Java™ 2 Platform, Enterprise Edition (J2EE™)

Overview for Web Application Development

George Grigoryev
Senior Product Manager, Sun Microsystems, Inc.

Pierre Delisle
Senior Staff Engineer, Sun Microsystems, Inc.
Agenda

• Introduction, Java Platforms
• J2EE Community and Future Directions
• Web Application Development in the J2EE World
Java: J2SE – J2ME – J2EE

- **J2SE**: WORA, new generation language
  - Additional APIs (AWT, SWING)
  - Core Language APIs
  - JVM

- **J2ME**: small JVM profiles
  - MIDP
  - CLDC

- **J2EE**: System Services, Middleware
  - Components
  - Containers
  - Services
Good Overview of J2EE Architecture:

*Designing Enterprise Applications with the J2EE Platform, 2nd Edition,*
Inderjeet Singh, Beth Stearns, Mark Johnson, Enterprise Team, 2002
Strength of Java™ Community

- 2.5 Million of registrations at Sun's Java™ Developer Connection
- 652 Java Community Process℠ members
  - 554 companies
  - 98 individuals
- 227 JSRs
  - 60% are led by Java™ partner companies
  - 40% are led by Sun
J2EE Success: Credit Suisse

• Implemented SECOR Monitor, a monitoring tool for the bank's order routing system
  – Checks accuracy of trades
  – Expected to make 10% cost saving and improve efficiency

• Web Services based solution from Ergon Informatik

• Developed on J2EE™ Platform on the Server and J2SE™ Platform on the Client

• Credit Suisse Group employes 80,000 staff with $1.3 Trillion CHF under management
J2EE at Sun Microsystems

- http://java.sun.com/j2ee

Java 2 Platform, Enterprise Edition (J2EE)

Introduction
J2EE technology and its component based model simplifies enterprise development and deployment. The J2EE platform manages the infrastructure and supports the Web services to enable development of secure, robust and interoperable business applications. The J2EE platform is the foundation technology of the Sun ONE platform and Sun's Web services strategy.

Overview | List of Technologies | Sun ONE Developer

Hot Topics

J2EE 1.4 Platform Beta 2 Now Available

The Java 2 Platform, Enterprise Edition (J2EE) defines the standard for developing multi-tier enterprise applications. The J2EE 1.4 platform introduces new APIs which implement core Web services protocols stack. It also introduces new Management and Deployment APIs, new versions of the JavaServer Pages, Enterprise JavaBeans, and Connector APIs, along with other features which establish J2EE 1.4 technology as a premier Web services and enterprise application integration platform.

Java Application Verification Kit (AVK) for the Enterprise 1.4 Early Access
J2EE at BEA

- http://dev2dev.bea.com

Updated August 4, 2003

**WebLogic Platform 8.1 Arrives**
It’s here, and it’s better than ever! Say hello to WebLogic Platform 8.1, which offers powerful new functionality for building and integrating enterprise applications. Don’t be shy — learn more about it today.

**A Tour Through WebLogic Workshop 8.1: Westside Auto Sales**
Learn about the new features of WebLogic Workshop 8.1 as author Scott Eemyan builds a custom end-to-end automotive sales tool and walks you through every step of the process.

**Five Questions about WebLogic Workshop**
Curious about all the buzz? Carl Sjogreen, the product manager for WebLogic Workshop, talks with dev2dev about what makes the newest version of Workshop a must-have for developers.

**Get Your Free Development License**
Have you obtained your free development license yet? BEA now offers free one-year development licenses for WebLogic Platform with no strings attached. Download or order your free license now, and start building enterprise-class applications today.

**WebLogic Integration 8.1 Use Cases**
The WebLogic Integration product team has created this set of use cases to help...
J2EE at Oracle

- [http://otn.oracle.com/tech/java](http://otn.oracle.com/tech/java)

New Tutorial: Managing Component Relationships
This tutorial describes how OTN developers used Oracle9i JDeveloper and the Oracle Business Components for Java (BC4J) framework to manage relationships between components in the Virtual Shopping Mall sample application (from the Industrial-Strength BC4J in Oracle9i JDeveloper Tutorial Series).

