Java Web-based Programming & J2EE Architecture

Peter.Cheng
Email: founder_chen@yahoo.com.cn
http://www.huihoo.org
Course Goal

- The main goal of this course is to provide you with the knowledge for Java web-based programming (JSP, Servlet, Struts), MVC Design pattern, J2EE architectures
Course Overview

This course covers the following areas:

- J2EE Technology overview
- Web-based Programming Model
- MVC Design Pattern and Struts Framework
- J2EE Architectures
- Build a J2EE developer Team
How To Use Course Materials

- **Lecture** The instructor will present information specific to the topic of the module. This information will help you learn the knowledge and skills necessary to succeed with the exercises.

- **Exercise** Lab exercises will give you the opportunity to practice your skills and apply the concepts presented in the lecture.

- **Think Beyond** Thought-provoking questions are posed to help you apply the content of the module or predict the content in the next module.
Software Architecture

- Application
- Virtual Platform
- Upper Platform
- Lower Platform

- OS (Unix, Windows, Linux, Mac, others)
- J2EE Infrastructure (SunOne, Weblogic, Websphere, etc)
- APIs, specifications

Huihoo - Enterprise Open Source
http://www.huihoo.com
Architecture and the cube

- Client
- Presentation
- Business
- Integration
- Resource
- Security
- Manageability
- Availability
- Scalability

- Application
- Virtual Platform
- Upper Platform
- Lower Platform

- Client
- Presentation
- Business
- Integration
- Resource
- Security
- Manageability
- Availability
- Scalability

APIs, specifications
Middleware (Web, app server)
OS (Solaris, Windows, Linux, etc.)

Huihoo - Enterprise Open Source
http://www.huihoo.com
Java Platform Overview

Java Programming Language

Java 2 ME APIs

Optional Package

Java 2 Standard Edition (J2SE) Core APIs

Optional Package

Java 2 Enterprise Edition (J2EE) Core APIs

Java Hotspot ™

JVM

KVM

Card VM

Fundamental Profile

Personal Profile

RMI Profile

Other CDC Profiles...

MIDP

Java Card APIs
Java is everywhere
J2EE 3-Tier Architecture

Client Side
- Presentation
  - Browser
    - Pure HTML
    - Applet
  - Desktop
    - Java Application
  - Mobile Device
    - J2ME

Server Side
- Presentation
  - Web Server
    - JSP
    - Java Servlet
  - J2EE Platform
- Business Logic
  - EJB Container
    - EJB
    - EJB
  - J2EE Platform

Enterprise Information System
- EIS
- Other System

Huihoo - Enterprise Open Source  http://www.huihoo.com
Computing Model is changing

<table>
<thead>
<tr>
<th>Model</th>
<th>Client/Server</th>
<th>Browser/Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>UI</td>
<td>Application</td>
<td>Browser (html)</td>
</tr>
<tr>
<td>Maintain</td>
<td>Hard</td>
<td>Easy</td>
</tr>
<tr>
<td>Couple</td>
<td>High</td>
<td>Lower</td>
</tr>
</tbody>
</table>
History of Web Applications

- **Common Gateway Interface**

  ![Diagram showing the interaction between requests and processes]

  - Request for CGI1
  - Request for CGI2
  - Request for CGI1

  - Child process for CGI1
  - Child process for CGI2
  - Child process for CGI1
History of Web Applications

- **Java Servlet**

![Diagram showing Java Servlet process]

- Request for Servlet1
- Request for Servlet2
- Request for Servlet1
Web Tier – HTTP Request/Response Model

• A Client – a web browser
• Send a request for a resource to a server
• Server sends back a response corresponding to the resource
The Servlet

@see HelloWorldServlet.java
http://localhost:8081/servlet/HelloWorldServlet
The Servlet – single instance

@see SingleInstance.java

http://localhost:8081/servlet/SingleInstance
The Servlet – Single Thread Model

Web Server

Servlet Pool

Servlet Instance
Servlet Instance
Servlet Instance
Servlet Instance

request
request
request
request
Web Tier - What’s JSP?

- JavaServer Pages, for short—is a Java-based technology that simplifies the process of developing dynamic web sites.

- A JSP page contains standard markup language elements, such as HTML tags, just like a regular web page.

- A JSP page also contains special JSP elements that allow the server to insert dynamic content in the page.
## HTML vs. JSP

<table>
<thead>
<tr>
<th>HTML</th>
<th>JSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;html&gt;</td>
<td>&lt;html&gt;</td>
</tr>
<tr>
<td>&lt;head&gt;</td>
<td>&lt;head&gt;</td>
</tr>
<tr>
<td>&lt;body&gt;</td>
<td>&lt;body&gt;</td>
</tr>
<tr>
<td>&lt;LI&gt;&lt;A&gt;Why Use JSP&lt;/A&gt;&lt;/LI&gt;</td>
<td><a href="">jsp:useBean…/</a></td>
</tr>
<tr>
<td>&lt;/body&gt;</td>
<td>&lt;/body&gt;</td>
</tr>
<tr>
<td>&lt;/html&gt;</td>
<td>&lt;/html&gt;</td>
</tr>
</tbody>
</table>
Why use JSP?

