straightforward is to invoke a Web service as a remote procedure call in a synchronous manner. Activities can be synchronous when they can be processed in a reasonable amount of time. However, there are often activities whose response cannot be determined in what we would consider a reasonable amount of time. They may have dependencies on other activities, such as a loan officer approval. Activities with dependencies or that require larger amounts of time to complete may require a flow language to exhibit asynchronous behavior.

In BPEL4WS, there are three ways that things can be done inside a

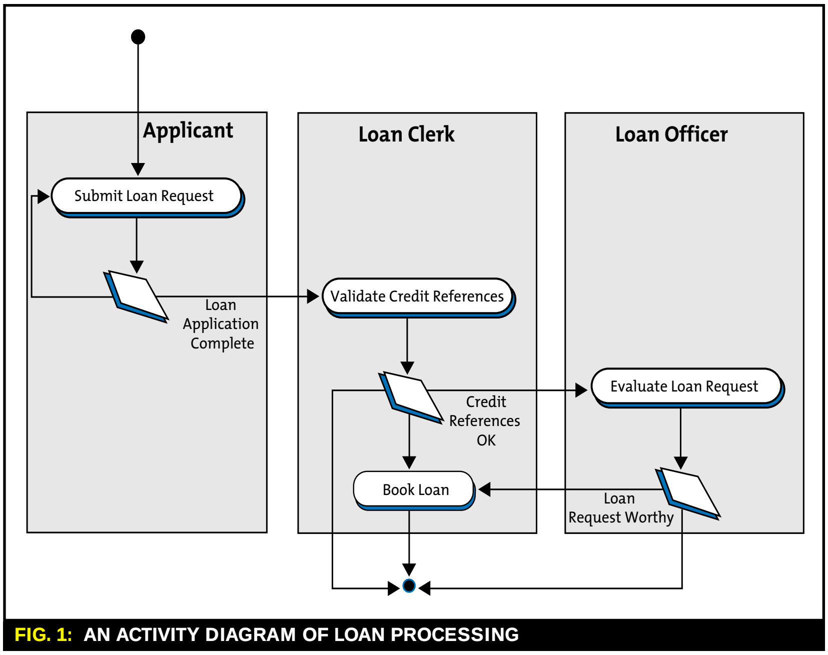


FIG. 1: AN ACTIVITY DIAGRAM OF LOAN PROCESSING

flow. A flow can invoke a Web service, send a message (called a reply), or receive a message. Invoking a Web service corresponds to performing an activity. Sending messages and receiving replies allows activities to be performed concurrently or asynchronously.

The send and receive behavior also allows multiple roles to be coordinated. As loan requests are submitted by customers, credit references can be validated by the clerks. The process moves this way through its transitions until the final loan information is replied back to the customer. The corresponding BPEL4WS code looks like that in Listing 2.

Turning Into Java

WSFL and BPEL4WS were meant to be deployed to application servers such as WebSphere. The common complaint with scripting languages is their speed and scalability. This area is covered with these two flow languages. An interesting feature of flow languages is their recursive composition, or their ability to become Web services themselves. Each flow is itself a Web service and can be called by another flow. This allows processes to be decomposed into subprocesses that can be decomposed into activities.

A subprocess can then be rewritten in Java and exposed as a Web service for the flow language to call. The idea is that you break Web service flows into granular pieces and if performance becomes an issue, build them into the coarse-grained flows that perform best on the server side.

Finally, business process simulation is always a good idea when engineering your business process. Web services can quickly be made in Java and "stubbed" into a new business process to "try it out." In this way, we can prototype new business processes without a whole lot of investment. Rapid application development products such as WebSphere Studio Application Developer IE and Together WebSphere Edition can help you model and build business process prototypes quickly and easily.

Conclusion

The climate for business process modeling and development is rapidly converging on the set of tools necessary to deliver business value in the Web services arena. Paramount to this process are the Web service flow languages that allow rapid development of business processes using Web service components. These languages can be utilized to engineer business processes for more efficient operations.

References

- Curbera, F.; Goland, Y.; Klein, J.; Leymann, F.; Roller, D.; Thatte, S.; Weerawarana, S. *Business Process Execution Language for Web Services, Version 1.0.* www-106.ibm. com/developerworks/webservices/library/ws-bpel
- Leymann, F. *Web Services Flow Language (WSFL 1.0).* www-3.ibm.com/software/solutions/webservices/pdf/WSFL.pdf

LISTING 1: WSFL FOR LOAN PROCESS

```
<flowModel name ="loanProcessingFlow" serviceProviderType = "loanProcessing">
       <serviceProvider name="applicant" type="applicant" />
       <serviceProvider name="loanClerk" type="loanClerk" />
       <serviceProvider name="loanOfficer" type="loanOfficer" />

        <activity name="submitLoanRequest">
             <performedBy serviceProvider="applicant" />
              <implement>
                      ..
               </implement>
         </activity>

        <activity name="validateCreditReferences">
             <performedBy serviceProvider=" loanClerk" />
              <implement>
                      ..
               </implement>
         </activity>

        <activity name="evaluateLoanRequest">
             <performedBy serviceProvider="loanOfficer" />
              <implement>
                      ..
               </implement>
         </activity>

        <activity name="bookLoan">
             <performedBy serviceProvider="loanClerk" />
              <implement>
                      ..
               </implement>
         </activity>
</flowModel>
```

LISTING 2: CORRESPONDING BPEL4WS FOR LOAN PROCESS

```
<sequence>

     <invoke partner="applicant"
                  portType="lns:submitPT"
                  operation="submitLoanRequest"
                  container="LoanRequest">
      </invoke>

     <reply partner="loanClerk"
                portType="lns: validatePT"
                operation="validateCreditReferences"
                container="LoanRequestFault"
                fault="cannotValidateReferences"/>
</sequence>
```

Best option for flexibility and performance

The Aggregate Entity Pattern Makes a Case for Using BMP

BY KEN MITSUMOTO

Entity beans have been much maligned lately, largely for their inability to scale and difficulty to develop and maintain. However, there remain some compelling reasons to consider using entity beans to persist data in a J2EE application. For example, survivability of entity beans, where state can be recovered following a hard crash, lends to increased reliability when the server is running.

The serialization of state to an alternate store allows entity beans to utilize limited resources more efficiently. That is, activating and passivating beans can allow a server to complete a transaction even when its headroom has been exceeded. A developer can mimic such facilities in session beans, but why reinvent the wheel? Of course, the most significant difference between entity beans and session beans is conceptual, not concrete. Whereas a single session bean instance serves only one client, an entity bean, by virtue of a primary key, serves many clients. Entity beans are more conducive to an architecture where the persistence layer of the application is distributed.

Entity bean evangelism is not the point of this article. The intent here is to promote the use of coarse-grained entity beans when entity beans make sense. A fine-grained bean creates a bijection, or one-to-one relationship, between a bean and a single row in a single table. An architecture built on fine-grained beans requires as many beans as there are tables – not exactly efficient. A coarse-grained bean, on the other hand, draws data from multiple tables. The implication is that a coarse-grained entity bean can persist an entire object graph. Effectively used, the number of entity beans required to model a complex schema can be reduced to a handful. Designing entity beans in this fashion follows the Aggregate Entity pattern, part of the Sun Java Center J2EE Patterns. The benefits of this pattern are explained well on the Sun Java Center site (http:// developer.java.sun.com/ developer/ technicalArticles/J2EE/patterns). Briefly:

- Reduced network traffic between all tiers
- More efficient utilization of resources
- Simplified client interface to data
- Improved manageability

As you would expect, these benefits do not come for free. Those familiar with the two paradigms of entity beans, container-managed persistence (CMP) and bean-managed persistence (BMP), know that the Aggregate Entity pattern cannot be implemented with CMP. For many, this is a deal-breaker. It doesn't have to be.

The release of WebSphere 5.0 has many developers looking forward to using CMP 2.0. The promise of robust persistence and relationship management, coupled with superior performance, seems to obviate the need for BMP. However, in terms of flexibility and performance, BMP remains the best option. The server-implemented persistence in CMP is still limited. Relationship management lacks granularity. The performance improvements are only the result of application server vendors implementing a few tricks, such as optimizing data retrieval for finder operations.

There is no reason why such optimizations cannot, and in fact, should not, be implemented in a BMP bean. Beyond these shortcomings, CMP's Achilles' heel is the inability to draw data from more than one table. The current CMP specification precludes coarse-grained entities. As I have briefly discussed, in a fine-grained architecture, each bean is tied to a single table, rigidly binding the application to a schema. The client interface to the data becomes a mess of remote interfaces and disparate value objects. Any operation that draws data across more than one bean issues a cascade of ejbLoad() and ejbStore() calls, each running roundtrip to the database. The container must allot precious resources to all the bean implementation classes, their remote objects, and primary key classes. The Aggregate Entity pattern resolves these issues by providing one value object, one remote interface, one bean.

ABOUT THE AUTHOR
Ken Mitsumoto is a senior sales engineer for THOUGHT, Inc. (www.thoughtinc.com). He has been working with J2EE technologies since their inception. Ken has a background in mathematics and likes very fast motorcycles.

E-MAIL
ken.mitsumoto@thoughtinc.com

If the Aggregate Entity pattern has a flaw, it's that it is difficult to implement. The difficulty lies in managing an object graph efficiently within the bean. There are two primary hurdles: state management, and relationship management. With proper state management, the bean is always aware of the state of the graph. Because the bean provides an object view of data to its client, the bean is responsible for translating changes to the graph into SQL. The algorithm that resolves these changes should be intelligent enough to issue SQL only for objects that have changed. Efficiency mandates the removal of all unnecessary trips to the database. Moreover, how these selective operations are executed is important. Locking strategies and graceful exception handling help maintain data validity, another primary directive. Equally challenging, relationship management addresses the issues of how the objects in the graph interact. Granular control over cascade behavior, such as lazy loading, is critical to performance, especially with large datasets. There is no reason to carry around more data than you need. Supporting patterns such as inheritance and polymorphism is conducive to good object-oriented design. The inability to represent these patterns in the persistence layer should not be a constraint on the object model.

I will not discuss the deeper specifics of implementing the Aggregate Entity pattern in this short article. The reasoning is that even with instructions spelled out, use of this pattern would still be too risky for most managers. Entity beans have been branded with two common complaints: "They don't perform," and "They're too hard." The developer's choice is often reduced to one or the other. CMP is easy, but its performance is poor. BMP using the Aggregate Entity pattern performs well, but is difficult. With the current EJB specification, CMP is not a good performance option. The logical alternative is to consider ways to make the Aggregate Pattern easier to implement.

There is a way. Developers are always asking, "What good are Object to Relational mapping tools?" Object graph management in entity beans would be a great answer. Some of these tools even allow developers to generate entire entity beans that implement the Aggregate Entity pattern with a click of the mouse. Moreover, these tools decouple the application from the schema, allowing for new possibilities in design, implementation, and management. If entity beans seem like the right solution for your project, one of these tools can be a low-cost, quick time-to-market solution to implement Aggregate Entity pattern BMP with the ease of CMP.

A Conversation with IBM's Suchitra Joshi

The Competitive Technology Labs' road show brings its message to the world

Suchitra Joshi, a senior project manager for Competitive Technology Labs, talks with Jack Martin, editor-in-chief of ***WebSphere Developer's Journal****, about "Winning with WebSphere," IBM's technical seminar focusing on WebSphere, its related products, and e-business.*

WSDJ: TELL ME A LITTLE BIT ABOUT WHAT YOU DO AT IBM.

SJ: My main purpose is to stage a competitive technical seminar called "Winning with WebSphere," all over the world. These seminars are all about WebSphere and its family of products. And it's not just WebSphere; we are beginning to incorporate more and more material on other arms of the software group, such as Domino, Lotus, and Tivoli. We show how these IBM middleware products come together for an enterprise or a company to do e-business. In this one-day seminar we also show competing products and prove through our live demonstrations and presentation material that the WebSphere family of products is a far superior solution for any company to do e-business. You will hear about our competitors such as Microsoft, TIBCO, BEA, and others.

WSDJ: CAN YOU DESCRIBE THE COMPONENTS OF A TRUE E-BUSINESS?

SJ: Come to our seminar and you will learn about them for yourself – that's what the seminar is all about. Today, e-business embodies all aspects of your company's business processes. Even in the ever-growing IT marketplace business initiatives have not changed. What's changed is the way we achieve the end goal. Business ambitions have grown at a faster pace. To achieve these ambitious plans, a company needs better technology that will help integrate their Web and non-Web applications from start to finish. E-business helps you bring that concept to reality.

WSDJ: FOCUSING ON WEBSPHERE SPECIFICALLY, HOW DO YOU TOUCH WEBSPHERE?

SJ: We touch on all aspects of the WebSphere family, starting with the basics like the J2EE, W3C (XML) and Web services standards, which are a must in today's complicated IT marketplace, where a vast array of products are available. We show you the advantages of open standards versus proprietary solutions such as Microsoft's .NET. We then dive into technical details, covering WebSphere Application Server and WebSphere Studio Development Tools. We prove to you that WebSphere Application Server is a better foundation for your e-business platform than BEA's WebLogic. We show through live demonstrations that WebSphere Studio tools will improve your developers' productivity when compared to Microsoft and BEA tools. We have the numbers to prove this to you.

We then discuss products such as WebSphere Portal, WebSphere Personalization, Translation, Voice, Commerce Suite, etc., and last, but certainly not least, touch on products that are used for business integration to integrate your back-end systems with your front-end systems. To summarize, we truly make our audience understand how and why a company can do e-business using the WebSphere family of products.

WSDJ: SO IF I WENT TO ONE OF YOUR SEMINARS, WOULD I SEE AN ACTUAL WEBSPHERE ENVIRONMENT RUNNING?

SJ: Absolutely. Our speakers and demonstrators are experts in many aspects of WebSphere and products from competitors such as Microsoft, TIBCO, BEA. We carry a set of about five ThinkPads with us that have not only WebSphere products installed, but also have some of the competition's products. For example, in one of our past seminars, we had a BEA WebLogic Server and a WebSphere Application Server on those ThinkPads.

Typically, for a given presentation, after a set of foils are covered we transfer the microphone to a demonstrator sitting at the ThinkPad table, to show a demo of the topic being discussed – seeing is believing! These demos are not just done in a bland product demo format; our developers have created programs or demos that will highlight for the audience how a certain aspect of a WebSphere product works. Another demo might show how the competition's product fails to work. Together, these demos cover topics ranging from application servers all the way to integrating several internal and external business processes.

WSDJ: CAN YOU GIVE SOME EXAMPLES OF AREAS IN WHICH WEBSPHERE IS CLEARLY SUPERIOR TO THE COMPETITION, SAY BEA AND .NET?

SJ: As you already know, the WebSphere family of products helps businesses realize the promise of e-business. Hence, the superiority of WebSphere is seen in many of its products, such as the application server, the tools, message queuing, etc.

I am impressed with the way WebSphere Application Server handles scalability issues as compared to BEA's WebLogic Server. If one of the WAS servers in a cluster is down, WebSphere detects it and will try to bring up that server automatically. This saves time and money, especially if it's past midnight and no administrator is available to monitor these machines.

WebSphere Portal will let you create different kinds of portals depending on who your audience is. This is a very powerful concept – especially when information is at your employees' or customers' fingertips – and the turnaround time for a company can be shortened by implementing these technologies within an enterprise. The multidevice enablement makes IBM's Portal server advantageous because users want access anyplace and in any format.

WSDJ: THERE ARE A LOT OF SEMINARS IN THE WORLD ABOUT COMPUTERS AND INFORMATION TECHNOLOGY. WHAT IS THE KEY REASON THAT SOMEONE SHOULD COME TO ONE OF YOUR SEMINARS?

SJ: This seminar is absolutely unique, lively, and informative. Everyone in the audience will go home thinking, "Gee, I didn't know about that aspect of the product." The seminar brings all the pieces of the e-business puzzle together for the audience. A company or enterprise can be in the starting stages of doing business over the Internet or may have already adopted certain e-business technologies and now wants to grow further. For both kinds of companies, we have valuable information to share.

