Application Servers
G22.3033-011

Session 9 - Main Theme
CORBA 3 Component-Based Computing Environments

Dr. Jean-Claude Franchitti

New York University
Computer Science Department
Courant Institute of Mathematical Sciences

Agenda

- J2EE Client Component Infrastructures (continued)
- CORBA 3 OMA Services
- CORBA 3 CCM
- CORBA 3 Environments
- Enterprise Component Development with CORBA 3
- Designing CORBA 3 Component-Based Architectures
- Summary
- Readings
- Assignment #7
Summary of Previous Session

- Previous Session Continued
  - OMA Services Detailed
  - BEA WebLogic 8.1, IBM WebSphere 5.0, JBoss.org JBoss
- Other Open Source J2EE Environments
  - Enhydra.org/Lutris.com Enhydra
  - ExoLab.org OpenEJB
- Other Commercial J2EE Application Servers
- Summary
- Readings
- Assignment #6

Application Servers Architectures

- Application Servers for Enhanced HTML (traditional)
  - a.k.a., Page-Based Application Servers
  - Mostly Used to Support Standalone Web Applications
- New Generation Page-Based Script-Oriented App. Servers
  - First Generation Extensions (e.g., Microsoft IIS with COM+/ASP)
  - Servlet/JSP Environments
  - XSP Environment
  - Can now be used as front-end to enterprise applications
  - Hybrid development environments
- Distributed Object Computing Platforms
  - Provide an infrastructure for distributed communications enabling
  - Still need to merge traditional web-oriented computing with object computing
- Object Management Architectures
  - DOC Platform + APIs to reusable services and facilities
  - OMA + Component Models -> J2EE, CCM, DNA
Part I

J2EE Client
Component Infrastructures

Applet-Based and JWS Clients

- Conventional Client Technology Support (Thick/Thin)
  - Applets, Servlets, and Java Server Pages
- Javasoft’s Applet Tutorial:
- Swing Applets:
- Java Web Start:
Java Web Start (JWS) Technology

- See http://java.sun.com/products/javawebstart/
- Applications do not require a browser once downloaded
- Files are cached on the client
- Checks are performed with the web server to ensure that the latest version of the application is being used (ZAC!)
- Built on top of Java 2 Security
- Links point to JNLP files which syntax is defined as part of a Java Community Process specification
  - http://java.sun.com/aboutJava/communityprocess/jsr_056_jnlp.html

JWS Application Launching
Sample JNLP File

```xml
<?xml version="1.0" encoding="utf-8"?>
<jnlp spec="1.0+" codebase="http://javaweb.eng.com/jaws/apps" href="swingset2.jnlp">
  <information>
    <title>SwingSet2 Demo Application</title>
    <vendor>Sun Microsystems, Inc.</vendor>
    <homepage href="docs/help.html"/>
    <description>SwingSet2 Demo App</description>
    <description kind="short">A demo</description>
    <icon href="images/swingset2.jpg"/>
    <offline-allowed/>
  </information>
  <security/>
  <resources>
    <j2se version="1.3"/>
    <jar href="lib/SwingSet2.jar"/>
  </resources>
  <application-desc main-class="SwingSet2"/>
</jnlp>
```

JNLP Element to Launch an Applet

```xml
<applet-desc
documentBase="http://..."
name="TimePilot"
main-class="TimePilot.TimePilotApp"
width="527"
height="428">
  <param name="key1" value="value1"/>
  <param name="key2" value="value2"/>
</applet-desc>
```
JNLPI API

- BasicService
- ClipboardService
- DownloadService
- FileOpenService
- FileSaveService
- PrintService
- PersistenceService

AWT & Swing Components

http://java.sun.com/j2se/1.4/docs/guide/swing/index.html

- Swing is part of the Java Foundation Classes (JFC)
- Swing implements GUI components with “PLAF”
  - PLAF: “Pluggable Look and Feel”
  - GUI components automatically adapt their look and feel to that of the OS platform
- Swing includes the AWT component
  - Button, Scrollbar, Label, etc.
- Swing also includes higher level components
  - TreeView, list box, tabbed panes, etc.
Swing vs. SWT

- Eclipse Open Platform for Tool Integration
  - 100% implemented in Java
  - Support extensions based on “plug-ins”
  - Does not use a single line of AWT or Swing!
  - Used the Standard Widget Toolkit (SWT) and JFace for more complex widgets, images, and font registries
- SWT uses native widgets except when a class of widget is not commonly available on all target platforms.
  - In that case, SWT emulates the widget

Swing vs. SWT

(continued)

- SWT Advantages
  - GUIs look and feel exactly like native GUIs
  - Graphics performance is close to native
  - Platform-specific features can be accessed if desired
    - E.g., embedding ActiveX controls, and using native DnD
- SWT Drawbacks
  - No support for pluggable look and feels
  - Door open to write non-portable code
- See STW/Swing comparison sample program at:
SwingML
(http://swingml.sourceforge.net/)

- Markup language to render JFC/Swing based GUIs in a Web browser
- Alternatives?
  - Use JSP, JavaBeans, XSLT, and Servlets to generate dynamic HTML
    - HTML cannot support complex functionality such as a Tree component with D&D
  - Use JavaScript and Applets
    - JavaScript support is not uniform across browsers
  - Applets combined with JFC/Swing components
    - Difficult to retrieve information from server: use RMI, HTTP Tunneling, or … XML!