From Java Pro Magazine:
Deploy a Web Application with OC4J
This new Java Pro technical article (from the September 2003 issue) overviews the use of Oracle9iAS Containers for J2EE (OC4J) as an application deployment and testing environment.

Build Your Ideal J2EE Environment
Combine the power of Oracle9i JDeveloper and the
J2EE Development Tools

• Borland: http://www.borland.com/jbuilder/
  – JBuilder Personal (free)
  – JBuilder Developer, Enterprise (90 days eval)

• Sun: http://wwws.sun.com/software/sundev/
  – SunONE Studio

• BEA: http://www.bea.com/
  – Workshop 8.1 (12 months free development)

• Oracle: http://www.oracle.com/ip/develop/ids/
  – JDeveloper (free development license, OTN)
J2EE Development Runtime

  - Borland App Server (90 days eval)
  - SunONE App Server 7 PE (free)
- **BEA**: [http://www.bea.com/](http://www.bea.com/)
  - WebLogic 8.1 (12 months free development)
  - Oracle 9iAS (free development license, OTN)
J2EE Deployment Runtime

  - SunONE App Server 7 PE (free)
- Sun: http://java.sun.com/webservices/webservicespack.html
  - Java Web Services Developer Pack 1.2 (free)
- Apache: http://jakarta.apache.org/tomcat/
  - Apache Tomcat 4.1 (free)
  - Apache Tomcat 5.0 (free)
• Java Community Process
  – http://www.jcp.org/

• J2EE Community/News
  – http://www.theserverside.com/

• J2EE hardcore Opensource

• J2EE Technology Journal
  – http://www.sys-con.com/java/

• J2EE EJB Reference
J2EE and Microsoft

- J2EE is winning against .NET in the Enterprise (for now)
- J2EE interoperates with .NET
  - J2EE will support WS-I Basic Profile 1.0
- Allows to annotate fields, methods and classes with attribute-value pairs
  - JVM maps to CLR
  - Java maps to C#
  - J2EE maps to .NET
- J2EE provides a “safe harbor” against MSFT for software startups
J2EE Web Services in J2EE 1.4

- **JAX-RPC 1.1**
  - Implements WS-I Basic Profile 1.0
  - SOAP and WSDL based interoperability
  - Implements WSDL to Java mapping
  - Servlet based endpoint
  - Stateless session bean endpoint

- **JAXR 1.0**
  - Implements UDDI Registry protocol

- **JSR 109 (Web Services for J2EE)**
  - Deployment descriptors
Ease-of-Development in J2EE

• Main thrust for J2EE Platform going forward: **Ease-of-Development**

• J2EE Platform is designed to serve the needs of every developer:
  - Enterprise Developer
  - Corporate (Workgroup) Developer
  - Content (Web) Developer

• J2EE is becoming a ubiquitous platform for every type of application
  - Not just the Enterprise
Currently, to create an EJB:

```java
Context initial = new InitialContext();
Object objref = initial.lookup("java:comp/env/ear/EJBExample/SimpleFoo");
FooHome home = (FooHome)
    PortableRemoteObject.narrow(objref,FooHome.class);
Foo myFoo = home.create();
```

Wouldn't It be Nice, instead:

```java
private @create Foo myFoo;
```
What is the Ease-of-Development?

- Tools are important, but...
- Code should be easy to write, understand and maintain as well

Ease-of-Development

Content Developer
- Web Dev Tools: Macromedia, etc...
- JSTL-JSP

Corporate Developer
- Tools: RAVE, others
- J2EE Metadata

Enterprise Developer
- Tools: IDEs, emacs

“Easy-to-Develop-On” J2EE Platform
Web Application Development in the J2EE World
The Web Tier
What is a Web Application?

The WEB

HTTP

Server
Application
HTTP Communication Model

- Stateless
- No immediate feedback
- No details on how request is made
HTTP Request

GET /index.html HTTP/1.1

Host: java.sun.com
User-Agent: Mozilla/4.5 [en]
Accept: image/gif, image/jpeg, image/pjpeg, */*
Accept-Language: en
Accept-Charset: iso-8859-1,*,utf-8

optional request body
HTTP/1.1 200 OK

Last-Modified: Mon, Aug 4 2003 22:10:40 GMT
Date: Wed, Aug 8 2003 14:23:35 GMT
Status: 200
Content-Type: text/html
Content-Length: 59

<html>
  <body>
    <h1>Hello World!</h1>
  </body>
</html>
Request Parameters

- Request can also contain additional info in the form of parameters
  - In the URI itself as a query string
  - As part of the request body

http://acme.com/register?name=Paul+Smith&state=CA
What is a “Java Servlet”? 