The seminar will never put you to sleep. In a typical seminar if you are just listening to information for six hours a day, toward the end of the seminar you will start dozing off. In our seminar you will retain the majority of the information because not only do we show it to you through demonstrations, but we show it in a very interesting fashion. For the seminar, we've created a fictitious company called Friendly Finance, and we show you how this Friendly Finance becomes an e-business company, and how you can touch several aspects, several departments of this Friendly Finance enterprise, whether you're building portals for B2B, B2C, or B2E; or you're personalizing your Web site; or you're trying to automate some of your business processes such as a banking application serving requests for information. It does not matter what size your company is – large enterprise or small to medium-size company, or even if you are just a developer scouting around for new technical information – the seminar has something for everyone.

WSDJ: OBVIOUSLY, PEOPLE LOOKING TO BUY IBM TECHNOLOGY CAN GO TO YOUR SEMINAR, BUT FOR SOMEONE WHO WANTS TO GET A BACKGROUND ON HOW THE ENTIRE E-BUSINESS PARADIGM WORKS, FROM A TO Z, IT SOUNDS LIKE THE WHOLE THING IS THERE.

SJ: Exactly! They also get an understanding of what the competition's products can and cannot do. Sometimes a salesman from another company may not reveal the whole truth about their product. We take the time to compare apples to apples.

The other interesting aspect of the seminar that I want to point out is that the seminar encapsulates all the aspects of the e-business marketplace, whether you are a line-of-business person within your company trying to understand technical details of a product; an executive wanting to make decisions on return on investment or productivity gains or trying to see the big picture of an e-business platform; an architect trying to understand which products are scalable and manageable without giving up performance; or a project manager trying to contain costs for your project or seeing which products will help you complete the project on time.

WSDJ: WHAT IS IBM'S GOAL IN HAVING THESE SEMINARS?

SJ: IBM's goals in these seminars is to showcase IBM's WebSphere technologies but also to compare them to our competition and let the audience decide who is the winner. We have overtaken BEA in the application server market and have seen tremendous growth in business integration. Most of our competitors' revenues and market shares have actually dropped. Through this seminar IBM intends to prove to the world that our technologies are top-notch for enterprise and SMB businesses.

WSDJ: SO WHEN AND WHERE ARE THESE SEMINARS GOING TO BE?

SJ: Your readers can go to www-3.ibm.com/software/info1/websphere/partners/ index.jsp?tab=ed/competitiveroadshow to find the dates for cities all over the world. Last year we reached an audience of close to 19,000 people. Our one-day road show seminar alone drew close to 14,000 people. The seminar is open and at no cost to the attendee.

WSDJ: IS THERE ANYTHING YOU'D LIKE TO CLOSE WITH?

SJ: Come and see for yourself the many advantages of going with the leader in middleware technologies. You will not be disappointed, I promise you. The seminar is open and at no cost to the attendee. Come and spend a day with some IBMers and IBM software and hardware.

Helping to glue together the enterprise

An LTPA Custom User Registry

BY MARCEL HEIJMANS

A three-letter acronym seen a lot recently is EAI (Enterprise Application Integration). EAI deals with the question of how to create a coherent enterprise system infrastructure within a heterogeneous application environment. One of the effects of mergers and the lack of standards for system integrators is the enormous amount of work it generates to glue all the parts of merging enterprises together. The major challenge is to do this in a logical manner while avoiding inconsistencies.

ABOUT THE AUTHOR

Marcel Heijmans is a senior software engineer and founder of Mnemonics. He created the J2EE Development Coaching concept, which trains and supports novice developers and architects within their projects while minimizing the project risks.

E-MAIL

marcel.heijmans@mnemonics.nl

The aspect of EAI that I want to focus on is security, specifically the authentication and authorization of users. The J2EE platform requires user information in order to authenticate a user or group of users and to authorize access to a J2EE component or a Web resource. However, there is no consensus on where or how to store employee or customer information. Databases, LDAP (Lightweight Directory Access Protocol), or even the operating file system are used to store user information.

The WebSphere Security Center

The WebSphere Application Server Administrator's Console supports several user information registries for authentication purposes. Local registries are limited to a single application server. Centralized registries use the LTPA protocol to access a supported LDAP service. Customer-defined registries or pluggable registries use the WebSphere Custom Registry interface that facilitates access to a custom user registry. After enabling security in the WebSphere Security Center, WebSphere uses the local registry (operating system) by default to authenticate users. Although LDAP is becoming one of the major user repositories, there are still many companies that store user information in a database. I am not going to enter the LDAP versus database debate here. I will confine myself to stating that both have their strengths and weaknesses in particular uses.

LTPA

If neither the OS or the LDAP authentication is applicable for the target platform, WebSphere provides a third, more generic authentication mechanism, called Lightweight Third Party Authentication (LTPA) or do-it-yourself authentication. Another great excuse for developers to take legal advantage of the Not Invented Here syndrome, LTPA offers the possibility to use a nonstandard or a legacy solution that is not natively supported by WebSphere as a custom user registry for authentication purposes. A database, for instance, is not a "standard" solution, because there is no industry-wide agreement on the exact data schema, when used in case of authentication.

LTPA circumvents this problem by providing an interface that can be implemented. The CustomRegistry interface of WebSphere holds all the methods that the application server uses for its authentication. Building your own WebSphere custom user registry on top of any technology is easy. Just give an implementation for these methods and WebSphere LTPA can do the rest. Even the more advanced authentication features of WebSphere, such as single sign-on, (SSO) are supported by LTPA. Because the LTPA interface is WebSphere specific, changing the J2EE server platform means finding another solution for the LTPA custom user registry.

A Custom User Registry

The sample custom user registry in this article uses a database as the user repository and runs on any DBMS that supports JDBC. The sample code is independent of a specific database, with all its harmful consequences, such as the passwords being in clear text due to the lack of a universal encryption function. It is therefore advised not to use this sample implementation in a real production environment. The selection of the database is configured in the WebSphere Security Center. I have used both MySQL and Microsoft SQL Server to test the sample code. Before going right into the gory details of LTPA, first there are some prerequisites that you will need when building your own custom user registry. Besides the sample code for this article, which you can download from www.sys-con.com/websphere/sourcec.cfm, you also need the JDBC drivers for the DBMS you use. To make life easier the authentication form and error page are taken from the "Big3 application" downloads.zip file from the IBM site (www-3.ibm.com/software/webservers/appserv/doc/v40/aes/infocenter/was/0601_downloads.html). This zip file contains a login.jar file, which you will need later on. IBM also provides a sample custom user registry in the WebSphere InfoCenter, based on text files.

The User Database

The user database in this sample has a very simple layout. Its only purpose is to hold the information needed by LTPA. A "real" solution could provide a much more complicated data model, but for this sample I will stick to the data model in Figure 1. It contains a user table, a group table, and a member table to store the relations between the users and the groups. In many cases the database layout is predetermined by an existing system rather than a variable. The difficulty in these situations is to locate the corresponding fields in the database. Once the database is up and running, you can start with the actual custom user registry coding.

It is not possible to use the "normal" JDBC DataSource for enterprise applications to access the database from your custom user registry. The reason is that the WebSphere Application Server provides the JDBC DataSource when it is running and the custom user registry should also be available in the WebSphere Administration Server, even when no application server is active. For instance the Administrator's Console is protected by LTPA (if selected in the Security Center). So the old-fashioned JDBC Driver is used in the custom user registry to obtain the connection to the database.

CustomRegistry Interface

The custom user registry itself is just one Java class that implements the CustomRegistry interface. I will not discuss the complete sample source code of the custom user registry. It is well documented and not very complex. I will just highlight particular issues.

The CustomRegistry interface consists roughly of four groups of methods, i.e., two general methods (initialize and getRealm), two authentication methods (checkPassword and mapCertificate), eight user-related methods, and eight group-related methods.

The initialize method takes a properties argument to initialize the registry. The administrator sets these properties in the Security Center (Special Custom Settings, see Figure 2). In the sample registry the properties are used to provide the JDBC driver (name), the connection URL, and the username and password to connect to the database. The method getRealm returns the realm of the registry. A realm determines the scope of security data. It is the region to which a security ID or permission applies. The realm is shown in the login screen, when you start the Administrator's Console. All other methods include pretty straightforward SQL statements to provide the functionality.

During deployment you can search through users and groups with the help of regular expressions. I have added an extra method (regular ExpressionTo SQL) in the Custom Registry implementation class, which provides a very simple SQL wildcards translation.

The most important concern is the correctness of your code. It is advisable to write a small JUnit test program to test your code. Correctness is crucial because you can lock yourself out of the Administrator's Console. If your custom user registry contains a bug that prevents you from gaining access, you will not be able to start up the Administrator's Console, without having to go into the WebSphere configuration files.

Build the Registry

In order to compile and build the custom user registry, you need to add the websphere.jar on the Java build path. In WebSphere Application Developer, add the JAR file via the variable WAS_PLUGINDIR/lib/websphere.jar. If you use some other IDE, you can find this JAR in the WebSphere directory (WebSphere/AppServer/lib/websphere.jar).

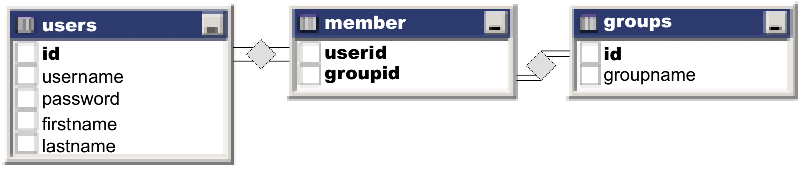
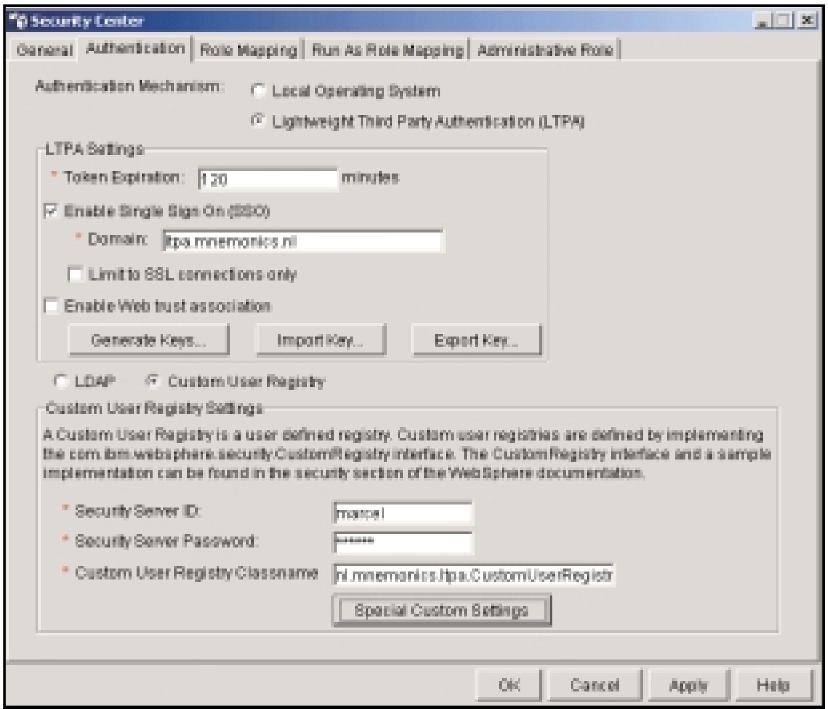


FIG. 1: DATA MODEL FOR A SIMPLE CUSTOM USER REGISTRY



Authentication Mechanism: ○ Local Operating System ● Lightweight Third Party Authentication (LTPA)

LTPA Settings

* Token Expiration: 120 minutes

☑ Enable Single Sign On (SSO)

* Domain: ltpa.mnemonics.nl

□ Limit to SSL connections only

□ Enable Web trust association

○ LDAP ● Custom User Registry

Custom User Registry Settings

A Custom User Registry is a user defined registry. Custom user registries are defined by implementing the com.ibm.websphere.security.CustomRegistry interface. The CustomRegistry interface and a sample implementation can be found in the security section of the WebSphere documentation.

* Security Server ID: marcel

* Security Server Password: ******

* Custom User Registry Classname: nl.mnemonics.ltpa.CustomUserRegistr

FIG. 2: SELECTING LTPA FOR AUTHENTICATION IN THE WEBSPHERE ADMINISTRATION CONSOLE SECURITY CENTER

The class file compiled during the build must be copied to the WebSphere/Appserver/classes directory. Don't forget the directory structure of the Java package, or else the Security Center will not be able to use the registry. In the case of the sample registry the full path is WebSphere/Appserver/classes/nl/mnemonics/ltpa/ CustomUserRegistry.class.

Note that it is important to be very careful when upgrading to a newer version of your custom user registry. Use a different class name for the new registry, or select OS authentication first, reboot the administration server, overwrite the registry class, and select the new class in the Security Center. Overwriting a registry class while the Administration Server is running will result in a lockout. Remember always to be careful when altering security settings, especially the implementation class.

Library JARs

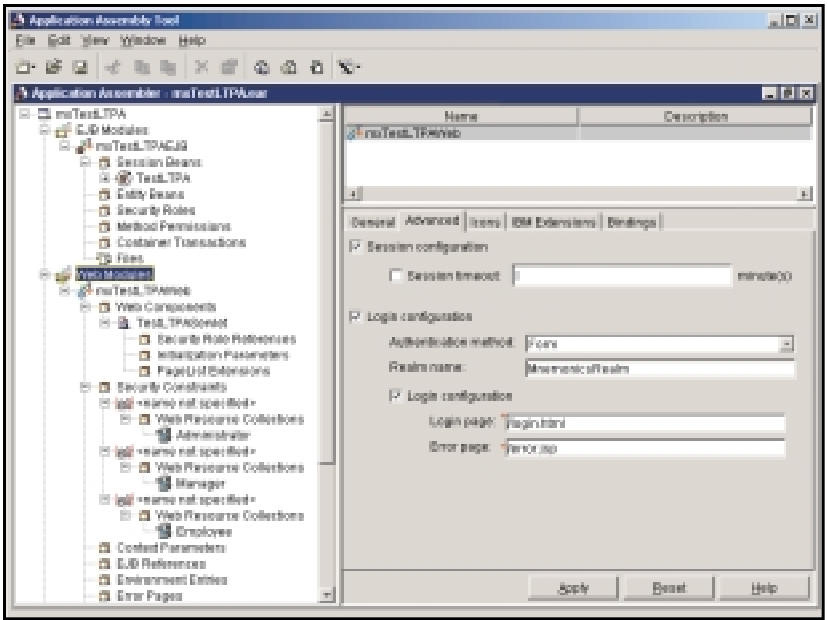
There are many opinions as to where to put library JARs. Because there are several classpaths (for the specific class loaders in WebSphere), there are also several places to store libraries. A rule of thumb is to put JARs as close to your application as possible, to minimize classpath problems. In other words, put it in the EAR file. In this case, however, there is no EAR file, at least not one specific for LTPA. The next best thing is to put the library JARs in the WebSphere Library Extension directory (WebSphere/Appserver/lib/ext).The JDBC driver library JARs for the selected DBMS are stored at this location.

Security Center

When the registry code is tested and installed in the classes directory of WebSphere and the JDBC Driver library files are available in the library extension, the LTPA custom user registry can be configured in WebSphere. Open the Security Center in the Administrator's Console, enable security, and go to the Authentication tab (see Figure 2). Select LTPA and enter the full domain name of the WebSphere Application Server. It is very important to enter the full name and not just the server name or else the registry will not work. It took me over an hour to figure that out.

Select Custom User Registry and enter the registry settings (user name, password, and class name). The user name and password of the registry settings represent the administrator's account. The next time, you need to use these credentials to log in to the Administrator's Console.

Click the Special Custom Settings button and enter the name/value pair settings for the initialize method of the registry (driver, URL, user, and password). The user and password of the Special Custom Settings belong to the corresponding JDBC connection (URL).