SwingML Architecture

- Server-side component produces SwingML and returns it to the client side
- On the client side, an applet (“renderer”) receives the SwingML markup text and uses it to render the GUI
XML-based User Interface Language (XUL)  
(http://www.mozilla.org/xpfe/xptoolkit/xulintro.html)

```xml
<?xml version="1.0"?>
<?xml-stylesheet href="chrome://global/skin/xul.css" type="text/css"?>
<!DOCTYPE window>
<window id="main-window" xmlns:html="http://www.w3.org/1999/xhtml" xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">
  <menubar>
    <menu name="File">
      <menuitem name="Hello World!" onclick="dump('Hello world!');"/>
    </menu>
  </menubar>
  <html:iframe id="content-frame" src="contentframe.html" flex="100%"/>
</window>
```

Other Rich, Cross-Platform, ZAC GUI Solutions

- Known Variations
  - (1) Browser-plugins (a la Applets, SVG) - Flash
  - (2) Language (a la Java, JavaScript) - Curl, Rebol
  - (3) Improved HTML (a la XUL) - NexaWeb, SashXB
  - (4) Improved installers (a la Web Start) - AppStream

- Luxor XUL (http://luxor-xul.sourceforge.net)
- XML Windowing Toolkit - XWT (http://www.xwt.org)
- JellySwing (http://jakarta.apache.org/commons/sandbox/jelly/jellyswing.html)
- Thinlets (http://www.thinlet.com/)
- KoalaGM (http://koalagml.sourceforge.net/)
- Java GUI Builder (http://jgb.sourceforge.net/)
- Not Yet Xulux (Nyx), Motlib.Net, Rocklyte Systems, Altio, Bali Spidertop, droplets, Ultra Light Client (ULC), Kenamea, Softricity, etc.
Light Client GUI Solutions Selection Criteria

- Must be able to support alternative GUI toolkits
  - Sun’s Swing, Eclipse’s SWT, Trolltech Qt, GnomeGtk+, etc.
- Must be able to support various programming language engine
  - Java, C#, Shark, etc.

Part II

CORBA 3 OMA Services Detailed

Also See:

and
CORBA OMA Services

- Activation Services
  - CORBA POA
  - CORBA Lifecycle
- Naming Service
  - CosNaming
- Directory and Trading Services
  - CosTrading
- Object Transaction Service (OTS)*
- Messaging Services*
  - CORBA Event Service
  - CORBA Notification Service
- Security Service*

Transaction Service at Work
CORBA OTS and Messaging Services

- CORBA Transaction Service
  - Available as an OMG specification
  - See slides http://www.cs.wustl.edu/~schmidt/CORBA-docs/
  - Implementations: Orbacus OTS, Encina++, OrbixOTM, etc.
  - Specification used to derive JTS described in part II of this slide set
    (implements Java mapping of the CORBA OTS 1.1)

- CORBA Messaging Service
  - Based on CosEvents and CosNotification Services
  - CORBA 3 adds Filters, Quality of Service, and Structured Events
    - CORBA 3 adds Filters, Quality of Service, and Structured events
    - Structured events promote lightweight notification
  - Implementations: TAO 1.1
  - Contenders: IBM MQSeries, Microsoft Message Queue Server, TIBCO
    ETX, and PeerLogic Inc. PIPES

CORBA 3 Structured Events

![CORBA 3 Structured Events Diagram]
CORBA services - Security

CORBA Security Reference Model

Client

Object Implementation

ORB(s)

Request

Client-side and target-side security on invocations, security association, access control, message protection and audit.
**TAO’s SSLIOP (IIOP over SSL)**