• Extension mechanism for dynamic content generation over HTTP, based on the Java™ Servlet API

• Simple architecture where the servlet container manages the lifecycle of servlets

• Extremely powerful
  • The full Java platform available to your web application
Servlet Invocation

- Request
- Response
- init()
- service()
- destroy()

Server

- Session
public class HelloServlet extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response) {
        response.setContentType("text/plain");
        PrintWriter out = response.getWriter();
        out.println("Hello World!");
    }
}
public class HelloServlet extends HttpServlet {
  public void doGet(HttpServletRequest request,
         HttpServletResponse response) {

    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    out.println("<html>");
    out.println("Hello World!");
    out.println("<br>");
    JspCalendar clock = new JspCalendar();
    out.println("Today is");
    out.println("<ul>");
    out.println("<li>Day of month: ");
    out.println(clock.getDayOfMonth());
    out.println("</ul>");
    out.println("</html>");
  }
}
Servlets are great, but...

- Processing code and HTML elements are lumped together → BAD
  - Need to know Java to design HTML pages
  - Need to update servlet code to update L&F and support new types of clients
  - Hard to take advantage of web-page development tools when designing application interface
JavaServer Pages™
public class HelloServlet extends HttpServlet {
    public void doGet(HttpServletRequest request,
                       HttpServletResponse response) {
        response.setContentType("text/plain");
        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println("Hello World!");
        out.println("<br>");
        JspCalendar clock = new JspCalendar();
        out.println("Today is");
        out.println("<ul>");
        out.println("<li>Day of month: ");
        out.println("clock.getDayOfMonth()");
        out.println("</ul>");
        out.println("</html>");
    }
}
<jsp:useBean
    id="clock" class="calendar.JspCalendar" />

<html>
    Hello World!<br>
    Today is
    <ul>
        <li>Day of month: <%= clock.getDayOfMonth() %>
        <li>Year: <%= clock.getYear() %>
    </ul>
</html>
How does it work?

- **JSP page**
  - A textual document describing how to create a *response from a request*
  - Mixture of template data and dynamic content
  - “Inside-Out” Servlet

- **JSP engine**
  - Compiles JSP page to Servlet
  - Executes servlet's service() method

- **JSP page is**
  - Compiled once
  - Executed many times
Key JSP concepts

• Directives
  • `<%@ directive %>`

• Scripting elements
  • `<%! declaration %>`
  • `<% scriptlet %>`
  • `<%= expression %>`

• Implicit objects
  • request, response, session, application, out, page, pageContext...

• Standard Actions
  • `<jsp:useBean>`, `<jsp:getProperty>`, `<jsp:setProperty>`
  • `<jsp:include>`, `<jsp:forward>`,
Example: Output Shopping Cart

```jsp
<jsp:useBean id="cart" class="acme.com.bean.Cart" scope="request"/>

<html>
  <body>
    <table>
      <%
        for (int i=0; i<cart.size(); i++) {
          CartItem item = cart.getItem(i);
      %>      <tr>
        <td> <%= item.getName() %> </td>
        <td> <%= item.getPrice() %> </td>
      </tr>
      <% } %>
    </table>
  </body>
</html>
```
Very nice, but who's the target audience?

**Mike**
Hard-core developer. 
Handles all business logic and back-end matters. 
Expert in Java, XML, databases, etc.

**Ernie**
Jack of all trades. 
Not an expert in anything, but will eventually get the job done...

**Philippe**
Web site designer. 
Knows how to make a website look really cool! 
HTML/JavaScript expert
Higher levels of abstraction make it easier to develop with JSPs:
  - Less skills required
  - Faster time to market
How should the technology evolve to reduce the cost of developing sophisticated web applications?
Custom Actions