☑ Session configuration

☐ Session timeout: minute(s)

☑ Login configuration

Authentication method: Form

Realm name: MnemonicsRealm

☑ Login configuration

Login page: /login.html

Error page: /error.jsp

FIG. 3: CONFIGURING THE LOGIN, WEB RESOURCE SECURITY CONSTRAINTS, AND EJB METHOD PERMISSIONS

If everything is okay, applying the settings will cause the system to respond with something about the settings becoming active after the Administration Server is restarted. The Security Center checks the validity of the Security Server ID and the Security Server password during this process. If there is a bug in your custom user registry code, the settings are wrong, or WebSphere is unable to connect to the database, you will get an error. You can pinpoint the source of the problem with the provided stack trace.

When you restart the Administration Server the custom user registry will be active. Notice the custom user registry realm name during startup of the Administrator's Console.

WebSphere Versions

Although I have not experienced problems with different versions of WebSphere, there are reports of failing LTPA security in some versions of WebSphere Application Server. I have tested the custom user registry in WebSphere 4.0.3 and 4.0.4, and both worked fine.

Application Assembly

The sample application, mxTestLTPA, is not really an enterprise application at all. It is just a single dummy EJB with some simple Web resources for testing the custom user registry. The EJB has three methods (employee, manager, and administrator) that coincide with the J2EE roles defined in the EAR file and the groups defined in the database. I have added method permissions with the Application Assembly Tool (AAT) to map the groups directly on the J2EE roles. Of course, it is not mandatory that J2EE roles have names identical to those of the groups in a user registry.

There are also three HTML pages (employee.html, manager.html, and administrator.html) that are protected by the security constraints of the Web resource. I have added the login.html and error.jsp from the IBM "Big 3" application (login.jar). Notice the name of the HTML form and fields in login.html. This naming scheme enables WebSphere to use the username and password information for authentication purposes.

The Login Configurations can be found in the Web Modules node of the AAT tree in the Advanced tab (see Figure 3).

The J2EE roles, method permissions of the EJB, security constraints of the Web resources, and login configuration can also be set by WSAD. In many cases you can skip the AAT part completely when you use WSAD.

Authorization

Once the security settings of the EAR file are completed, the application can be deployed on the application server. During deployment the Administrator's Console will ask you to map the J2EE roles defined in the EAR file onto the users and groups of the LTPA custom user registry (see Figure 4). Now you can see some of the methods of your registry in action. For instance a search with simple regular expressions will fill the available users and groups list (see Figure 5). After the role mapping and the rest of the deployment, you can use the application to test the custom user registry.

Conclusion

Using the WebSphere feature to allow custom user registries via LTPA is an option and not the standard solution. Personally, I avoid using the custom user registry and use LDAP if there is a choice. The custom user registry is provided as a last resort, in case the project, the infrastructure, or the system environment dictates the type of authentication registry.

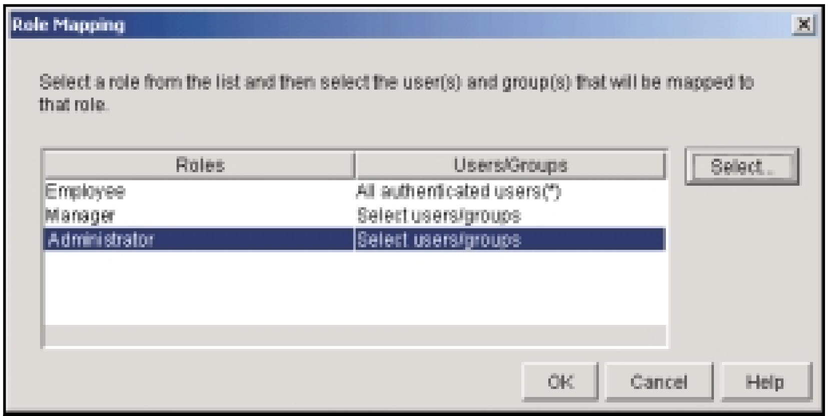
As described earlier, building an LTPA custom user registry in WebSphere is very easy. You have to be careful, though, to produce correct code or you might lock yourself out of the Administrator's Console. Always test a custom user registry and never overwrite the custom user registry class while it is active in WebSphere.

There is one characteristic of user registries in WebSphere that might be confusing, especially in the case of a custom user registry. For performance reasons, the authentication of users and groups is cached in WebSphere. It is understandable that WebSphere doesn't check the users and groups for every call to a protected EJB method or Web resource. The performance would drop to an unacceptable level in the case of the database custom user registry described here. It will only check the validity of a user the first time and after expiration of the timeout.

The reason to have a custom user registry in the first place is because of an existing system that stores the user information. Inconsistencies may occur when other (non-WebSphere) applications are also manipulating the user information repository (e.g., database). If another application changes the membership of users and groups, the altered relationship will not be reflected in WebSphere immediately. Dropping the caching timeout is a possible remedy, but it will hurt the performance significantly.

References

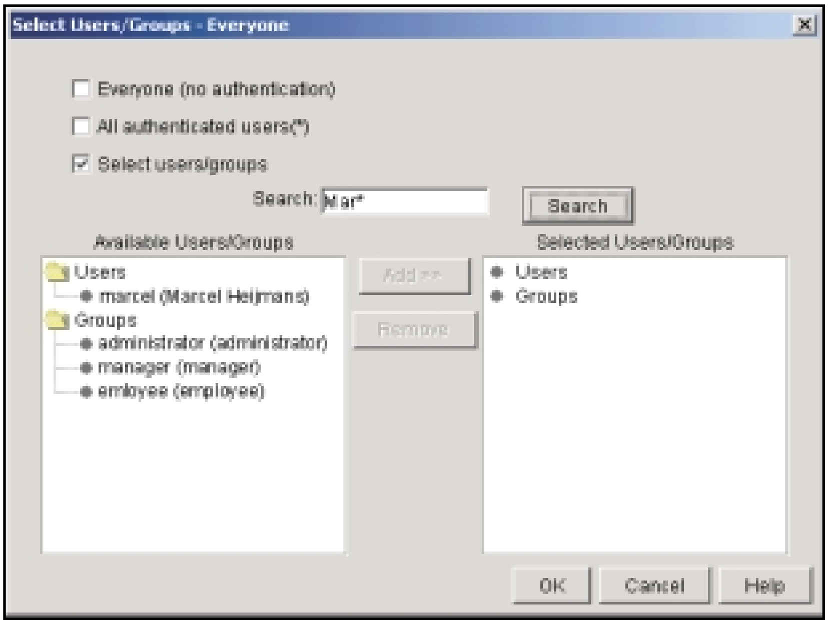
- *IBM WebSphere InfoCenter, 5.2: Introduction to custom registries:* www-3.ibm.com/software/webservers/appserv/doc/v40/aee/wasa_content/0502.html
- *IBM WebSphere V4.0 Advanced Edition Security,* IBM Redbooks.
- Ben-Natan, R., and Sasson, O. (2002). *IBM WebSphere Application Server: The Complete Reference,* McGraw-Hill Osborne Media.



Select a role from the list and then select the user(s) and group(s) that will be mapped to that role.

Roles	Users/Groups
Employee	All authenticated users(*)
Manager	Select users/groups
Administrator	Select users/groups

FIG. 4: MAPPING J2EE ROLES ONTO THE USERS AND GROUPS FROM THE CUSTOM USER REGISTRY



☐ Everyone (no authentication)

☐ All authenticated users(*)

☑ Select users/groups

Search: Mar*

Available Users/Groups

- Users
  - marcel (Marcel Heijmans)
- Groups
  - administrator (administrator)
  - manager (manager)
  - emloyee (employee)

Selected Users/Groups

- Users
- Groups

FIG. 5: DURING DEPLOYMENT, WEBSPHERE CAN FIND THE USERS AND GROUPS VIA THE IMPLEMENTATION OF THE LTPA INTERFACE

The Dynamic Caching Services

Eliminate bottlenecks and improve response time

– BY GENNARO CUOMO AND CATHERINE DIEP –

Server-side caching techniques have long been used to improve Internet performance of Web applications. In general, caching improves response time and reduces system load. Most techniques cache static content, which is data that changes rarely, if at all, such as graphics and text files. While the solutions for caching static content have resulted in excellent performance for some Web applications, they have little or no value in enhancing the performance of Web applications with dynamically generated pages.

ABOUT THE AUTHOR

Gennaro (Jerry) Cuomo is a distinguished engineer at IBM. He has been a core member of the WebSphere development and architecture team since its inception in 1998. Jerry is CTO of the WebSphere Technology Institute, which is responsible for pursuing the next generation of WebSphere technologies.

E-MAIL

gcuomo@us.ibm.com

For years, IBM Research has developed and refined technologies that enable the caching of dynamic content. These technologies were implemented, deployed, and verified at various high-volume sporting event sites such as the 1998 Winter Olympic Games in Nagano. The success of the sports sites demonstrated the feasibility and significance of caching dynamic content and confirmed the scalability and reliability of the caching technologies. Based on these proven and scalable caching technologies, IBM developed a dynamic content caching solution for Java 2 Enterprise Edition (J2EE) applications running on WebSphere Application Server Version 5.0. In this article we introduce WebSphere dynamic caching services.

WebSphere Application Server (WAS) offers a built-in dynamic caching service for serving dynamic content and caching data. The WebSphere development team often refers to this service simply as dynacache. There is no time-consuming installation and integration work needed to activate dynacache. The cache is enabled/disabled declaratively using simple XML configuration files or using the WAS Administrative User Interface; these methods not only allow caching to be brought up quickly and easily, but also provide great flexibility and control at runtime. Also, you can define your existing caches, such as the caching component of WebSphere Edge Server or IBM HTTP Server, as external caches and use them in conjunction with dynacache.

This article introduces the presentation-level caching features of dynacache. These features are:

- ***Servlet/JSP result cache:*** Nonintrusive, effortless, and ready to cache any existing whole page or fragment generated by a servlet or JSP.
- ***Command cache:*** Used to cache dynamic data before it is transformed into a presentable format (i.e., HTML).
- ***Replication support:*** Enables cache sharing and replication among multiple servers and tiers.
- ***Invalidation support:*** Includes rules-based, time-based, group-based, and programmatic cache invalidation techniques to ensure the freshness, consistency, and accuracy of the content.
- ***Edge of network caching support:*** Extends the WAS caches into network-based caches, through the implementation of external cache control and distributed-fragment caching and assembly support.
- ***Tools:*** Assist in configuring the cache and monitoring runtime.

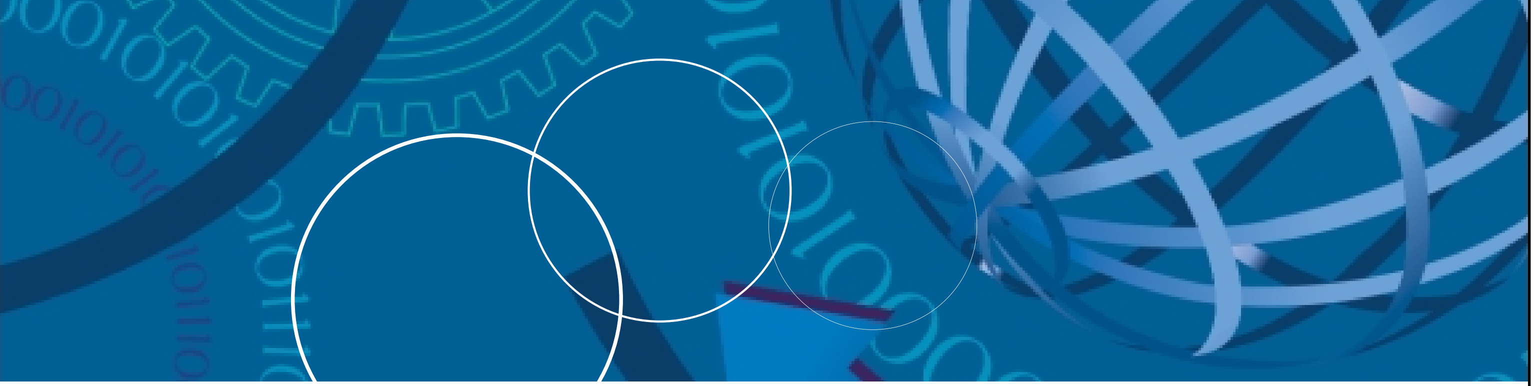
Caching Dynamic Content

The key issues for caching dynamic content involve determining what should be cached, where caching should take place, and how to invalidate cached data.

WHAT SHOULD BE CACHED?

Content or data that is changing and that at the same time must be stable over a long enough time for meaningful reuse to occur is a candidate for dynamic content caching . If access is frequent, such as with pricing information of a popular stock, then even a short period of stability may be of enough benefit to warrant the caching of dynamic content.

All dynamic Web pages consist of smaller and simpler page fragments. Some fragments are static (such as headers and footers), while others are dynamic (such as fragments containing stock quotes or sports scores). Breaking a page into fragments or components makes effective caching possible for any page, even a highly dynamic page. The goal of creating fragments or components is to maximize fragment reusability and cache utilization.



For example, Figure 1 shows that a page can be broken down into fragments based on reusability and cacheability. Some fragments – headers, footers, and navigation bars, for example – are reusable by all users who visit this site. Other fragments, such as watch lists and news, are not typically cacheable at the presentation level because of the personalized content being displayed. While some presentation fragments may not be reusable across a large set of users, the raw data contained within may very well be cacheable.

Caching can involve a final formatted whole page (such as HTML or XML), a final formatted fragment, or a piece of unformatted raw data. Each, in its own way, contributes to the ultimate benefit of caching dynamic content. Dynacache offers features that enable dynamic content to be cached at various granularities, namely whole pages, fragments, and raw data.

WHERE SHOULD CACHING TAKE PLACE?

Theoretically, caching of dynamic content should take place as close to the user as possible. In reality, other factors such as security and user-specific data may influence the choice for the best place to cache dynamic content.

Web application design can play an important role in determining where dynamic data is cached. One example is personalized pages. Although personalized, these pages may contain user-specific, nonuser-specific, locale-sensitive, secure, or nonsecurity-sensitive dynamic data. To maximize the benefit of caching dynamic content, these types of data should be fragmented as finely as possible so they can be cached independently at different locations. For example, the nonuser-specific, nonsecurity-sensitive fragments or components are generally useful to many users, and thus can be cached in a more public space and closer to users. The security-sensitive data should be cached behind the enterprise firewall.

In a multitier e-business environment, dynacache can be activated at the business logic and/or presentation layer. It can also control external caches on servers, such as IBM WebSphere Edge Server and IBM HTTP Server. When external caching is enabled, the cache matches pages with their universal resource identifiers (URIs) and exports matching pages to the external cache. The contents can then be served from the external cache instead of the application server, which saves resources and improves performance. Additionally, dynacache's replication and invalidation support extends the cost-effectiveness of caching dynamic content by enabling cache sharing and cache replication in an environment containing multiple tiers and servers. Finally, WAS's edge of network caching support expands the application caches into the network.

HOW ARE CACHES INVALIDATED?

The biggest challenge when caching dynamic content is to guarantee the freshness, consistency, and accuracy of the content. This requires efficient and comprehensive mechanisms for identifying and updating pages/fragments/data that are no longer valid, a process called invalidation.

Dynacache provides invalidation techniques that are rules-based, time-based, group-based, and programmatic. It can also invalidate the remote caches that were configured as its external caches. Dynacache uses a facility called the Data Replication Service, which is a JMS-based facility, to replicate cached data and propagate invalidate events within a WebSphere cluster.

Dynamic Cache Service Architecture

Figure 2 illustrates the architecture of dynacache. Dynacache operates within WebSphere's JVM to provide generalized Java object caching for use by various consumers. In order to minimize memory consumption, the caching service utilizes a modified LRU (least recently used) eviction policy for cached entries. As a service, dynacache can be configured, tuned, and monitored through WebSphere's systems management interfaces.