- Embedding Dynamic Elements in HTML Pages
- Compilation
  - CGI/Perl require the server to load an interpreter and the target script each time the page is requested.
  - a JSP page is always compiled before it's processed by the server.
- Integration with Enterprise Java APIs
  - JDBC, JMS, EJB, etc
Inside the web server
Web Programming Model - 1

Client Application

request

response

JSP (View)

JavaBeans (Model)

Data
Web Programming Model - 2

@see ControllerServlet.java welcome.jsp login.jsp
MVC (Model-View-Controller)

**Model**
- Encapsulates application state
- Responds to state queries
- Exposes application functionality
- Notifies views of changes

**View**
- Render the model
- Requests model update
- Send user input to controller

**Controller**
- Define application behavior
- Maps user actions to model update
- Select view for response
- One for each functionality

**Diagram:***
- Query model state
- Update state change
- Change notification
- User input
- View select
Basic MVC Control Flow

ControllerServlet

HandlerMapping

Controller

ViewResolver

View

:getHandler(request)

Handler: Controller

:handleRequest(request, response)

:resolveViewname(name, local)

:render (model, request, response)
Open Source MVC Framework

- Struts  www.apache.org
- WebWork  www.opensymphony.com
- Maverick  
  http://sourceforge.net/project/mav
Framework vs. Component

- Application 1 Extension
- Application 3 Extension
- Application 2 Extension
- Library 1
- Library 2
- Main Application

Framework
Functional Application Tier
The Struts framework is used within the web tier
The execute() method call by controller
What’s J2EE Technology

- **Components:**
  - Application clients
  - Applets
  - Web components
  - Business components

- **Containers**
  - Manage lifecycle of business components
  - Provide a federated view of J2EE APIs
  - Provide runtime support for components
J2EE Architectures
J2EE Application Model

Enterprise Information Systems (EIS):
- Relational-Database
- Legacy Applications
- ERP Systems

Other Services:
- JNDI, JMS
- Java Mail

Application Server
- EJB
- EJB

JSP/Servlet

Client

Firewall

Huihoo - Enterprise Open Source
http://www.huihoo.com
J2EE Platform Specification
Vertical Scalability

Any machine leads to a system failure

Capacity ↑ Availability ↓
Reliability ↓ Manageability ↑
Design J2EE-Based Application

- EJB is not the only technology for implementing the middle tier in J2EE applications

- Design to Java interfaces, not concrete classes, and not technologies
Non-distributed Architectures

Web Application with Business Component Interfaces

- UI Tier
- Middle Tier
- (Business Logic)
- Servlet/Web Tier classes
- Business interface
- Implementation
- EIS
- DB
- Legacy
- Web Container
- J2EE Application Server

Huihoo - Enterprise Open Source  http://www.huihoo.com
Web Application that Accesses Local EJBs

UI Tier

Servlet/Web Tier classes

Business Interface

Business Delegate

Middle Tier

Local EJB Invocation

Session Bean

Entity Bean (Optional)

EIS

DB

Legacy

Web Container

EJB Container

J2EE Application Server (Single JVM)
Distributed Architectures

UI Tier

Servlet/Web Tier classes

Business Interface

Business Delegate

Middle Tier

RMI Remote EJB Invocation

Session Bean

Entity Bean (Optional)

(Ebusiness Logic)

J2EE App Server

J2EE App Server

EJB Container

J2EE App Server

EJB Container

EIS

DB

Legacy

Huihoo - Enterprise Open Source

http://www.huihoo.com
Service-Oriented Architecture
New J2EE Architecture

Rich Clients

J2EE Server: JSP/Servlet/EJB

MIDP Devices

Browsers

DBMS

JDBC

JMS Connectors

Existing Apps

XML/soap

JAX-RPC

JAXM

JAXB

JAXP

EJB

Services

HTML/XML

XHTML/WML

XML/soap

XML/soap

XML/soap

Huihoo - Enterprise Open Source

http://www.huihoo.com
Application Driven
Service Driven
Service on demand
Builder a team

- **Markup developer**
  This will encompass HTML/XHTML development and possibly JavaScript.

- **Presentation developer**
  Responsible for more technical presentation-tier work, rather than designing good-looking HTML.

- **Web-tier Java developer**
  Responsible for MVC framework action classes, JSP tag handlers, and web-tier helper classes.
Builder a team

- **Business object developer**
  Responsible for implementing application business logic. Business object developers will use EJB where appropriate, and should have a sound understanding of EJB.

- **Data access specialist**
  Java developer responsible for efficient DBMS access. Often this role will be taken on by business object developers.

- **DBA**
  Specialist in the underlying database. DBAs can provide valuable assistance to J2EE developers in vital areas such as ensuring good performance and correct locking behavior.
Exercises

- Rewrite, compile, and run a program that use the JSP, Servlet

- Rewrite, compile, and run a program that use the MVC design pattern
Think Beyond

- How to apply MVC framework to your project?
Common Abbreviations and Acronyms

- API - Application programming interface
- CORBA - Common Object Request Broker Architecture
- EIS - Enterprise Information System
- EJB - Enterprise JavaBeans
- ERP - Enterprise Resource Planning
- JAR - Java Archive
- J2EE - Java 2 Platform, Enterprise Edition
- JMS
- JNDI
- JSP
- JTA
- JTS
- OMG
- ORB
Further Reading

- Architecting and Design J2EE Applications SL-425 Sun Microsystems 2000
- Enterprise JavaBeans Programming SL-351 Sun Microsystems 2000
- Rod Johnson Expert One-on-One J2EE Design and Development Wrox Press 2003
Resources

- http://java.sun.com/j2ee
- http://java.sun.com/jcp
- http://www.w3.org/