- Design your own tags - Extremely powerful!
- Java code is executed when tag is encountered
- Easy to use by page authors
  - Look like HTML
- Encapsulate functionality
  - Reusable, maintainable
  - Keep Java Code off the JSP page
- Organizations can build a rich repository of tag libraries
Much easier...

```html
<html>
  <body>
    <table>
      <acme:cart id="item">
        <tr>
          <td><%= item.getName() %></td>
          <td><%= item.getPrice() %></td>
        </tr>
      </acme:cart>
    </table>
  </body>
</html>

But...
Still need to refine...

- Tag Libraries
  - Should everyone reinvent the wheel?
  - Standard tag libraries → JSTL

- Java as the scripting language in JSP scares many people (e.g. Philippe)... can we simplify?
  - Expression Language: A language adapted for the web developer
Expression Language
Easy Access/Manipulation of Application Data

Request Data
- Parameters
- Headers

Beans

JSP Scoped Attributes
- Page
- Request
- Session
- Application

Business Logic

Presentation
Before...

1. Must declare

```jsp
<jsp:useBean id="product" type="acme.com.Product" scope="request"/>

...

Product Name: <%= product.getName() %>

...<% if (product.getManufacturer().equals("ACME")) { %>

...<% } %>
```

2. Must know type

3. Awkward syntax

4. Knowledge of scripting language required even for simple manipulations
After

1. Direct access

Product Name: ${product.name}
...
<c:if test="${product.manufacturer == param.manufacturer}">
  ${product.name}
</c:if>

2. Easier syntax

3. All app data easily accessible

4. Better adapted expression language
Expression Language

1. Syntax

${expression}$

value="product_${name}.${type}"
2. Application data directly accessible

${foo} -> pageContext.findAttribute("foo")

Implicit objects:
• \{page|request|session|application\}Scope
• param/paramValues, initParam
• header/headerValues, cookie
• pageContext
3. Beans and collections rule

${user.address.city}$

${products[product.id]}

${products["DCR-PC100"]}$
4. Operators

relational:
  ==, !=, <, >, <=, >=

arithmetic:
  +, -, *, /, %

binary:
  and, or, not
5. Automatic type conversions

\[
\text{int} \leftarrow \text{Integer} \quad \text{begin} = \"{\text{request.beginValue}}\"
\]
6. Default values

```xml
<c:set var="city"
value="${user.address.city}"
default="N/A"/>
```
7. Extensibility through functions

truncate name to 30 chars and display it in uppercase

${fn:toUpperCase(fn:substring(name, 0, 30))}$
EL Support Actions

• `<c:out>` – Display EL expression

```html
<c:out value="\${customer.address.city}" default="unknown"/>
```

• `<c:set>` – Set scoped variable

```html
<acme:foo>
    <acme:bar>
        <x:atag>...</x:atag/>
    </acme:bar>
</acme:foo>
```

```html
<c:set var="bar">
    <x:atag>...</x:atag/>
</c:set>
```

```html
<acme:foo bar="\${bar}"/>
```

• `<c:remove>` – Remove scoped variable
Conditional Actions

- Simple conditional execution
  ```xml
  <c:if test="${user.visitCount == 1}">
    This is your first visit!
  </c:if>
  ```

- Mutually exclusive conditional execution
  ```xml
  <c:choose>
    <c:when test="${verbosityLevel == 'short'}">
      ${product.shortDescription}
    </c:when>
    <c:otherwise>
      ${product.longDescription}
    </c:otherwise>
  </c:choose>
  ```
Iteration Actions

1. All J2SE collection types supported

2. Current item

3. Iteration status

• begin, end, step
• `<c:forTokens delims="...">`
Standard Actions for Common Needs

URL related Actions
Hypertext Links

```xml
<c:url url="/exec/add"
    var="myUrl">
    <c:param name="name" value="${param.name}"/>
    <c:param name="country" value="${param.country}"/>
</c:url>