In its simplest form, dynacache can be thought of as a highly functional Hashtable. One of its primary purposes is to store (put) and retrieve (get) Java objects from memory. As the various caching entry-points interact with the object store, dynacache manages the entries by controlling growth by running replacement or eviction algorithms. Entries in the cache are managed by a policy based on the class of entry (e.g., servlet/JSP, EJB, command, etc.). The policy is expressed as an XML file called cachespec.xml. The policy defines both global and class-specific characteristics of the cache. For example, cache size, replication policy, and disk offload configurations are globally defined for an instance of dynacache. However, much of the policy data is class specific, describing the rules by which Cache IDs are generated. Cache policies can be written to control servlet/JSP, command bean, and Web services response caching.

Dynacache provides a disk overflow capability, known as Hash Table on Disk (HTOD), that is an optimized file store for serialized Java objects. The HTOD takes the form of a single logical disk file (which can expand to multiple physical 1GB files). Leveraging dynacache and HTOD allows for robust and centralized memory management and provides the potential to leverage networked file systems via NAS (network attached storage) and SANs (storage area networks) for high availability.

To support caching done outside of the J2EE application space, the caching service can cooperate with external caches through its external cache adapter to actively trigger invalidations of cached Web content to registered adapters, which include the WebSphere HTTP Plugin, IBM

ABOUT THE AUTHOR

Catherine Diep is a senior software engineer at IBM, where she is a member of the High-Volume Web Sites team, which supports proofs-of-concept, scalability design and testing, and presales opportunities. Her team has won for IBM the business of many large customers' e-business infrastructures.

E-MAIL
cdiep@us.ibm.com

HTTP Server, and its FRCA (Fast Response Cache Accelerator) cache, and the WebSphere Edge Server now shipped as part of WebSphere Application Server Network Deployment.

Servlet/JSP Result Cache

The servlet/JSP result cache intercepts calls to a servlet's service method, and checks whether the invocation can be served from a cache. If the servlet cannot be served from cache, it is invoked to generate the output that will be cached. The resulting cache entry contains the output; the side effects of the invocation, for example, calls to other servlets or JSP files; and the metadata about the entry, including timeout and entry priority information

Servlet/JSP result cache caching can be based on:

- Request parameters and attributes
- The URI used to invoke the servlet or JSP
- Session information
- Other options, including cookies

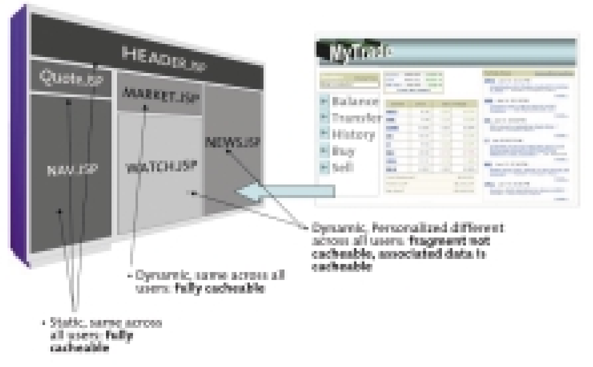

HEADER.JSP
Quote.JSP
MARKET.JSP
NEWS.JSP
NAV.JSP
WATCH.JSP
MyTrade
Balance
Transfer
History
Buy
Sell
• Dynamic, Personalized different across all users: fragment not cacheable, associated data is cacheable
• Dynamic, same across all users: fully cacheable
• Static, same across all users: fully cacheable

FIG. 1: EXAMPLE OF FRAGMENT CACHING

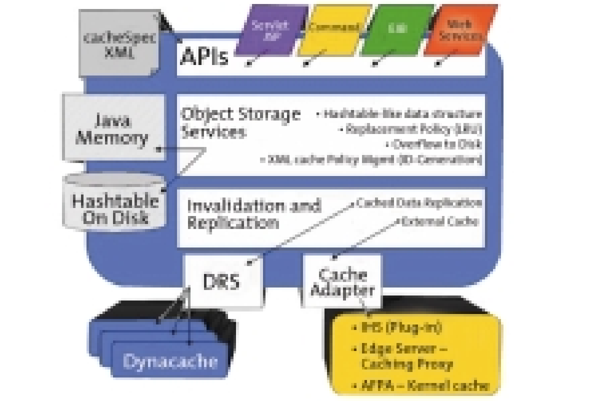
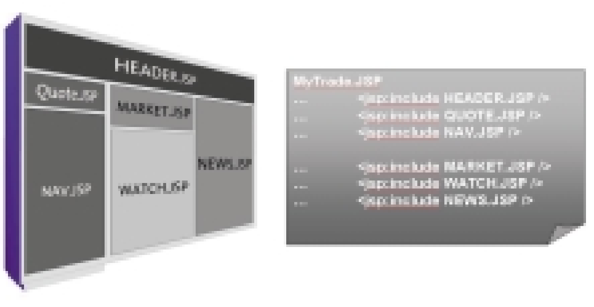


FIG. 2: WEBSPHERE DYNACACHE ARCHITECTURE



```
MyTrade.JSP
...          <jsp:include HEADER.JSP />
...          <jsp:include QUOTE.JSP />
...          <jsp:include NAV.JSP />

...          <jsp:include MARKET.JSP />
...          <jsp:include WATCH.JSP />
...          <jsp:include NEWS.JSP />
```

FIG. 3: CACHING FRAGMENTS USING THE SERVLET/JSP RESULT CACHE

Figure 3 shows a sample dynamic Web page, MyTrade.JSP, which is composed of six included JSPs. Using servlet/JSP result caching, either the whole page generated by MyTrade.JSP or individual included JSP fragments can be independently cached within dynacache.

Alternatively, the cache entries generated by the servlet/JSP result cache can be pushed to external caches, such as the cache component of WebSphere Edge Server or IBM HTTP Server. For J2EE applications that have high read/write ratios, the servlet/JSP result cache creates an opportunity for significant gains in server response time, throughput, and scalability. Moreover, since the servlet/JSP result cache is nonintrusive and is enabled declaratively, currently deployed and running servlets/JSPs can be configured to take advantage of dynamic content caching without changing any code.

The Command Cache

The command cache is used to cache dynamic data that requires back-end data retrieval, such as back-end database JDBC calls. The command cache forms a good synergy with the servlet/JSP result cache because in some cases even caching the most granular, final-formatted fragment is not sufficient. For example, a personalized page may contain a stock watch-list fragment (see Figure 4). This fragment consists of two sets of information: the watch list, which is highly personalized, and the corresponding stock symbol pricing information, which is generalized information and usable by many users.

Suppose the stock list is the customer's stock portfolio, which is highly sensitive and is stored at the back-end server. In this case it is not effective to cache the final formatted fragment. Since the stock quotes are highly volatile, this fragment will be regenerated repeatedly. The net result is that every time this fragment is reconstructed, it is necessary to retrieve the stock list from the source. A better approach is to use the command cache to cache the stock list and avoid fetching the list continually from the back-end database.

To use the command cache, user applications need to write to the WebSphere Command Framework API. The WebSphere Command Framework is based on the Command Design Pattern widely used in Java programming paradigms. Typically, these applications use "setters" to set the command's input states, one or more "execute" methods, and "getters" to retrieve the results. The results can be either formatted or raw data.

The command cache intercepts the "execute" method call of a command written for the command cache and checks whether the command object can be served from the cache. If the command object does not exist in the cache, the logic in the execute method is performed and the resulting object is cached.

The caching behavior of the command cache is defined declaratively with the XML cache policy file, which describes whether and how a command should be cached. The command cache can be used at the presentation and/or business logic layer in multitier environments.

The command cache is easy to use. For existing applications that follow a Model-View-Control (MVC) design pattern, the command cache can be implemented with minimal impact to existing code.

Replication Support

Replication support further extends the value of caching in an e-business application. With this support, caches can be shared (central cache) or replicated (local cache) among servers. Replication can be enabled and configured declaratively with XML cache policy files. Cache policy also defines the cache entries or groups that should be shared or replicated.

Replication support uses a built-in high-performance standards-based JMS messaging system as its underlying engine for data replication.

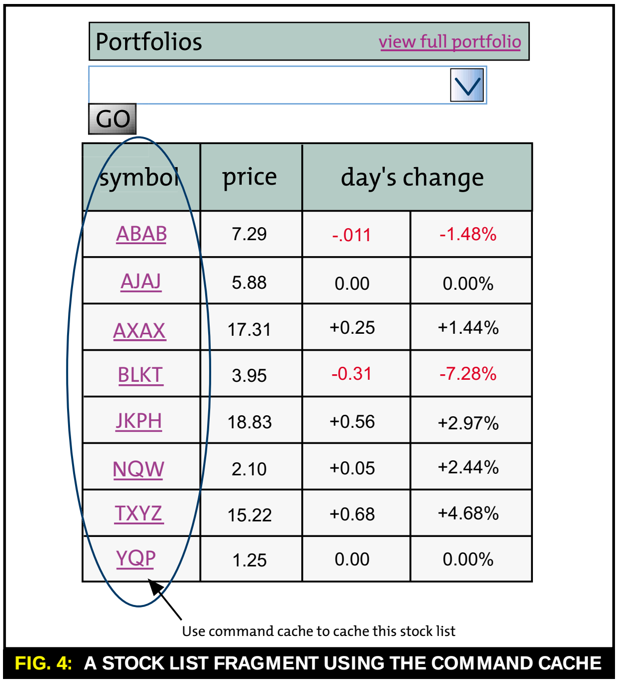
Figure 5 shows an example of three servers configured to have their local caches replicated with each other. The messaging broker can be the built-in JMS broker belonging to one of the three servers. In this case, the broker is the built-in JMS broker at server one. A request for data (1) hits server one (or any of the three servers). If the requested data is not in the cache, the data is fetched from the back-end server (2). Resulting data from the back-end server (3) is cached at server one and then returned to the requester (4). When the cache of server one detects the cache update request, it publishes a message (a) regarding the change to the messaging broker. Whatever change occurred at server one is automatically replicated to server two (b) and server three (c).

Invalidation Support

The difference between caching static and dynamic content is the requirement for proactive and effective invalidation mechanisms, such as event-based invalidation, to ensure the freshness of content. Time-based invalidation alone is no longer adequate for dynamic cache invalidation.

Dynacache provides rules-based, time-based, and group-based invalidation techniques. The WebSphere Application Server Enterprise Edition, version 5, offers access to programmatic cache and invalidation techniques. Invalidation policies can be defined with XML cache policy files. Invalidation policies allow triggering events to invalidate cache entries without the need to write explicit code. More complex invalidation scenarios may require code that invokes the invalidation API.

The responsibility for synchronizing the dynamic cache of external caches with the application server is shared by both systems. For example, a public Web page dynamically created and cached at the application server using the servlet/JSP result cache can be exported by the application server and cached by the edge server's Caching Proxy. The Caching Proxy can then serve the application's execution results repeatedly to many different users until notified that the page is invalid. Content in the Caching Proxy's servlet response cache is valid until the proxy server removes an entry because the cache is congested, the default timeout set by the Caching Proxy's configuration file expires, or the Caching Proxy receives an invalidate message directing it to purge the content from its cache. Invalidate messages originate at the application server that owns the content and are propagated to each configured Caching Proxy.



symbol	price	day's change	
ABAB	7.29	-.011	-1.48%
AJAJ	5.88	0.00	0.00%
AXAX	17.31	+0.25	+1.44%
BLKT	3.95	-0.31	-7.28%
JKPH	18.83	+0.56	+2.97%
NQW	2.10	+0.05	+2.44%
TXYZ	15.22	+0.68	+4.68%
YQP	1.25	0.00	0.00%

Use command cache to cache this stock list

FIG. 4: A STOCK LIST FRAGMENT USING THE COMMAND CACHE

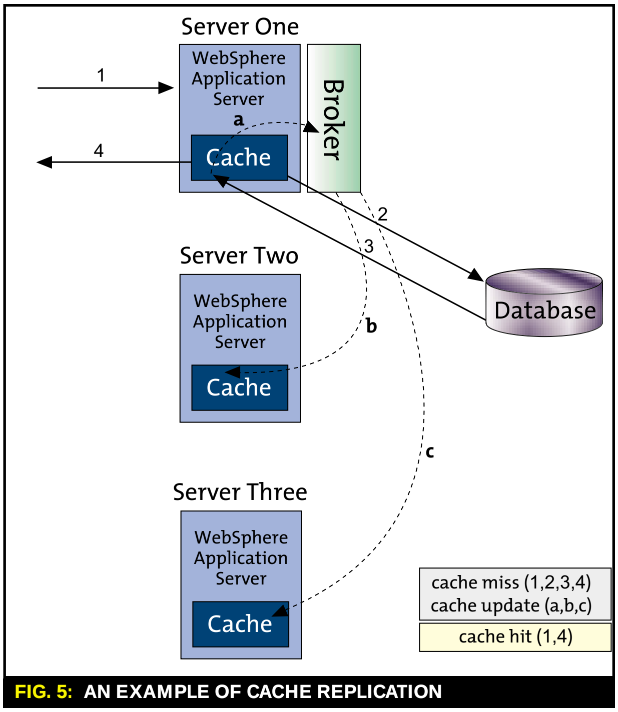


FIG. 5: AN EXAMPLE OF CACHE REPLICATION

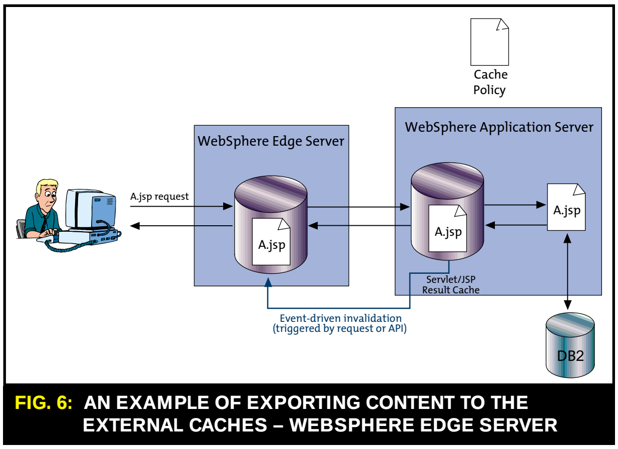


FIG. 6: AN EXAMPLE OF EXPORTING CONTENT TO THE EXTERNAL CACHES – WEBSPHERE EDGE SERVER

Edge of Network Caching Support

EXTERNAL CACHE

Dynacache can use IBM HTTP Server's high-speed cache, referred to as the Fast Response Cache Accelerator, as its external cache, to cache whole pages and fragments.

The Edge Components of WebSphere Application Server Network Deployment v5.0 can also be configured as the application server's external cache for whole-page caching. In such cases, dynacache can be enabled to match pages with their universal resource identifiers (URIs) and export matching pages to the external cache. The contents can then be served from the external cache instead of the application server to significantly save resources and improve performance.

Figure 6 shows an example of exporting a dynamically cached page from the application server to the cache component of the edge server. More information about caching dynamic content at the edge of the network can be found in the white paper, "WebSphere Edge Services Architecture Guide to Edge Applications" (www-3.ibm.com/software/webservers/edgeserver/doc/ Guide_ to_ Edge_Apps_2.pdf).

With dynacache's external cache control, distributed-fragment caching, and assembly support, dynamic content can be exported, cached, and assembled at the most optimal location, closer to the end user. More important, the WebSphere Application Server can maintain control of the external cache through its invalidation support to ensure the freshness of cached content. As a result, WebSphere Application Server is equipped to create and serve highly dynamic Web pages without jeopardizing page performance and user experiences.

In Table 1, you can see which caching technique and cache distribution technique are most useful for each type of Web content.

Tools

WebSphere Application Server provides a variety of graphical user interface tools to help configure and monitor the dynamic content cache.

APPLICATION ASSEMBLY TOOL

The Application Assembly Tool (AAT) is used to package application code components into the needed modules, which comply with the J2EE specification for deployment onto the application server. AAT can also be used to configure the cache policy associated with the application for caching.

