



Tag Libraries in JavaServer Pages

Eduardo Pelegri-Llopart

January 25th, 2000

Outline

- **Standard and Custom Actions**
- **Roles**
- **Types of Tag Libraries**
- **Using Tag Libraries**
- **Implementing Tag Libraries**
- **Jakarta-Taglibs**
- **JSR on Standard Tag Library**

Standard Actions

- Available in JSP 1.0 and JSP 1.1
- Common actions:
 - useBean
 - setProperty
 - getProperty
 - include
 - forward
 - plugin

Custom Actions

- **Encapsulate functionality**
 - Reusable
 - Maintainable
- **Easier to**
 - Author manually
 - Manipulate by tool
- **Provide**
 - Portable semantics
 - Portable tool support

Roles

- **User of Tag Library**
 - Web page author
 - Expert at UI and presentation
- **Author of Tag Library**
 - Developer
 - Expert in access to data, service, etc.
- **Customizer of Tag Library**
 - Not yet well developed

Types of Tag Libraries

- **Web Site Development Group**
 - Uniform Access to Data
- **Services Provider**
 - Access
 - Presentation
 - Customization of Service
- **Platform Vendor**
 - Access to custom functionality
 - Specialized implementations of standard functions

Using Tag Libraries

- **Activated using a taglib directive**
`<%@ taglib uri="identifier" prefix="prefix" %>`
- **TagLib introduces new actions**
- **Action at request time**
 - "Tag Handler" object
 - Has access to all page objects

Simple Action

- / <x:includeSaleOfDay att1="one" />
- / <x:logAction .../>
- / <x:forwardBasedOnClient .../>
- / <x:applyXSL .../>
- Do an action passing some arguments
- Most custom tags are like this
- Very easy to write

