WebSphere Portal, Portlets and Web Services

June 2002

Peter Fischer
Developer, WebSphere Portal Server

IBM Software Group

Portal Architecture
Introduction

IBM Software Group
What are Portals?

- **Common access point to distributed information and applications**

- **Typical Functions:**
  - User registration
  - Authentication and Authorization
  - Pluggable portal components: Portlets
  - Personalization based on profiles/behavior
  - Customization of pages by users
  - Search
  - Content Management

Example of a Portal View
WebSphere Portal Quick Overview

- Multi-Platform: Windows, AIX, Solaris
- Multi-Client: PCs, WAP, iMode
- Multi-Language: Serves different locales concurrently
- Multi-Authentication: Supports most relevant auth proxies
- Provides well-defined Portlet API and portlet auto-deployment
- Many Portlets on the IBM Portlet Marketplace (200+)

- Portlets can exploit WebSphere web service functions
- Supports pluggable, interactive, user-facing web services
- Can publish portlets as web services using admin UI
WebSphere Portal Server Architecture

WebSphere Portal Server Engine

- WebSphere Portlet Services
- WebSphere Portal Server Data Store
- WebSphere Member Subsystem
- WebSphere J2EE/WebSphere APIs
- J2EE/WebSphere APIs
- Corporate UDDI Directory
- Corporate RPWS Services
- Public RPWS Services
- Public Web Services
- Secure Way LDAP, Domino LDAP, Netscape LDAP, Active Directory, or WMS DB
- DB2 or Oracle

Portlets and Portlet API

IBM Software Group
Portlets

- Components designed to be aggregated in portals
- Aware of portal context
  - User profile information
  - Per-portlet instance data stored by portal
  - Per-portlet settings managed by portal
  - Portlet window state (NORMAL, MIN, MAX)
  - Portlet modes (VIEW, EDIT, CONF, HELP)
  - Portlet events (action / message events)
- Can be packaged in normal WAR Files with descriptor extensions

Portlet Action Handling and Aggregation

Portlet Container

Portal Infrastructure, User Registry
Persistent and Transient Portal State
Example of a Stock Quote Portlet

Stock prices for user-selected list of stock symbols:
- VIEW mode shows stock prices: doView method
- EDIT mode lets user change stocks: doEdit method
- HELP mode explains the portlet: doHelp method
- CONFIG mode lets administrator select stock quote source to use: doConfig method

Example of a Portlet

- Portlets are specialized Servlets

```java
public class StocksPortlet extends Portlet {
    public void init(PortletConfig config) {
        ... initialize portlet ...
    }

    public void doView(PortletRequest req,
                        PortletResponse rsp)
    throws PortletException, IOException {
        ... generate the portlet view ...
    }

    public void destroy(PortletConfig config) {
        ... destroy portlet ...
    }
}
```
Example of Portlet View Mode

- Use of PortletData, Beans and JSP™ components

```java
public void doView(PortletRequest req,
                   PortletResponse rsp)
    throws PortletException, IOException {
    // Get stock symbols from portlet instance data
    PortletData data = req.getData();
    String symb = (String) data.getAttribute("symbols");
    // get prices for symbols from stocks service ..
    StockBean stockBean = new StockBean();
    // put stock symbol/price pairs in stock bean ..
    req.setAttribute("stockBean", stockBean);
    getPortletConfig().getContext().include(
        "/WEB-INF/ViewStockQuotes.jsp", req, rsp);
}
```

Example of Portlet Edit Mode

- Tie Actions to PortletURLs to process Forms

```java
public void doEdit(PortletRequest req,
                   PortletResponse rsp)
    throws PortletException, IOException {
    // Create URI pointing to this portlet instance
    // and attach a portlet action
    PortletURI saveUri = rsp.createURI();
    PortletAction saveAction = new Action(SAVE);
    saveUri.addAction(saveAction);
    EditListBean editListBean = new EditListBean();
    // put saveURI and other data into the bean ..
    req.setAttribute("editListBean", editListBean);
    getPortletConfig().getContext().include(
        "/WEB-INF/EditSymbolListForm.jsp", req, rsp);
}
```
More Information about Portlet Development

- see Portlet Development Guide available at the portal library

Java Portlet API (JSR 168)