CACHE MONITOR SERVLET

The Cache Monitor Servlet is provided as an installable Enterprise Archive file that can be installed and used for monitoring cache statistics.

Conclusion

As more e-business sites seek to retain customers by serving personalized content, they face potential server-side bottlenecks, slow response time, and increasing infrastructure costs. The dynacache service employed by WebSphere can solve these critical challenges. Caching dynamic content that requires back-end requests or CPU-intensive computations can reduce server-side bottlenecks and maximize system resources, thus boosting performance and reducing infrastructure cost.

Dynacache users can benefit from using the available comprehensive functions to cache their dynamic content. The servlet/JSP result cache and the command cache make possible the caching of dynamic content at various levels of granularity for the highest possible number of cache hits. The replication and invalidation support facilitates caches to be shared, replicated, and synchronized in multitier or multiserver environments. The edge of network caching support, with its external caches and fragment support, generates a virtual extension of application server caches into the network.

Using dynacache can improve throughput and performance. As expected, improvements vary depending on how dynamic the data is, how well the pages are designed (fragmented) for maximum cache hits, and how costly it is to fetch and construct the content. Nevertheless, considerable benefits can be realized. Furthermore, the success of the IBM sports sites confirms the scalability and reliability of these underlying technologies in a truly high-volume environment.

TYPE OF WEB CONTENT	CACHING TECHNIQUE	CACHE DISTRIBUTION TECHNIQUE
Published/static (*.html, *.gif, ...)	WebSphere Edge Server Caching Proxy, FRCA	
Dynamic fragment/page (dynamically generated and reusable)	WebSphere Application Server Servlet/JSP Result Cache WebSphere Edge Server cache component and/or IBM HTTP Server as external cache WebSphere Application Server's HTTP plug-in distributed-fragment caching and assembly support	WebSphere Application Servlet/JSP Result Cache configured with WebSphere Edge Server and/or IBM HTTP Server as external cache WebSphere Application Server's Cache Replication Support
Dynamic data (data object)	WebSphere Application Server Command Cache	WebSphere Application Server Cache Replication Support

TABLE 1: CACHING TECHNIQUES FOR WEB CONTENT

Premiering
June 2003
at
JavaOne
Sun's 2003 Worldwide Java Developer Conference
www.sys-con.com
SERVER FOCUS: RUNNING ENTERPRISE LINUX
LINUX
BUSINESS & TECHNOLOGY
SPECIAL PREMIER ISSUE!
Linux Takes a Road Trip
A guide to laptop Linux
FREE CD!
Migrating Your Existing Servers to Linux
Maintaining and Running Oracle on Linux
What Skills Should Linux Experts Arm Themselves With?
Look at Dell's Latest Offerings
Millions
of Linux Users
One Magazine

Linux Business and Technology

There is no escaping the penetration of Linux into the corporate world. Traditional models are being turned on their head as the open-for-everyone Linux bandwagon rolls forward.

Linux is an operating system that is traditionally held in the highest esteem by the hardcore or geek developers of the world. With its roots firmly seeded in the open-source model, Linux is very much born from the "if it's broke, then fix it yourself" attitude.

Major corporations including IBM, Oracle, Sun, and Dell have all committed significant resources and money to ensure their strategy for the future involves Linux. Linux has arrived at the boardroom.

Yet until now, no title has existed that explicitly addresses this new hunger for information from the corporate arena. *Linux Business & Technology* is aimed squarely at providing this group with the knowledge and background that will allow them to make decisions to utilize the Linux operating system.

Look for all the strategic information required to better inform the community on how powerful an alternative Linux can be. *Linux Business & Technology* will not feature low-level code snippets but will focus instead on the higher logistical level, providing advice on hardware, to software, through to the recruiting of trained personnel required to successfully deploy a Linux-based solution. Each month will see a different focus, allowing a detailed analysis of all the components that make up the greater Linux landscape.

Regular features will include:

Advice on Linux Infrastructure
Detailed Software Reviews
Migration Advice
Hardware Advice
CEO Guest Editorials
Recruiting/Certification Advice
Latest News That Matters
Case Studies

SAVE 30% OFF!

REGULAR ANNUAL COVER PRICE $71.76

YOU PAY ONLY

$$49^{99}$

12 ISSUES/YR

**OFFER SUBJECT TO CHANGE WITHOUT NOTICE*

SUBSCRIBE TODAY!

WWW.SYS-CON.COM
OR CALL
1-888-303-5282

LINUXEDGE conference&expo

June 3–5	LONDON
June 24–26	BERLIN
September	HONG KONG
October	CALIFORNIA

FOR ADVERTISING INFORMATION:
CALL 201 802.3020 OR
VISIT WWW.SYS-CON.COM

SYS-CON MEDIA
The World's Leading i-Technology Publisher

ALL BRAND AND PRODUCT NAMES USED ON THIS PAGE ARE TRADE NAMES, SERVICE MARKS, OR TRADEMARKS OF THEIR RESPECTIVE COMPANIES.

Part 1: The business problem

Discovering and Documenting Business Application Patterns

BY BRENT CARLSON
AND JAMES CAREY

You're probably saying to yourself, "Oh, no! Not another patterns article!" Technically, that's what this is. However, instead of simply showing you a finished pattern, we're going to look at pattern discovery. And, while what we'll talk about can help you capture fundamental patterns (that is, if there are any fundamental patterns left to be defined) we're going to focus on the capture of more workaday patterns. These are patterns that may be useful to only you and your team and might never be made into formal patterns or even shared outside of your group. In fact a pattern purist might say they aren't even patterns.

ABOUT THE AUTHOR

Brent Carlson is vice president of technology and cofounder of LogicLibrary, Inc., a leading provider of enterprise software and services for enabling existing software development assets as Web services. With James Carey, Brent has coauthored SanFrancisco *Design Patterns: Blueprints for Business Software,* and *Framework Process Patterns: Lessons Learned Developing Application Frameworks.*

E-MAIL

brent.carlson@logiclibrary.com

We believe that what they're called isn't important, instead what is important is making knowledge gained solving one problem useful when solving another similar problem (or even more simply to document what we've done and why we've done it when we have to go back and maintain the code). To effectively do this we have to have a means of capturing, refining, and sharing the knowledge we gain. We believe that capturing this knowledge as a pattern is a great way to do this (and by the way, in doing so, we also believe that you are going to learn more about your design than you expected, very likely including some ways to improve it).

We're going to look at a business application pattern that we discovered and documented. We'll take you through the steps we went through as we discovered and captured it. As we progress through the steps, we'll show you not only what we learned as we captured this pattern, but also things we learned as we captured other patterns.

We'll look at our initial solution to a problem and why we thought it might be a pattern – even with just one instance of the problem. In our next installment, we'll look at another business problem to which we were able to apply our original pattern and, more important, how applying the pattern helped us to define and refine it. In our final installment, we'll look at applying the pattern (with others) to the construction of applications, components, and Web services.

The Problem

The first time we encountered the problem of managing configurable balance information was in the area of warehouse management. We needed some way to manage cached product balances (how much of any particular product was in the warehouse), but we needed to do so in a way that would allow users of our product balances to keep the balances on only the information they wanted. For example, some businesses need to keep track of a product's lot – the "batch" that it was part of. This is especially important when the product is dyed cloth, for example, since only bolts of cloth from the same lot will have exactly the same color.

Other businesses don't care about lots at all. Some businesses use detailed product balances (down to individual zones and bins) for defining pick lists, while other businesses need to track only gross amounts to ensure that product orders can be fulfilled within specified lead times. Our product balances needed to support all of these situations flexibly and without a lot of overhead.

We also knew that we had situations in which there would need to be more product information taken into account when caching the balance. For example, sometimes a product may have multiple colors, so color needs to be taken into account in the balances. On the other hand, many companies will have a different product for each color, so they don't need to define additional attributes to manage color within a specific product. All these possibilities added up to the need for a lot of flexibility in the information the balances could be kept for.

The Solution

Our initial solution was to create a multivalued map that tied items of interest (i.e., the attributes we needed to track) to the cached balance. In other words, we created a lightweight object called the ProductBalanceMapKey that contained the things we supported caching balances over and used it as a key in a map to a number.

Figure 1 shows an example of our simple cached product balances implementation. In our original solution each product had separate balances, so the product ID didn't need to be part of the ProductBalanceMapKey. However, we were interested in the balances for each warehouse and lot, so they do show up in our example. (We're interested in the warehouse because it allows us to know what the delivery time to a customer is – quick if they are next door and slow if they are in a different country.)

While there could be an entry for every possible ProductBalanceMapKey in our map, that wouldn't be the most efficient design. Instead of returning a null when there isn't an entry for a key, we would prefer to return a balance of zero. This behavior could be coded by every user of the map. However, it makes more sense to encapsulate this behavior and the map into a CachedProductBalances class.

Besides being able to retrieve balances, we also needed to keep them up to date. This design allows us to do that – simply build a ProductBalanceMapKey with the right information and modify or add the associated value. This behavior is also encapsulated by the CachedProductBalances class.

The retrieve and update methods are written in terms of the ProductBalanceMapKey. They only use equals (during the map lookup) and don't need to know the details of what's in the key. This design allows us to fulfill our requirement of adding information we're keeping balances over by changing the ProductBalanceMapKey to add the information. However, this doesn't allow us to easily ignore information, for example, in the case where lots are passed in, but we don't need to keep our product balances based on lots.

We resolved this problem by introducing a usage indicator within CachedProductBalances for each piece of information in the key (a Boolean array). We then implemented the CachedProductBalances methods to replace the value in the key with a placeholder whenever the indicator for that key entry is turned off. Thus, for our example above we have a usage array set to all true. If we were to change the lot indicator to false, then the methods on CachedProductBalances would (as part of their processing) replace the values for lot with a placeholder.

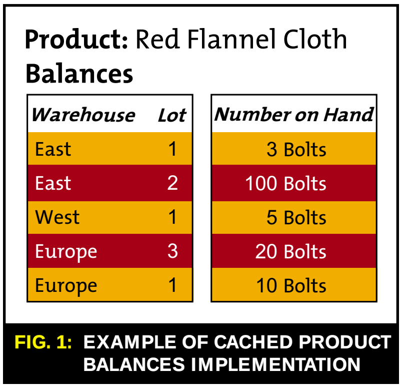
The implementation of updateBalance() accumulates entries with duplicate keys; in this case lot values are ignored during the accumulation process. Note that CachedProductBalances doesn't know it is working with a lot. If we think of the key as being an array of information, it simply knows that the information in the same relative position is false and should be replaced.

What we ended up with was a special kind of map key, a key that contained information (that could be changed programmatically), with an associated usage array (that could be dynamically configured). This simple design is shown in Figure 2.

The Start of a Pattern

At this point, we'd normally be happy that we were done and move on to the next problem. However, in this case we stopped and took a longer look at what we'd done. Why this particular problem? On our project we had a number of team members who were very experienced in the business domains we were targeting. They were able to use this experience to tell us that similar problems were going to come up. For example, they knew that we were going to need to cache account balances both for general ledger accounts (for financial reporting and analysis) and customers (for credit checking purposes).

Had we not had the advantage of our domain experts' insight, we would have waited until at least the second or third time we encountered a similar problem before we even thought of our original solution as a candidate for a pattern. Trying to generalize a pattern from a single use is difficult, time-consuming, and error-prone – if not impossible. In general, when evaluating existing designs, you have to be on the lookout for similarities, but don't go overboard – not everything is a pattern and some patterns aren't worth capturing.

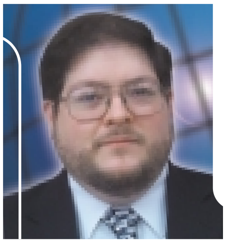


Product: Red Flannel Cloth Balances

Warehouse	*Lot*	*Number on Hand*
East	1	3 Bolts
East	2	100 Bolts
West	1	5 Bolts
Europe	3	20 Bolts
Europe	1	10 Bolts

FIG. 1: EXAMPLE OF CACHED PRODUCT BALANCES IMPLEMENTATION

Probably the biggest driving force behind pattern definition and capture is the cost/benefit trade-off related to capturing a pattern. Who will be the consumers of our pattern, and how much will they benefit from our pattern documentation? We're very pragmatic in the capture of our patterns. When we identify a design as a candidate pattern we begin by being very informal about documentation – especially when the developer who worked on the first instance is going to be around to answer questions. An important factor in how much information we choose to document is how broadly we expect the pattern to be used. If it's only going be used within the team, often informal documentation is sufficient – there is somebody a new user can talk to about the pattern and the concepts behind it. On the other hand, when a pattern is something that will go outside your organization, it is a good idea to formally document it, since the customer normally can't simply stop by and ask you a question.



ABOUT THE AUTHOR

James Carey, senior software engineer, is lead architect for IBM's WebSphere Business Components project. This project develops EJB components that provide business content to support application development. With Brent Carlson James has coauthored *SanFrancisco Design Patterns: Blueprints for Business Software*, and *Framework Process Patterns: Lessons Learned Developing Application Frameworks*.

E-MAIL

jecarey@us.ibm.com

Remember that candidate patterns aren't worth spending a lot of time documenting. The second and third applications of the pattern are very likely to dramatically change the pattern. Spending lots of time documenting your candidate pattern is a waste of time, especially if you discover you were wrong and the candidate pattern turns out not to be a pattern (or simply a pattern that it isn't worth your time to document). Often, just documenting what you've done in a simple UML model with associated notes discussing why you think this might be a pattern and some of the key ideas behind it (in other words, leaving a trail of bread crumbs for you to follow later as you refine your pattern) is plenty of detail for this level of pattern definition.

The most important notes to capture in a candidate pattern are any alternatives you explored and why you rejected them. For example, in our case study we looked at simultaneously having ProductBalanceMapKeys of differing sizes. We rejected this design because although it could be done, there was too much overhead associated with it. We may come back later and readdress these alternatives as we apply the pattern to subsequent uses. However, this documentation serves as a starting point and, when the pattern changes, it helps us understand whether or not the pattern still applies in the original case.

As you refine your pattern, you may find that it no longer applies to the first case that started you thinking of the design as a pattern. In other words, the first case may have been an exception to the pattern. The key is not to get trapped by trying to have the pattern solve every single case. *SanFrancisco Design Patterns* has families of patterns that solve similar problems (proxy, mediator, facade), and you may have run into a similar situation. If the pattern is getting in the way of solving the problem, or adding complexity without value, it is time to start questioning the value or applicability of the pattern.

Even if the original problem turns out to be covered by the final pattern, you need to keep in mind that you may need to rework the original solution. Again this is a trade-off. The solution you have is working, so you don't have to make it adhere to the final pattern. You need to make sure that something wasn't discovered that would be of value in the original case and you have to weigh the value of consistency (which becomes more important when the pattern is exposed to customers and also when maintaining the solution).

The identification of a potential pattern may make you rearrange your development schedule. In our case we started looking at account balances earlier than we originally planned and we put one of our better developers on it. In addition, as the technical leads we paid more attention to this component than we originally planned, periodically reviewing the pattern as it was being developed and actively looking for other appropriate applications of the pattern.