<a href="${myUrl}">Add Member</a>
```

Parameter name and value automatically URL encoded
<jsp:include> — Webapp Centric

Web Application
Same Context
Relative URL
/foo/bar.jsp
bar.jsp

<jsp:include>
<c:import> — URL Centric

Internet Resources
Absolute URL
http://acme.com/foo
ftp://xyz.org/README

Web Application
Same Context
Relative URL
/foo/bar.jsp
bar.jsp

Web Application
Foreign Context
Relative URL
context="/global"
/copyright.html

<c:import url="..."/>
Standard Actions for Common Needs

Internationalization and Text Formatting
Webapp for Global Markets

Worldwide Users

Web Application
Locale

Worldwide Users

Chinese, …

Preferred locales

Spanish, …

Web Application

1. Request sensing

2. Application-based

Login

Prefs

Session

Locales
I18N Architecture

Worldwide Users

Web Application

1. One page per locale

controller

JSP

ch

fmt:formatNumber>

fmt:parseNumber>

JSP

es

fmt:formatDate>

fmt:parseDate>
I18N Architecture

Worldwide Users

Web Application

2. One page for all locales

Resource Bundles

JSP

<fmt:message key="...">
How Easy to I18N a Webapp?

<fmt:message key="welcome"/>
How Easy to I18N a Webapp?

Uses Default Localization Context

Request

1. Client’s Locale preferences   2. Resource bundle basename

3. Fallback Locale

**J2SE Resource Bundle Selection Algorithm**

```
<fmt:message key="welcome"/>
```
I18N Flexibility

1. Client's Locale preferences
2. Resource bundle basename

Request

J2SE Resource Bundle Selection Algorithm

Configuration parameter

1. fmt.locale
   <fmt:locale>

2. fmt.basename
   <fmt:bundle>
   bundle attribute

3. Fallback Locale

JSP Expression

<fmt:message key="welcome"/>
   <fmt:messageArg value="${visitCount}"/>
Standard Actions for Common Needs

Database Access
Database Access

Dynamic Content

Database

.jsp
MVC Architecture

Beans → Business Logic → SQL → Database

.jsp
SQL Actions

Query the database
<sql:query>

Easy access to
result set
Result
ResultSupport

Update the database
<sql:update>
<sql:transaction>
DataSource

- All DB actions operate on a DataSource
- Different ways to access a DataSource
  - Transparent collaboration
    - via configuration parameter `sql.dataSource`
      ```xml
      <db:query query="..." />
      ```
  - Explicit collaboration
    - Object provided by application logic
    - Object provided by `<sql:dataSource>` action
      ```xml
      <sql:dataSource var="dataSource"
          driver="org.gjt.mm.mysql.Driver"
          url="jdbc:..."/>
      <sql:query dataSource="${dataSource}" .../>
      ```
<sql:query var="customers" dataSource="${dataSource}">
    SELECT * FROM customers
    WHERE country = 'China'
    ORDER BY lastname
</sql:query>

<table>
<c:forEach var="row" items="${customers.rows}">
    <tr>
        <td>${row.lastName}</td>
        <td>${row.firstName}</td>
        <td>${row.address}</td>
    </tr>
</c:forEach>
</table>
Updating a Database

```xml
<sql:transaction dataSource="${dataSource}"

  <sql:update>
    UPDATE account
    SET Balance = Balance - ?
    WHERE accountNo = ?
    <sql:param value="${transferAmount}"/>
    <sql:param value="${accountFrom}"/>
  </sql:update>

  <sql:update>
    UPDATE account
    SET Balance = Balance + ?
    WHERE accountNo = ?
    <sql:param value="${transferAmount}"/>
    <sql:param value="${accountTo}"/>
  </sql:update>

</sql:transaction>
```
Standard Actions for Common Needs

XML
XML Everywhere
1. Easy Access to XML Data

XPath

$doc/employee[@name=$param:name]
1. Easy Access to XML Data

XPath – expression language for XML actions

<x:out>
<x:set>
<x:if>
<x:choose>
<x:forEach>

select attribute to specify XPath expression
2. Get the XML Data

```xml
<x:parse>
in  xmlUrl, xmlText, body content
out  var, varDom
perf filter (org.xml.sax.XMLFilter)
```
3. XSLT Transformation

```xml
<x:transform>
  in  xmlUrl, xmlText, body content
  in  xsltUrl, xsltText
  out var, result, page
  perf transparent caching of Transformer objects possible
</x:transform>
```
Demo / Q&A
Resources

- JSP home page
  http://java.sun.com/jsp

- JSTL home page
  http://java.sun.com/jstl

- Java Web Services Developer Pack
  http://java.sun.com/xml

- Books
  - JavaServer Pages, Hans Bergsten, O'Reilly
  - JSTL in Action, Shawn Bayern, Manning