- Defines interaction between portals and portlets
- Defines interfaces for interoperability of portals and portlets
- Based on the Java™ Servlet API
- Provides additional abstractions for portal context
  - user object, persistent data objects, device information, portlet modes, window states, ...
- Standardization in JSR 168 lead by IBM and Sun
  (see http://jcp.org/jsr/detail/168.jsp)
- A Portlet API reference implementation will be donated to Apache Open Source Community
Java Portlet API Supporters

- Accenture
- Apache
- ATG
- BEA
- Boeing
- Borland
- Bowstreet
- Cap Gemini Ernst & Young
- Citrix
- DaimlerChrysler
- Documentum
- Enformia Ltd
- Epicentric
- Fujitsu
- Hewlett-Packard
- Hitachi
- IBM (Spec Lead)
- Interwoven
- Macromedia
- McDonal Bradley
- Plumtree
- SAP Portals
- Silverstream
- Sybase
- Tarantella, Inc
- Vignette
- IONA
- Sun (Spec Lead)
- Computer Associates
- Peoplesoft
- SAS
- Oracle

Portal WebServices

Remote Portlets

IBM Software Group
Web Services

- Web services are platform and language independent
- Description of Web services in WSDL (Web Services Description Language)
- Invocation of Web services via SOAP (Simple Object Access Protocol)
- Publish & Find through UDDI (Universal Description, Discovery & Integration)

Service Oriented Architecture – Publish, Find & Bind
"Traditional" Web Service Usage Scenario

- Portlets using data-oriented Web services
  - Different data-oriented Web services expose different interfaces
  - Specialized UI and proxy code required in specific portlets
  - Local deployment of code is still necessary

Remote Portlets Web Services

- Generic Proxies using user-facing web services
  - All RPWS services have a common API
  - No service specific portlets required
  - Generic RPWS proxy portlet is implemented once and used for all RPWS services
Publishing Portlets as RPWS Services

Remote Portlet Web Services Goals

- Allow **interactive**, user-facing web services to be easily plugged into all standards-compliant portals
- Let anybody **create and publish** their content and applications as user-facing web services
- Portal administrators browse public or private UDDI directories for RPWS services to plug into their portals as new portlets, **without any programming effort**
- Let portals publish portlets so that they can be consumed by other portals
- Make the internet a **market** of visual web services, waiting to be integrated
Business Scenario Examples

To plug into portals ...

- ... Content Providers publish content as RPWS services
  (e.g. Stock Quotes, News, Lottery Numbers, Sports Results, Flight Schedules, …)

- ... Application providers expose apps as RPWS services
  (e.g. Stock Option Programs, E-Mail, Calendar, CRM, Workflow, Tax Calculation, Ticket Ordering, Travel Booking,...)

- ... Portal providers publish local portlets as RPWS services to share them with other portals
  (e.g. a content provider’s portal might publish its portlets for re-use in employee portals of different companies)

Web Services for Remote Portals (WSRP)

- Standardization taking place in OASIS WSRP TC
  Chairman: Thomas Schaeck

- WSRP services are user-facing, interactive web services that may be aware of portal-side user profile information, devices, locales

- WSRP will standardize:
  - How to publish, find, and bind to WSRP services
  - Metainformation for WSRP services (name, supported locales/markups, titles, and descriptions, …)
  - Protocol for interaction between portals and WSRP services

- WSRP Home Page: http://oasis-open.org/committees/wsrp/

- Goal:
  - WSRP 1.0 Spec and Implementation year end 2002
Companies who participate in WSRP TC

- BEA
- Bowstreet
- Divine
- Epicentric
- Factiva
- France Telecom
- Fujitsu
- HP
- IBM
- Interwoven
- Lexis-Nexis
- Lotus
- Moravia IT
- Netegrity
- Oracle
- Peoplesoft
- Plumtree
- Silverstream
- Stellent
- Sun
- Sybase
- Tibco
- WebCollage
- SAP Portals
- SeeBeyond

How WSRP and Java Portlet API (JSR 168) fit

- Portlet API defines Java API for local portlets
- WSRP defines user-facing, interactive web services that plug & play with portals

Goals:
- Allow Java portlets to be wrapped and published to UDDI as WSRP services
- Allow WSRP services to be integrated in portals by using generic portlet proxies
Portal consuming .NET-based RPWS Services

Portal Servers -> RPWS -> IIS -> .NET object

Microsoft Office as OLE server

RPWS service inside a Word Document

.NET object -> WSRP -> Portal Servers
Thank you!

IBM Software Group