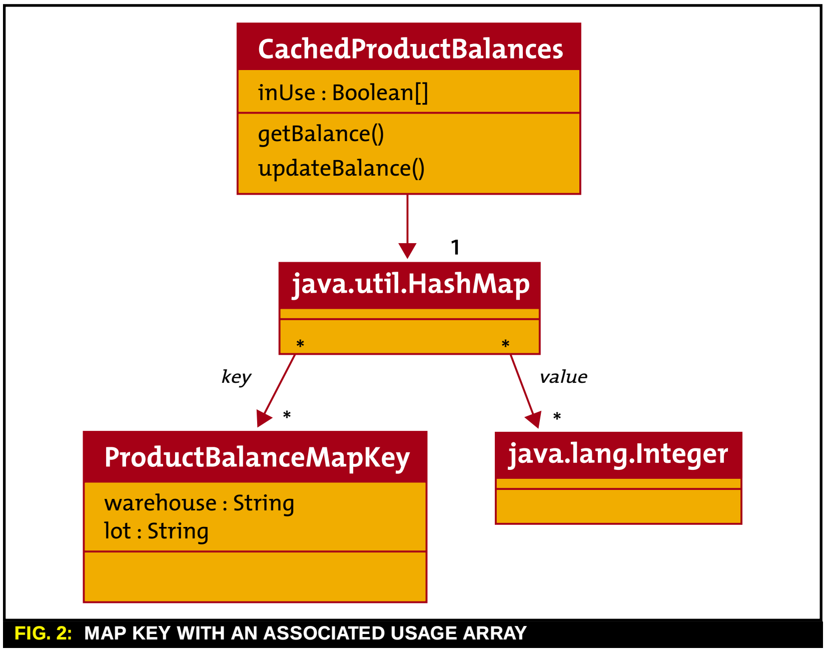


FIG. 2: MAP KEY WITH AN ASSOCIATED USAGE ARRAY

Summary

- When identifying potential patterns:
  - Use your team's technical and domain experts to guide you to interesting areas within your design.
  - Wait for the second or even third occurrence. Be watchful for these.
- Document patterns to the right level.
  - Candidate pattern documentation shouldn't consume lots of time.
  - Patterns for team use can be very informal.
  - Patterns for customer use usually end up being very formal.
- Don't get trapped by patterns (or the thrill of capturing them).
  - Sometimes the original problem can't be solved by the final pattern.
  - Sometimes there isn't a pattern.
- Consider reapplying the pattern once it is refined.
  - May be something new of benefit for the original implementation.
  - May need consistency across implementations.
- Potential patterns may cause development plan changes.
  - Look at other uses earlier.
  - Change assignments to more experienced developers.
  - Team leaders may be more involved.

In part 2 of this series, we'll show how our account balances requirements affected our original candidate pattern and how we used that information to provide both our pattern and our original product balances implementation.

THE LARGEST WEB SERVICES, JAVA, XML AND .NET CONFERENCE AND EXPO IN THE WORLD!

Boston 2003 | London 2003 | Berlin 2003 | Hong Kong 2003

JDJ EDGE conference&expo

web services EDGE conference&expo

.NET EDGE conference&expo

XML EDGE conference&expo

Register by February 14th SAVE UP TO $400

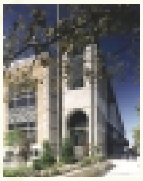
Register by March 14th SAVE UP TO $200

3rd Annual

Web Services Edge EAST Conference & Exposition

FINAL PROGRAM

Connecting the Enterprise with Web Services, Java, XML, & .NET



March 18–20, 2003

Hynes Convention Center
Boston

CALL TODAY TO REGISTER 201-802-3058 OR REGISTER ONLINE AT www.sys-con.com

Event Sponsors: Microsoft, Sun microsystems, Sybase, IONA, Digital Evolution, Compuware, Oracle, hp invent, Rational the e-development company, Altova

Media Sponsors: Java Developer's Journal, JavaWorld, Web Services Journal, Linux Business Week, XML Journal, SAMS, BASIS, PerfectXML, .NET Developer's Journal, SD Times, asp.netPRO, WebSphere Developer's Journal, BEA WebLogic Developer's Journal, wireless Business & Technology, CF Advisor, ColdFusion Developer's Journal, PowerBuilder Developer's Journal

OWNED BY SYS-CON MEDIA

PRODUCED BY SYS-CON EVENTS

Special Insert: Web Services Edge East Conference & Expo

Featuring FREE Tutorials, Training Sessions, Case Studies and Exposition

Special Insert: Web Services Edge East Conference & Expo

Featuring FREE Tutorials, Training Sessions, Case Studies and Exposition

web services EDGE conference&expo

JDJ EDGE conference&expo

.NET EDGE conference&expo

XML EDGE conference&expo

THE LARGEST WEB SERVICES CONFERENCE & EXPO

FEATURES & ATTRACTIONS

- **3 Days Packed with Education and Training**
- **4 Keynotes & 3 Panel Discussions**
- **60 Hard-Hitting and Current Seminars**
- **FREE .NET Tutorial with Russ' Tool Shed**
- **Web Services & XML Tutorials**
- **Java University Certification Training**
- **HOT** Industry-Leading Certification Programs
- **1 Day Mobile .NET Tutorial**
- **INFORMATIVE** Round Table Discussions
- Opening Day Welcome Reception
- SAMS Meet the Authors Hot Topics Lounge
- **COMPELLING** Case Studies & Best Practices
- **FEATURED** Product Demonstrations on the Show Floor
- **RIVETING** Real-time SYS-CON Radio Interviews

WHO SHOULD ATTEND

- **Software Developer**
- **Software Engineer**
- **Development Manager**
- **Application Developer**
- **Technical Director**
- **Analyst/Programmer**
- **IT Manager**
- **Technical Architect**
- **Team Leader**
- **Software Consultant**

Keynotes & Highlighted Speakers

KEYNOTE SPEAKERS

John Magee

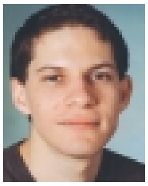
Vice President, Oracle9i

John Magee is vice president, Oracle9*i*, at Oracle. He has more than 14 years' experience in the enterprise software industry and has held positions in product development, product management, and product marketing. In his current role, Magee manages technical product marketing for Oracle's application server and development tools products, and is responsible for evangelizing Oracle technology initiatives around J2EE, XML, and Web services.

Mark Herring

Director Java, Web Services & Tools Business

Mark Herring is responsible for helping to define, set, and drive Sun Microsystems' product direction in the Java, Web Services & Tools Business. Prior to his current role, Herring was director of corporate strategy & planning, looking after Sun's interest in the Project Liberty Alliance and Network Identity. Herring joined Sun Microsystems in October 1999 as a result of Sun's acquisition of Forte Software. Forte Software was a leading provider of enterprise-class development and integration products. During his four years at the company, he ran several aspects of Forte's marketing organization, including product marketing and the Web channel.



Miguel de Icaza

Cofounder and CTO

As the founder and leader of the GNOME Foundation, Miguel de Icaza is one of the foremost luminaries in the Linux development community. With his seemingly boundless energy, de Icaza has galvanized the effort to make Linux accessible and available to the average computer user. He brings this same excitement to his role as CTO of Ximian. de Icaza was instrumental in porting Linux to the SPARC architecture and led development of the Midnight Commander file manager and the Gnumeric spreadsheet. He is also a primary author of the design of the Bonobo component model, which leads the way in the development of large-scale applications in GNOME.

Mark Hapner

Distinguished Engineer, Sun Microsystems

Mark Hapner is a Sun Distinguished Engineer and is currently lead architect for Java™ 2 Platform, Enterprise Edition (J2EE™). He has guided the overall architecture for J2EE 1.2, 1.3, and now the upcoming 1.4 release. In March of 1996, he joined Sun's JavaSoftware Division to participate in the development of the Java database connectivity API (JDBC). Following that, he was co-spec lead of the Enterprise JavaBeans specification and spec lead of the Java Message Service specification.

Simon Phipps

Chief Technology Evangelist, Sun Microsystems

Simon Phipps, currently chief technology evangelist at Sun Microsystems, speaks frequently at industry conferences on the subject of technology trends and futures. He was previously involved in OSI standards in the 1980s, in the earliest collaborative conferencing software in the early 1990s, and in introducing Java and XML to IBM.

Dave Chappell

VP, Chief Technology Evangelist, Sonic Software

Dave Chappell is the vice president and chief technology evangelist for Sonic Software. He has more than 18 years of industry experience building software tools and infrastructure for application developers, spanning all aspects of R&D, sales, marketing, and support services. Chappell has published in numerous technical journals, and is currently writing a series of contributed articles for *Java Developer's Journal*.

Eric Newcomer

Chief Technology Officer, IONA

In the role of chief technology officer at IONA, Eric Newcomer is responsible for IONA's technology roadmap and the direction of IONA's Orbix E2A e-Business Platforms as relates to standards adoption, architecture, and product design. Newcomer joined IONA in November 1999, and most recently served as IONA's vice president of engineering, Web Services Integration Products. He is a member of the XML Protocols and Web Services Architecture working groups at the W3C and IONA's Advisory Committee representative to UDDI.org.

Dean Guida

CEO and President, Infragistics

Dean Guida is CEO and president of Infragistics and was CEO and a cofounder of ProtoView Development Corporation. Mr. Guida has over 15 years of experience in the technical industry and oversees all aspects of the company's business operations and corporate direction. He is also responsible for cultivating strategic alliances and other external relationships, as well as managing corporate financial affairs.

Conference Advisory Board

Sean Rhody
Editor in Chief,
Web Services Journal
Partner, CSC

Alan Williamson
Editor in Chief,
Java Developer's Journal
Chief Technology Officer, n-ary

Derek Ferguson
Editor in Chief,
.NET Developer's Journal
Chief Technology Evangelist,
Expand Beyond Corp.

Bob Familiar
.NET Architect,
Microsoft
New England

Thomas Robbins
Senior Technology Specialist,
Microsoft
New England

Hitesh Seth
Editor in Chief,
XML-Journal
Chief Technology Officer, ikigo

J.P. Morgenthal
Chief Services Architect, Software AG

Special Insert: Web Services Edge East Conference & Expo

Featuring FREE Tutorials, Training Sessions, Case Studies and Exposition

Special Insert: Web Services Edge East Conference & Expo

Featuring FREE Tutorials, Training Sessions, Case Studies and Exposition

Conference at-a-Glance

THE LARGEST AND THE MOST

		JAVA	WEB SERVICES	.NET
TUESDAY MARCH 18	8:00AM —4:00PM	Registration Open		
	9:00AM —9:50AM	(JV1) Squeezing the Best Out of Java Alan Williamson, Java Developer s Journal	(WS1) Web Services Infrastructure Carl Sjogreen, BEA	(NT1) .NET Framework Overview Bob Familiar, Microsoft
	10:00AM —10:50AM	Web Services Keynote: John Magee, Oracle		
	11:00AM —11:50AM	(JV2) Testing Your Java Using JUnit Kyle Gabhart, LearningPatterns	(WS2) Web Services Management James Phillips, Actional	(NT2) Introduction to ASP.NET Russ Fustino, Microsoft
	1:00PM —1:50PM	WS-I Panel: A Road Map for Web Services Standards - Moderated by Rob Cheng, WS-I		
	2:00PM —2:50PM	.NET Keynote: The MONO Project - Miguel de Icaza, Ximian		
	3:00PM —3:50PM	(JV3) Building/Deploying the Ant Way Kyle Gabhart, LearningPatterns	(WS3) Strategies for Using Databases in a World of Web Services Mike Lehmann, Oracle	(NT3) Introduction to VB.NET Russ Fustino, Microsoft
	4:00PM —4:50PM	(JV4) Unlocking the Secrets of JDK1.4 Raghavan Srinivas, Sun Microsystems	(WS4) Using Web Services to Integrate J2EE and .NET Enterprise Applications - Odysseas Pentakalos, SYSNET International	(NT4) How to Develop an End-to End .NET Connected Appilication Allan da Costa Pinto, Microsoft
WEDNESDAY MARCH 19	8:00AM —4:00PM	Registration Open		
	9:00AM —9:50AM	(JV5) Java APIs for Web Services Security Standards Sang Shin, Sun Microsystems	(WS5) Combining BPM and BRM Technologies: A Major Step Towards Corporate Agility Henry Bowers, ILOG	(NT5) .NET: The Virtualized Execution Engine Yahya Mirz, Aurora Borealis
	10:00AM —10:50AM	Java Keynote: Mark Herring, Sun Microsystems		
	11:00AM —6:00PM	EXPO OPEN 11:00 a.m. - 6:00 p.m.		
	11:00AM —11:50AM	(JV6) To Not Swing Is to SWT! The Swing Alternative - IBM	(WS6) Web Services for Real-Time Data Access in an Industrial Setting Stephan Van Dijck, ABB/SKYVA	(NT6) Introduction to DotGNU Barry Fitzgerald, DotGNU
	12:00PM —2:00PM	BREAK & EXPO		
	2:00PM —2:50PM	.NET Panel Discussion - Moderated by Derek Ferguson, .NET Developer s Journal		
	3:00PM —3:50PM	(JV7) Unlocking the Power of XML Hitesh Seth, ikigo	(WS7) Web Services Architecture: The Next Big Spec. from the Mouths of the W3C Eric Newcomer, IONA (moderator)	(NT7) Introduction to SSCLI Yahya Mirz, Aurora Borealis
	4:00PM —4:50PM	(JV8) Java and .NET Derek Ferguson, Expand Beyond	(WS8) Web Services: Next Steps After the Hype Claire Dessaux, Oracle	(NT8) Mobile Development with the Compact Framework Brad Mccabe, Infragistics
THURSDAY MARCH 20	8:00AM —4:00PM	Registration Open		
	9:00AM —9:50AM	(JV9) Writing SOAP Services Nigel Thomas, SpiritSoft	(WS9) Web Services Best Practices Chris Peltz, HP	(NT9) Best Practices for .NET Development Joe Stagner, Microsoft
	10:00AM —10:50AM	.NET Keynote - Jesse Liberty, Liberty Associates		
	11:00AM —4:00PM	EXPO OPEN 11:00 a.m. - 4:00 p.m.		
	11:00AM —11:50AM	(JV10) Working with Data the JDO Way Patrick Linsky, SolarMetric	(WS10) Web Services Startups: Telltales of the Future Simeon Simeonov, Polaris Venture Partners	(NT10) Best Practices for ADO.NET Development Thom Robbins, Microsoft
	12:00PM —2:00PM	BREAK & EXPO		
	2:00PM —2:50PM	Java Panel - The Future of Java , Moderated by Alan Williamson, Java Developer s Journal		
	3:00PM —3:50PM	(JV11) Enterprise: The Next Generation Mark Hapner, Sun Microsystems	(WS11) Open Standards for Web Services Messaging Dave Chappell, Sonic Software	(NT11) How to Debug with .NET Tony Denbow, STAR Information Technology
	4:00PM —4:50PM	(JV12) Overcoming the Challenges of J2ME Dr. Jeff Capone, Aligo	(WS12) Web Services Security Marc Chanliau, Netegrity	(NT12) XML and Web-Enabling Legacy Applications Using BizTalk Mike Cramer, Microsoft

COMPREHENSIVE *i*-TECHNOLOGY DEVELOPER CONFERENCE OF THE YEAR!

XML	VENDOR
(XM1) XML - A Manager s Guide JP Morgenthal, Software AG	Visit www.sys-con.com for details
(XM2) OASIS Standards Update Karl Best, OASIS	(VN2) The XMLSPY 5 Enterprise Edition Development Environment Trace Galloway, Altova
(XM3) A Definitive Introduction to XML Schemas Aaron Skonnard, DevelopMentor	(VN3) SOAP and Java: Marrying Them Off Skip Marler, Parasoft
(XM4) XML in Print - XSL:FO Frank Neugebauer, IBM	Visit www.sys-con.com for details
(XM5) XML Security Integration Challenges Phil Steitz, American Express	(VN5) Process-Centric Enterprises Eric Pulier, Digital Evolution
(XM6) Case Study: XML in Life Sciences Tim Matthews, Ipedo	(VN6) Pattern Driven Application Development Tom Shore, Compuware
(XM7) Using XML for EAI - Best Practices Dan Enache, TIBCO	(VN7) Managing the Developer Relationship Mike Bellissimo, Sun Microsystems
(XM8) Take XML with You: XML and Mobile Computing - Hitesh Seth, ikigo	(VN8) Web Services Diagnostics Dave Seidel, Mindreef
(XM9) XML, Ontologies and the Semantic Web - Ayesha Malik, Object Machines	Visit www.sys-con.com for details
(XM10) X Query Mike Champion, Software AG	(VN10) Model Driven Development of Web Services in UML for the J2ME Bill Graham, Rational Software
(XM11) XPath & XSLT 2.0 BEA Kurt Cagle, Cagle Communications	(VN11) Why Web Services Management? Jon Atkins, HP
(XM12) Third Generation XML Tools Michael Leventhal, Commerce One	Visit www.sys-con.com for details

JAVA UNIVERSITY℠ PROGRAM

9:00AM —5:00PM

Web Services Programming Using Java™ Technology and XML

This one-day seminar provides in-depth knowledge on Web services and shows how to develop Web services using the Java programming language and XML, the technologies of portable code and portable data respectively.