**Pluggable Protocol**

- Protocol must be loaded
  
  ```c
  dynamic SSLIOP_Factory Service_Object *
  
  TAO_SSLIOP:_make_TAO_SSLIOP_Protocol_Factory() ""
      static Resource_Factory "- ORBProtocolFactory"SSLIOP.Factory"
  
  SSLIOP enforces integrity, confidentiality, and secure invocation for client requests
- X.509 certificate-based request authorization can also be implemented programmatically via SSLIOP
CORBA Web-Enabling
(Second Generation Client/Server Architectures)

- Need Domain Boundary Controllers (e.g., www.xtradyne.com) until HTTP-NG, or CORBA 3 firewall specification get implemented

CORBA Web-Enabling
(Gatekeeper, Wonderwall, etc.)

- Application Servers include IIOP gateway products
- CORBA 3 includes a firewall spec. for IIOP support
Part III

**CORBA 3’s CORBA Components Container (CCM) Environment**

*See Sub-Topic 1 Presentation on “CORBA 3”*

*Also See John Siegel’s  
  *CORBA 3 Fundamentals and Programming*  
  and  
  *Quick CORBA 3*

---

**CCM v.s. EJB**

- EJBs are reusable into the CCM platform
- CCM supports component assembly
- CCM is programming language independent and tailored towards legacy application connectivity
- EJBs are not components
- EJB servers provide an object platform and manage containers that support bean deployment with an abstraction to the developer concerning persistence, transaction, and security management
### CCM Component Development

- **Component Declaration**
  - IDL 3 compiler generates stubs and metadata for components

- **Component Implementation**
  - Component implementations (structure and state) and homes are declared using CIDL (declarative language)
  - Developer provides implementation business logic

- **Component Packaging**
  - Archive that contains implementations, deployment info, etc.

- **Component Assembly**
  - Used to deploy a set of components that may need to be linked

- **Component Deployment and Installation**
  - System generates all connections for component assemblies

- **Component Instance Activation**
  - Retrieve Home and/or Component reference to invoke business methods

### Part IV

**CORBA 3 Environments**
CORBA 3 Environments

- See:
  - http://ditec.um.es/~dsevilla/ccm/

- Implementations (mostly open source):
  - GOAL Group OpenCCM - http://corbaweb.lifl.fr/OpenCCM/
  - ExoLab.org OpenCCM - http://openccm.exolab.org/
  - iCMG K2-CCM (C++) - K2-CCM
  - MICO/E (Eiffel ORB) - MicoCCM page
  - JavaCCM - http://dog.team.free.fr/details_javaccm.html
  - Other companies: Eurescom/GMD/Humboldt U, Computational Physics/Photon Research, Sprint, ONE, Siemens, Sourceforge MI-3 (“Mission Impossible 3”) and CIF projects (http://sourceforge.net/projects/cif/), etc.

Generic CORBA 3 Platform
K2-CCM Architectural Components

Part V

Conclusion
Summary

- CORBA OTS is implemented in several CORBA products and provides the basic standard on which J2EE JTS implementations are based.
- CORBA Messaging Event and Notification services are still being developed to allow for efficient filtering of messages currently unsupported in mainstream solutions.
- Interoperable secure CORBA implementations are complex to implement, and mainstream products only provide partial security solutions.
- CORBA 3’s added-value is in the area of Internet Integration, Quality of Service and CCM support.
- Only a few open source and commercial implementation of CORBA 3 are available today.

Readings

- Readings
  - Handouts posted on the course web site
  - Explore CORBA 3/CCM environments
  - Read related white papers/documentation on the CORBA 3/CCM environments
Project Frameworks

- Project Frameworks Setup (ongoing)
  - Apache Web Server (version 1.3.28/2.0.47, www.apache.org)
  - Perl (version 5.8.0, www.perl.com)
  - Microsoft IIS with COM+/.Net and ASP
  - Sun One Active Server Pages 4.0 http://wwws.sun.com/software/chilisoft/index.html
  - Apache Tomcat
  - Macromedia JRun4
  - Apache Cocoon 2/XSP
  - Visibroker, Orbacus
  - RMI-IIOP
  - WebLogic 8.1, WebSphere 5.0, JBoss
  - Inprise AppServer, Sun ONE, Sybase EAServer, Oracle 9i, IONA iPortal, Xoology
    Concerto, Aligo M-1, Advanced Network Systems Web1x
  - GOAL Group OpenCCM, ExoLab.org OpenCCM, iCMG K2-CCM (C++), MICO/E
    (Eiffel ORB), JavaCCM, TAO Group, IONA iPortal (no CCM), Borland AppServer
    (no CCM), Sourceforge MI-3 (“Mission Impossible 3”) and CIF projects

Structured Applications Design Tips

- Reuse: should focus on Domain Models/System Family Architectures
- Applications should separate the various information elements (i.e., content, logic, style, and architecture/handling schemes)
- Various content formats: presentation, message, storage, etc.
- Application architecture supports:
  - Web Enabling (WE), XML Enabling (XE), Data Enabling (DE), Enterprise System Assurance Enabling (ESAE)
- Various application support services to support:
  - Interactions with users via content (content + logic) - WE
  - Encoding of user requests as secure (portable) messages (content generation) - XE/ESAE
  - Processing of user requests via logic (content + logic) - XE
  - Rendering of content via logic using style (content + style + logic) - WE/XE
  - Querying information via logic (content + logic) - XE/DE
  - Interactions with back office via content (content + logic) - XE/ESAE
Assignment

- Explore the textbooks’ references to CORBA 3/CCM Application Server technology
- Homework #6 due date is 11/12/03
- Homework #7a: Investigate CORBA 3 CCM computing platforms’ development environments. Write a short report that documents your findings and recommendations with respect to selection criteria in support of development environments for application server technologies covered in this session
- Homework #7b: See homework #7 specification (due date is 11/19/03)

Next Session:

COM+ Component-Based Computing Environments

- COM/DCOM Component Model and OLE
- COM+ MTS Services: Transactions and Security
- COM+ Message Queuing
- .Net/COM+ Integrated Services
- Enterprise Component Development with .Net/COM+
- Designing .Net/COM+ Component-Based Architectures