9:00AM —5:00PM

Java 2 Platform Programmer Certification Fast Path

This session, developed and delivered by Philip Heller, author of the two leading Java technology certification preparation manuals, helps to prepare you for the Sun Certified Programmer for the Java 2 Platform exam. Philip provides code-level, detailed review of the skills and knowledge needed to confidently approach the exam.

9:00AM —5:00PM

Java 2 Platform Architect Certification Fast Path

This intense one-day session helps prepare attendees to pass the Sun Certified Enterprise Architect for J2EE Technology exam. This session provides an overview of the components comprising the J2EE architecture as a whole, emphasizes the incorporation of J2EE technology into an architecture, and reviews each of the certification exam s testing objectives.

FAST TRACKS & TUTORIALS

9:00AM —5:00PM

XML Certified Developer Fast Path

This tutorial is for programmers who have some knowledge of XML and related technologies and would like to pass the IBM Certified Developer Test 141 on XML and Related Technologies.

9:00AM —5:00PM

Russ Tool Shed

Join Russ as he shows you how to use Visual Studio.NET.

9:00-12:15 - Introduction to Web Services Using VS.NET

1:00-2:30 - Advanced Web Services Using ASP.NET

2:45-4:15 - .NET Remoting Essentials

9:00AM —5:00PM

Mobile .NET

In this session, Derek Ferguson, editor-in-chief of .NET Developer s Journal, will give you a thorough introduction to the use of .NET with all manner of mobile computing devices.

Special Insert: Web Services Edge East Conference & Expo

Featuring FREE Tutorials, Training Sessions, Case Studies and Exposition

Special Insert: Web Services Edge East Conference & Expo

Featuring FREE Tutorials, Training Sessions, Case Studies and Exposition

SUN MICROSYSTEMS Java University℠ Program

Java University℠ Program: Aggressive, code-level training courses for experienced developers using Java™ technology, brought to you by Sun Microsystems, Inc. Attend seminars designed by industry luminaries and recognized experts. Sessions cover Sun certification and Web services technology. Whether you're a beginner or a veteran developer, architect, or software engineer, you'll benefit from these value-packed full-day courses. Register now. Seating is limited.

The Java University℠ program complements this year's Web Services Edge conference by offering 3 full-day training lectures for experienced software developers, architects, and engineers.

Java University℠ Program Take-Aways:

- Training designed and presented by expert Java technology engineers
- Student guides full of source code, examples, references and copies of instructors' materials
- Free Web-based training courses from Sun Educational Services

Tuesday, March 18, 2003

Web Services Using Java™ Technology and XML

SANG SHIN, SUN MICROSYSTEMS, INC.

Who Should Attend

Web services designers and programmers, application developers, and programmers using the Java programming language who have experience using the Java™ 2 Platform, Enterprise Edition (J2EE™).

Prerequisites

Experience using the Java programming language and basic knowledge of XML

Overview

This one-day seminar provides in-depth knowledge on Web services and shows how to develop Web services using the Java programming language and XML, the technologies of portable code and portable data respectively.

The session will start with an introduction on fundamental concepts and characteristics of Web services. This will be followed by a detailed explanation of how to implement, describe, register, discover, and invoke Web services using core Web services standards - Simple Object Access Protocol (SOAP); Web Services Description Language (WSDL); and Universal Description, Discovery, and Integration (UDDI). In addition, the ebXML standard, which defines the framework for the global electronic marketplace will be talked about in detail. Also, the tools for building and deploying Web services will be discussed. Each topic will be presented with concrete examples and demonstrations when possible.

Attendees will also learn how to use standard Java APIs for Web services, mainly Java API for XML Messaging (JAXM), Java technology API for XML-based RPC (JAX-RPC), and Java technology API for XML Registries (JAXR) for developing and deploying Web services.

Benefits

- Learn the fundamental concepts and characteristics of Web services. Gain detailed understanding on core Web services standards: SOAP, WSDL, UDDI.
- Gain a detailed understanding of ebXML, the standard framework for electronic business.
- Learn Java programming language APIs for Web services – JAXM, JAX-RPC, JAXR

Wednesday, March 19, 2003

Java™ 2 Platform: Programmer Certification Fast Path

PHILIP HELLER, PRESIDENT, HELLER ASSOCIATES

Who Should Attend

This session is designed for programmers who have some exposure to the Java™ programming language, and are ready to prepare for the Sun Certified Programmer for Java 2 Platform exam.

Prerequisites

Object-oriented software development experience and familiarity with the syntax and structure of Java technology-based development.

Overview

The development community recognizes that competency developing solutions using Java technology is vital to productivity, reaffirms your value to your organization, and increases your career advancement opportunities. This session, developed and delivered by Philip Heller, author of the two leading Java technology certification preparation manuals, helps to prepare you for the Sun Certified Programmer for the Java 2 Platform exam. Philip provides code-level, detailed review of the skills and knowledge needed to confidently approach the exam.

Benefits

- Receive an intensive review of the advanced topics covered on the Sun Certified Programmer for the Java 2 Platform Exam
- Increase your understanding and knowledge of Java programming language syntax and structure
- Prepare for the exam by reviewing practice tests and questions
- Gain a strong understanding of Java fundamentals

Thursday, March 20, 2003

Java™ 2 Platform: Architect Certification Fast Path

SIMON ROBERTS, TECHNOLOGY EXPERT AND COURSE DEVELOPER, SUN MICROSYSTEMS, INC.

Who Should Attend

This session is designed for enterprise application architects, system analysts, experienced technologists, and developers using Java™ technology seeking certification as an architect for the Java™ 2 Platform, Enterprise Edition (J2EE™).

Prerequisites

Understand the benefits of Java technology solutions; experience with object-oriented analysis and design; familiarity with concepts of distributed computing.

Overview

Many of the solutions in today's "Net economy" are, or soon will be, developed using the Java 2 Platform, Enterprise Edition (J2EE) architecture. Gaining recognized competency architecting J2EE platform-based solutions is vital to your success as an architect, reaffirms your value, and increases your career opportunities.

Developed and presented by Mark Cade, this intense one-day session helps prepare attendees to pass the Sun Certified Enterprise Architect for J2EE Technology exam. This session provides an overview of the components comprising the J2EE architecture as a whole, emphasizes the incorporation of J2EE technology into an architecture, and reviews each of the certification exam's testing objectives. Multiple real-world case studies are used to demonstrate correctly architected J2EE technology–based solutions and pinpoint key topics presented within the architect exam.

Additionally, you will learn how to interpret exam objectives, what each of the three exam phases contains, and clear guidelines and resources to use after the course.

Benefits

- Receive an intensive review of the topics covered on the Sun Certified Enterprise Architect for the Java 2 Platform, Enterprise Edition Exam
- Increase your understanding and knowledge of successfully architecting solutions using J2EE technology
- Understand the system qualities: scalability, availability, extensibility, performance, and security
- Understand trade-offs of different architectural choices as they pertain to system qualities.
- Describe the benefits and weaknesses of potential J2EE technology–based architectures
- State benefits and costs of persistence management strategies
- Review real-world case studies of J2EE technology–based architecture
- Prepare for the exam by reviewing practice tests and questions

Major Sponsors at the World's Largest Web Services, Java, XML, and .NET Conference and Expo

2003 East Gold Sponsors

COMPUWARE

Compuware will be the best worldwide provider of quality software products and services designed to increase productivity. We will continue to create practical solutions that meet our customers needs and surpass their expectations. We will provide an environment for our employees where excellence is encouraged and rewarded and where diversity is promoted at all levels of the company. Compuware is built upon a diverse employee population with a set of key attributes that is common to every individual: Secure, Moral, Ethical, In Agreement with Economic Purpose, Team Player, Other Oriented, Positive, Optimistic, Energetic Achiever and Bright.

Microsoft®

Introducing Visual Studio .NET; visionary yet practical, the single comprehensive development tool for creating the next generation of applications has arrived. Developers can use Visual Studio .NET to:

- Build the next-generation Internet.
- Create powerful applications fast and effectively.
- Span any platform or device.

Visual Studio .NET is the only development environment built from the ground up for XML Web services. By allowing applications to share data over the Internet, XML Web services enable developers to assemble applications from new and existing code, regardless of platform, programming language, or object model.

ORACLE

Oracle Corporation (Nasdaq: ORCL) is the world's largest enterprise software company, providing enterprise software to the world's largest and most successful businesses. With annual revenues of more than $9.4 billion, the company offers its database, tools, and application products, along with related consulting, education, and support services. Headquartered in Redwood Shores, California, Oracle is the first software company to develop and deploy 100 percent Internet-enabled enterprise software across its entire product line: database, server, enterprise business applications, and application development, and decision support tools.

2003 East Silver Sponsor

hp invent

HP is a leading provider of products, technologies, solutions and services to consumers and business. The company's offerings span IT infrastructure, personal computing and access devices, global services, and imaging and printing. Our $4 billion (U.S.) annual R&D investment fuels the invention of products, solutions and new technologies, so that we can better serve customers and enter new markets. We invent, engineer, and deliver technology solutions that drive business value, create social value, and improve the lives of our customers.

2003 East Featured Sponsor

Sun microsystems

Sun was founded with one driving vision. A vision of computers that talk to each other no matter who built them. A vision in which technology works for you, not the other way around. While others protected proprietary, stand-alone architectures, we focused on taking companies into the network age, providing systems and software with the scalability and reliability needed to drive the electronic marketplace.

2003 East Corporate Sponsors

IONA

IONA is the leading e-Business Platform provider for Web Services Integration. IONA Orbix E2A enables the flow of information across disparate systems and the liberation of application assets for new business opportunities. For additional information about IONA, visit our Web site at http://www.iona.com.

Rational®
the e-development company™

Rational Software provides a software development platform that improves the speed, quality, and predictability of software projects. This integrated, full life-cycle solution combines software engineering best practices, market-leading tools, and professional services. Ninety-six of the Fortune 100 rely on Rational tools and services to build better software, faster. Additional information is available on the Internet at http://www.rational.com

SAMS

Sams Publishing has more than 500 titles in print, and is one of the most successful computer book publishers in the world. From introductory tutorials to comprehensive reference books, Sams Publishing focuses on teaching tomorrow's programmers, developers and system administrators the skills they need to build and manage emerging technologies.

SYBASE

Sybase has always delivered solutions that help customers to share data, is platform independent, and integrates everything: platforms, application servers, components, databases, portals, processes, message brokers, and mobile/wireless. Our technologies have the insight that will promote ease of use, leverage best practices, ensure positive ROI, and help your organization build a successful, pragmatic strategy based on next-generation technologies. Check us out at http://www.sybase.com/solutions/e-business

DIGITAL EVOLUTION
The Business Value of Web Services

DE solutions and thought leadership deliver the value of Web services to business users today. DE unleashes the value of underlying IT assets, giving enterprises the freedom to innovate and evolve business processes within a standards-based, vendor neutral, loosely coupled framework. Some of the largest corporations in the world rely on DE's Web services solutions. www.digev.com

Special Insert: Web Services Edge East Conference & Expo

Featuring FREE Tutorials, Training Sessions, Case Studies and Exposition

REGISTER ONLINE TODAY @ SYS-CON.COM

JDJ EDGE conference&expo

web services EDGE conference&expo

.NET EDGE conference&expo

XML EDGE conference&expo

Don't Miss the Developer Event of 2003

Meet with the industry experts, professionals, and visionaries connecting the enterprise with today s i-technology!

Conference
March 18–20, 2003

Exposition
March 19–20, 2003

Hynes Convention Center, Boston

WEB SERVICES EDGE CONFERENCE PAST SPONSORS & EXHIBITORS

Actional AltoWeb A Data Quality Company ANTENNA HOUSE Covasoft CTI DORIG eXcelon FatWire SOFTWARE webGain

altio attachmate CACHEON apress CORDA ASP street atg FORUM SYSTEMS

WebCharts webputty IOPSIS SOFTWARE

TogetherSoft SilverStream KENAMEA sitraka Rational the e-development company™ TIBCO systinet SEAGULL

ca ADOS aligo bea Bostning CAPE CLEAR COMPUWARE

Data Junction DataMirror HiTSOFTWARE ParaSoft InCert INTERDATA infoShark IONA

LEARNINGPATTERNS.COM MERANT NOVALABS pingtel SAVAJE

shinka sonic SOFTWARE spiritsoft TOWER ORACLE WHITEHILL ORACLE

Borland® xml spy Matisse IPEDO PolarLake VORDEL® Cohesion Technologies for eBusiness Altaworks

AXIOMATIC DESIGN bowstreet CSC hp iam pingtel

MFG NEXTEL Object Design PRAMATI INFRAVIO lazy MERCURY INTERACTIVE

Jinfonet KeyLabs macromedia MelissaDATA METASERVER MICRO FOCUS MIRACLE VULTUS webMethods

oakgrove OPENDEMAND SYSTEMS PointBase RSA SECURITY SYBASE Trivera WAKESOFT

ObjectBuilders ObjectVenture Optimyz plumtree serena teamshare Trellix

E.PIPHANY INTEGRATES WEBSPHERE INTO CRM ARCHITECTURE

(San Mateo, CA) – E.piphany, Inc., a full-suite provider of customer relationship management (CRM) software, has integrated IBM WebSphere into the E.piphany E.6 component-based platform. The integration combines the industry's first J2EE-based CRM suite, featuring integrated sales, marketing, service, and analytic capabilities, with the industry's leading J2EE application server. The result is a single-source, J2EE-based CRM solution suited for businesses standardizing their IT environments on component-based architectures.

"Our corporate IT strategy is based on J2EE technologies because of the greater flexibility and interoperability inherent in this proven set of standards," said John O'Donovan, first vice president of the e-business partnership at Mutual of Omaha. "Our recent selection of the E.piphany E.6 software system running on IBM WebSphere will help us achieve superior scalability and flexibility with existing systems."

www.epiphany.com

WORLDSPAN, IBM SIGN NEW STRATEGIC TECHNOLOGY AGREEMENT

(Atlanta, GA, and White Plains, NY) – Worldspan and IBM have signed a strategic technology agreement, valued at more than $350 million over five years, that will enable Worldspan to integrate additional IBM technology into Worldspan's growing line of travel solutions and extend its use of IBM's Transaction Processing Facility (TPF) platform. TPF is the high-bandwidth operating system and transaction processor that forms much of the software infrastructure for the travel industry.

Worldspan will extend the TPF operating system by integrating it with IBM WebSphere Internet infrastructure software. The company is also enhancing its overall IT development initiatives through access to a broader range of IBM products, including IBM WebSphere Application Server and development tools and consulting resources through IBM Global Services. Under the new contract, Worldspan expects to achieve lower costs and greater capabilities for its customers.

www.worldspan.com

DIRIG'S E-BUSINESS SOLUTIONS TO LEVERAGE WEBSPHERE 5.0

(Nashua, NH) – Dirig Software, a leading developer of enterprise performance management solutions for e-business, has announced that the Fenway solution (Dirig Agent with FMX or FMXplus) and Dirig PathFinder will support WebSphere 5.0. As an IBM Alliance Partner, Dirig's support of WebSphere 5.0 is scheduled to ship later in the first quarter of 2003 and will provide enterprises with a solution to manage Web infrastructure and JMX-enabled applications, and provide top-level management for Web services projects.

"With Fenway running on WebSphere, Dirig enables companies to reap a higher return on investment, while maintaining an efficient e-business infrastructure,"said IBM's Scott Hebner, director of WebSphere marketing.

www.dirig.com

SAP NETWEAVER INTEGRATES WITH WEBSPHERE

(New York, NY) – SAP AG has announced the launch of the next evolutionary step of its industry-leading integration and application platform, designed to provide extensibility across heterogeneous IT landscapes. SAP NetWeaver enables organizations to integrate people, information, and business processes across technologies and organizations. It is fully interoperable with Microsoft .NET and IBM WebSphere, providing customers with flexibility to manage heterogeneous infrastructures, minimizing complexity, and reducing total cost of ownership.

SAP was one of the first companies to deliver a technology platform to enable collaborative business when it introduced its mySAP Technology in 2001. The advanced iteration of SAP's proven technology stack, SAP NetWeaver is the backbone for the SAP suite of solutions, delivering a complete, open, and flexible infrastructure that allows companies to realize additional value from existing IT investments.

www.sap.com

Petroleum Geo-Services to Transform Business with e-Business on Demand

(Armonk, NY) – IBM has introduced a new option for companies that want supercomputing power but not the fixed costs and operational responsibility of owning a supercomputer. IBM customers now have the choice to either buy POWER or Intel processor–based supercomputer clusters or to access these systems on demand, paying for processing power based on the required capacity and duration of use.

Certain industry sectors, including petroleum, digital media, and life sciences, require the power of supercomputers, but only at certain times in their product development cycles. At other times the servers they own sit idle, or under-utilized. For example, in Hollywood, studios need massive supercomputing power to render animation used in movies. Once the film is completed, the need goes away. The new IBM e-business on demand service is designed to provide customers with the latest technology to solve massive computational problems, and they pay only for the computing power and capacity they need.

The first company to access supercomputing power from IBM on demand is PGS Data Processing, a division of Petroleum Geo-Services, for an advanced seismic imaging project in the deep waters of the Gulf of Mexico.

"Seismic imaging services employ the latest numerically intensive applications, but are also highly cost competitive. PGS has been looking for a more flexible business model which addresses peak computing requirements, assures rapid response to our customers, but minimizes long-term, incremental cost commitments to PGS," said Chris Usher, president of Global Data Processing.

www.ibm.com

Mastering the team approach to application management

Ending the Blame Game

BY MIKE MALLOY

When there are problems with a mission-critical application, playing the blame game can stall progress and destroy your team's morale. How do you put an end to chaos and get your team back on track?

The following is an actual incident related by one of my company's consultants. A nervous young man greeted the consultant in the lobby on the IT floor. He simply said, "Follow me." As he walked down the hall to the application team room, the sounds of a heated discussion echoed in the corridor and grew louder.

The large meeting room was divided into "camps." Employees and contractors working for the systems integration firm were gathered in one corner around tables with laptops and printouts. The IT operations team was in another corner. Other camps included database management and mainframe administrators. In the center of the room a man and a woman were squared off facing one another. "Your architecture won't work, that's obvious! This performance is completely unacceptable! How can we can do so well in QA, but as soon as we go live we choke?" said the woman. "It's not the architecture," retorted the other, his voice rising in volume. "We've gone over this a dozen times. It's got to be either the way you have configured the app servers or your mainframe application."

The consultant's host pulled him aside. "It's been like this for two months. This is destroying our team's reputation and morale, and some execs are talking about killing the whole project. Can you do something about it?"

The visitor said to his host, "You are playing the blame game. The first step is to stop the chaos."

Twenty-four hours later, the team room had been transformed. Members from each team were huddled around a projected image on the wall. "Look at this metric," one was saying. "Database response time goes off the chart when users enter their lookup using a social security number. Remember the one we found yesterday, with the security server? This is similar. Okay, Bob, have your DBAs take a look at your lookup/join. Here's something else. On the WebSphere monitoring dashboard, watch what happens when that lookup takes too long." The project leader smiled, clearly satisfied with this newfound visibility into the whole Java application environment.

What had transformed the group from angry camps into a focused, high-performing team in such a short time? There were no group hugs, no team-building outings, no singing "Kumbaya" around the fire. There were two primary reasons. First, the visitor was able to recognize the symptoms of the blame game. Second, he was equipped with the software to give the deep visibility needed to bring the game to an end. Through no fault of their own your staff may also be playing this game. This article is about the blame game – what it is, how it starts, how to recognize it, and how to stop it.

ABOUT THE AUTHOR

Mike Malloy is a vice president with Wily Technology, a developer of enterprise Java application management software – solutions that put an end to the blame game.

E-MAIL

mmalloy@wilytech.com

What Is the Blame Game?

Mission-critical applications need to be managed to achieve their business objectives – revenue generation, customer service, and so forth. Application performance is measured on three factors – availability, performance, and control. Get high marks on these three factors and your application – and your application team – will succeed.

But the blame game can stall your progress and devastate your team. Why? Java applications are unlike monolithic legacy applications. A Java application is constructed more like a three-dimensional mosaic (see Figure 1). The application stack can include an operating system; a Java Virtual Machine (JVM); WebSphere; connectors to databases, mainframes, or other connected applications; and finally the application components themselves – servlets, EJBs, etc.

Due to the complexity of the application, management becomes a team activity. If a team-based problem resolution process does not exist or if it does not include enabling tools shared across the whole team, your team will descend into an endless round of fruitless experiments to find the root causes of the problems. This is the blame game.

The blame game, therefore, is defined as the uninformed finger-pointing that occurs among application team members when an application fails to achieve its goals. The game can result in loss of revenue, damaged brand reputation, damaged IT credibility, and even lawsuits and terminations.

How Does the Blame Game Start?

The blame game starts when a lack of hard data about production application performance forces each functional team member to defend his or her portion of the application under the philosophy that "If you can't prove it's my fault, it isn't." In other words, "Don't blame me!"

Consider all the people in your organization needed to develop and manage production Java applications – developers, architects, operators, QA engineers, DBAs, WebSphere administrators, CICS or MQ administrators, outside partners and suppliers. Typically, when a problem arises, the help desk notes it. If it's simple, operators fix it, restart the application, and move on. However, the complexity of Java applications causes most problems to be escalated to someone with an understanding of Java, of the architecture, of WebSphere, etc. At Wily we call this individual the Level 2 application support manager (ASM.) It's the ASM who performs problem triage and directs the problem to the right specialist for correction (see Figure 2).

If your organization has not nominated an ASM or if the ASM is not equipped with the tools needed to perform rapid triage, it's likely you are playing the blame game.

How to Tell If You Are Playing the Blame Game

1. Cross-department finger-pointing is rampant. The project leader is caught up mediating battles between groups.
2. Your best people – architects, senior engineers, and directors – are taking help desk calls. If the blame game goes on too long, they may consider leaving for calmer waters and more interesting work.
3. Despite your investment in specialized coding, profiling, and testing tools, application quality is not improving. Development tools cannot be deployed in a production environment, where problems related to actual customer transactions first appear.
4. Operations management is hiring Java programmers because their own monitoring tools are inadequate to isolate the root causes of problems.
5. Development is hiring operators to monitor applications. They figure that since they are getting all the calls for poorly performing production applications anyway, they may as well do their own monitoring.
6. Every new application throws the team into chaos. Just when you thought it was safe to launch an application, all the problems start again. If the blame game is not stopped permanently, it will recur with each new project.

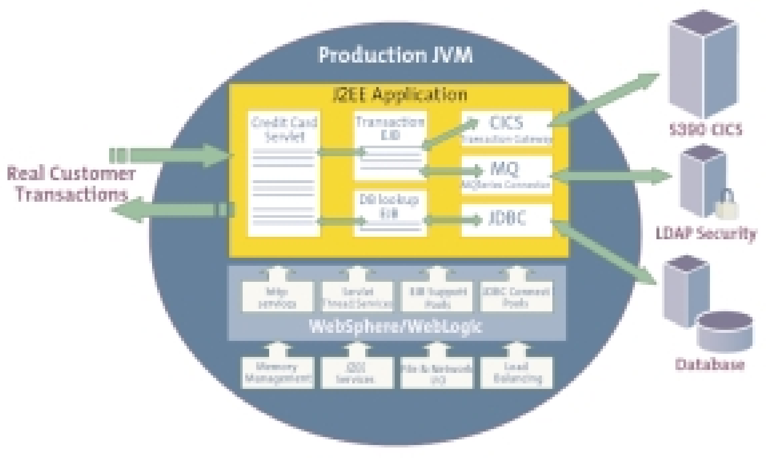


FIG. 1: JAVA APPLICATIONS ARE MULTILAYERED AND COMPLEX

Advertiser Index...

COMPANY	URL	PHONE	PG
BOWSTREET	WWW.BOWSTREET.COM/EASY	603-559-1900	5
DIRIG SOFTWARE	WWW.DIRIG.COM/ILLUSION/WEBSPHERE	603-889-2777	21
KENETIKS	WWW.KENETIKS.COM	888-KENETIKS	13
LINUX BUSINESS & TECHNOLOGY	WWW.SYS-CON.COM/LINUX	888-303-5282	35
MQ SOFTWARE	WWW.MQSOFTWARE.COM/GUIDE	952-345-8720	6,7
PERFORMANT	WWW.PERFORMANT.COM/WEBSPHERE1	866-773-6268 x0	23
PRECISE	WWW.PRECISE.COM	800-310-4777	17
PROLIFICS	WWW.PROLIFICS.COM	800-675-5419	15
SITRAKA (now Part of Quest Software)	WWW.SITRAKA.COM/JCLASS/WDJ	800-663-4723	51
SITRAKA (now Part of Quest Software)	WWW.SITRAKA.COM/PERFORMANCE/WDJ	800-663-4723	11
TRILOG GROUP	WWW.FLOWBUILDER.COM/XSP	800-818-2199	52
VERSATA	WWW.VERSATA.COM/BUSINESSLOGICDESIGNER	800-984-7638	2
WEB SERVICES EDGE EAST	WWW.SYS-CON.COM/WEBSERVICESEDGE	201-802-3058	39
WILY TECHNOLOGY	WWW.WILYTECH.COM	888-GET-WILY	3

Coming Next Month...

PERFORMANCE:

Practicing Software Performance Engineering

BY WILFRED JAMISON

WEBSPHERE 5.0:

Understanding EJB Entity Bean Caching Options and How to Use Them

BY HARVEY GUNTHER

PRODUCT REVIEW:

Together WebSphere Studio Edition from TogetherSoft

BY JAY JOHNSON

PRODUCT REVIEW:

Macromedia ColdFusion MX for WebSphere from Macromedia

BY ANTONIO VILLAFANA

If you've recognized any of these symptoms, your team is probably playing the blame game. You need to step in and put an end to it.

How to Stop the Blame Game

Again, IT teams fall into the blame game through no fault of their own. It's not because they are stubborn, stupid, or insensitive. The only way to end the game is to follow the steps outlined below.

1. ***Stop the Chaos:*** The chaos can only be stopped by better information about the whole application – the Java components, the JVM, WebSphere, and the connections to databases and mainframes – which can only come from better application management tools. You must adopt tools that enable your ASM and the entire application support team to see inside a running production Java application and monitor its performance 24/7.

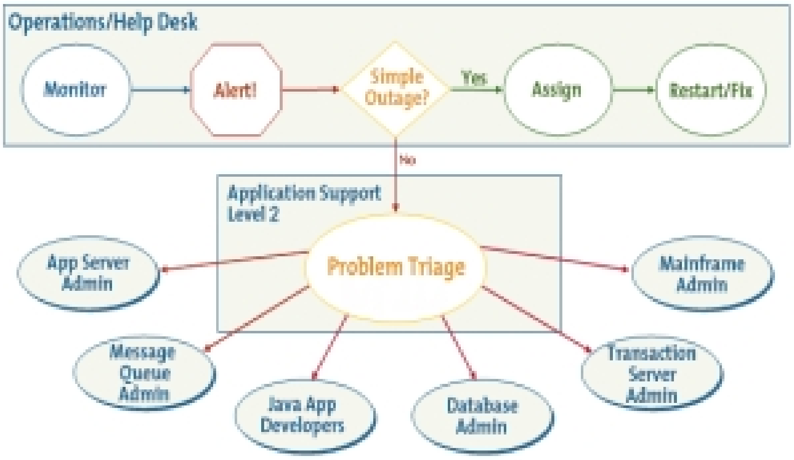
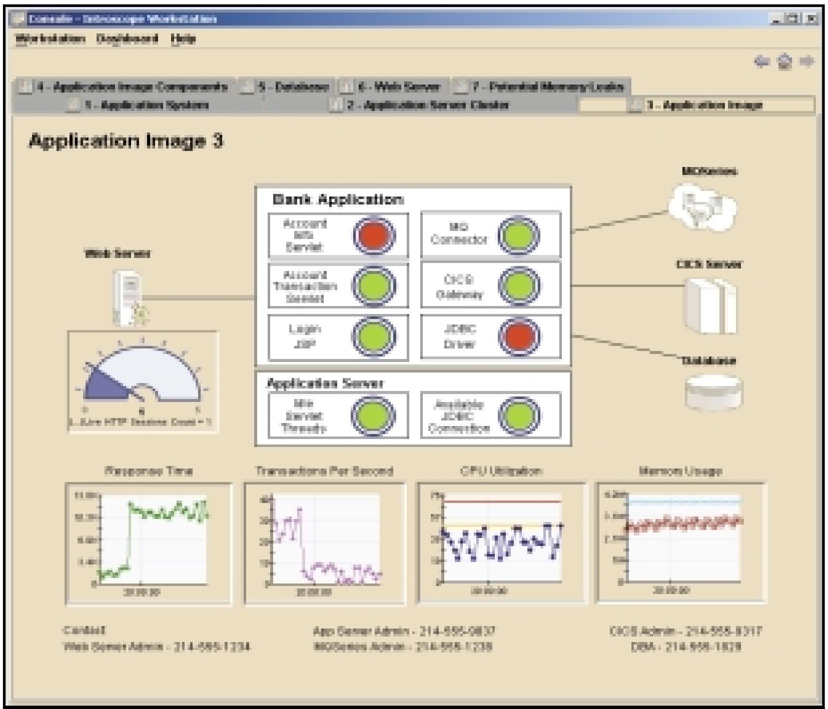


FIG. 2: THE APPLICATION SUPPORT MANAGER PERFORMS TRIAGE ON PROBLEMS



Contact
Web Server Admin - 214-555-1234
App Server Admin - 214-555-0837
MQSeries Admin - 214-555-1238
CICS Admin - 214-555-8317
DBA - 214-555-1828

FIG. 3: A DASHBOARD CONFIGURED TO MONITOR A PRODUCTION APPLICATION SHOWS DETAILED PERFORMANCE METRICS FOR CRITICAL COMPONENTS

When problems occur, the application management solution you use must be able to provide everyone on the team – developers, architects, DBAs, etc.– with the data they need. The solution needs to show real-time performance of components and how they interact with one another. Problem ownership will then be obvious, not a subject for debate. The chaos will become control.

2. ***Bridge the Gap Between the Teams:*** Once the chaos has been stopped and your team is at a point of relative stability and productivity, it is time to set the wheels in motion to permanently end the blame game.

One of the primary causes of the blame game is the breakdown in the key relationships between development and operations. Your team needs a common language and a shared view into the whole application. Your team members need a tool they all can use.

If you have not created a cross-functional application support team, this is the time. Establish the role of application support manager. Allow this team to monitor, improve, and manage applications to achieve service-level agreement targets for which they can be held accountable.

3. ***Deploy to Avoid:*** Ultimately your application support team must develop processes and disciplines before deployment that enforce standards for application quality. Have the architects and developers build the production-monitoring dashboards while the application is being tested (see Figure 3). No one knows better what is most critical to an application's successful performance than the team that built it.

Finally, use your application management system to establish benchmarks for application quality. Put teeth into your development-staging-deployment process by prescreening applications and certifying them for memory usage, expected service levels, runtime load, external dependencies, etc.

For many readers, just achieving the first step – stopping the chaos – may represent a victory. However best-in-class IT organizations will not stop there. They know that lurking just around the corner, waiting for their unsuspecting employees, lies the blame game.