Simple Nested Action

/ <x:includelfVIP>

VIP stuff <x:VIPgreeting/>

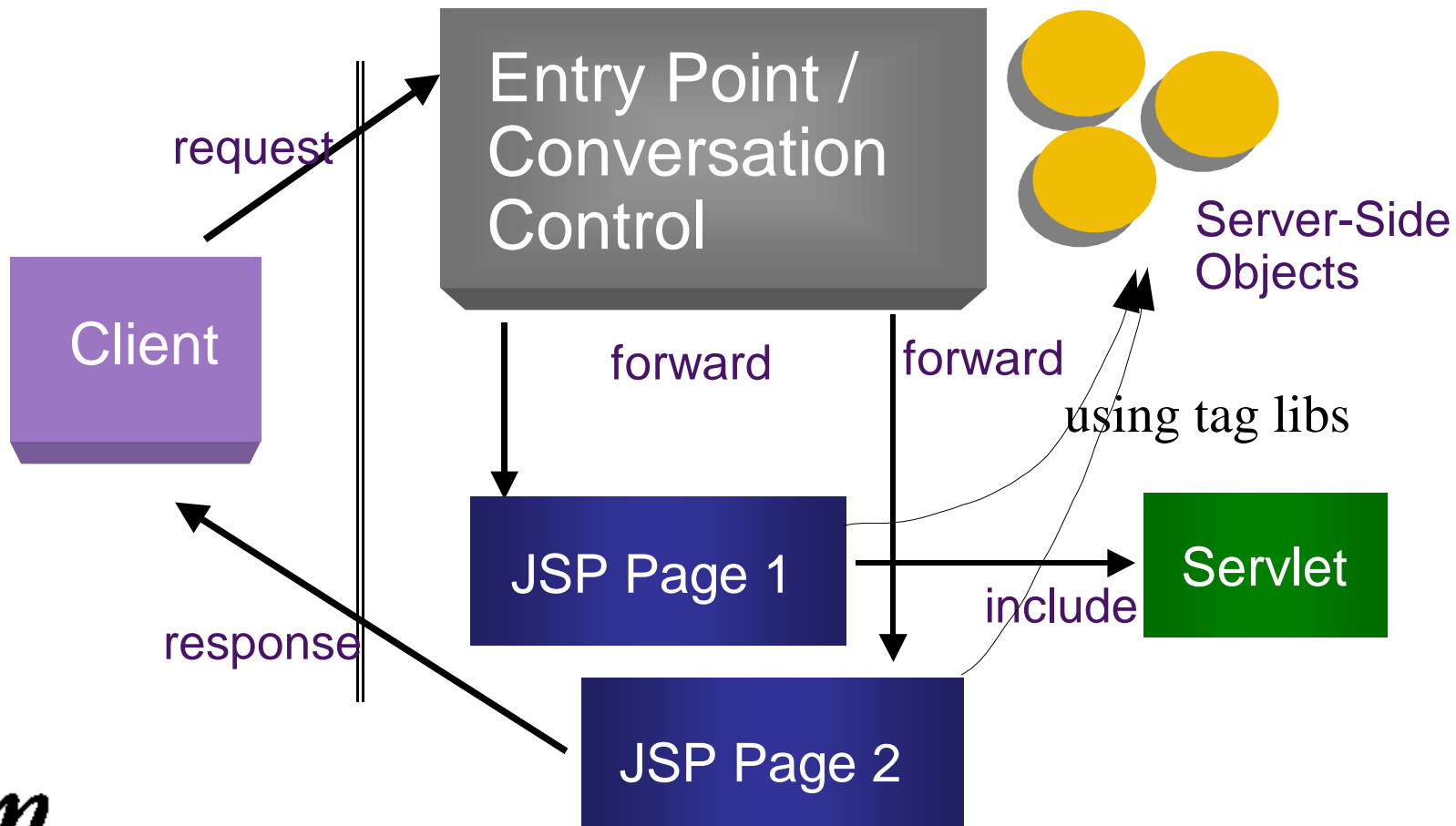
</x:includelfVIP>

- **Conditionally include body**
 - cannot manipulate body
- **May interact with inner actions**
 - e.g. To define a scope
- **Many custom tags are like this**
- **A bit more work but still easy to write**

Nested and Iterative Action

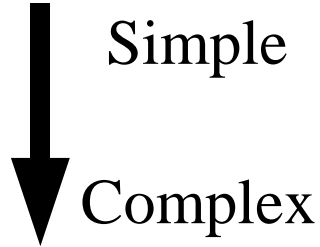
- / **<x:loopOverTopSales>**
 stuff <x:insertSaleItem/> more stuff
</x:loopOverTopSales>
- **Creates some context**
- **Iterates ≥ 0 times on body**
- **Can manipulate body**
- **Fewer of these**
- **Harder to write**

Typical Organization



Implementing Tag Libraries

- **Tag Library Descriptor**
- **Tag Handler**
 - **Simple**
 - **Nested**
 - **Iterative**
- **Tag Body**



Tag Library Descriptor

- **Defines**
 - new tag names
 - valid attributes
- **Maps**
 - action tags to tag handler classes
- **Provides Authoring Info**
 - very basic for 1.1
- **Organized as an XML document**

Tag Handler

- **Request-Time Server Side Object**
- **Evaluates an Action**
- **Has access to:**
 - attributes using setter/getter
 - server-side objects
 - enclosing actions
- **Can have access to its content**
- **Implements one of 2 interfaces:**
 - Tag - for simple handlers
 - BodyTag - for more power

Simple Action

/ <x:dolt att1="one" />

- **Handler needs to define**
 - `setAtt1(String s) { ... }`
 - `doStartTag() { ... }`
- **Handler implements Tag**
- **Usually extends TagSupport**
 - inherits other standard behavior

Simple Action with Body

```
/ <x:includeWhen condition="one">  
body  
</x:include>
```

- **Like Simple Action but**
 - **doStartTag()** returns
 - EVAL_BODY_INCLUDE - if include
 - SKIP_BODY - if skip
- **Cannot manipulate body**
- **Can insert before and after body**
 - **doStartTag()** and **doEndTag()**

Nested Action

/ **<x:outer>**

<x:inner />

</x:outer>

- **OuterHandler**

- **doStartTag()** returns **EVAL_BODY_INCLUDE**

- **InnerHandler**

- **doStartTag()** calls into
findAncestorWithClass(OuterHandler.class)

BodyContent

- " **Abstraction of the *evaluation* of body**
- " **Managed by page implementation**
- " **Nested**
- " **Provided to tag Handler object**
 - " **Can pass to its enclosing BodyContent**
 - " **Can read and process**
 - " **Uses methods in BodyTag interface**

Body with Manipulation

/ **<x:makeQuery>**

SQL statement

</x:loop>

- *doStartTag()* returns **EVAL_BODY_TAG**
- *doInitBody()* is called
- *setBodyContent()* passes *BodyContent*
- body gets evaluated into *BodyContent*
- *doAfterBody()* returns **SKIP_BODY**
- *doEndTag()* is called

Action with Iteration

/ `<x:loop n="34">`

 This is iteration `<x:insertCount/>`

`</x:loop>`

- **LoopHandler**

- `doStartTag()` initializes loop

- `doAfterBody()` checks end of loop

- **InsertCountHandler**

- `doStartTag()` looks for enclosing loop and inserts

Full Example

```
<html>
<%@ taglib uri="/tlds/simpleDB.tld" prefix="q" %>
<q:queryBlock connData="conData1">
  <q:queryStatement>SELECT account,
    balance...</q:queryStatement>
  The top 10 accounts and balances are:
  <table>
    <tr><th>ACCOUNT</th><th>BALANCE</th></tr>
  <q:queryCreateRows from="1" to="10">
    <td><q:queryDisplay field="ACCOUNT"/></td>
    <td><q:queryDisplay field="BALANCE"/></td>
  <q:queryCreateRows>
  </table>
</q:queryBlock>
</html>
```

Handling XML Content

- **JSP Pages Can:**
 - **Accept / Generate XML Through**
 - Raw parser (now)
 - Data binding (later)
 - **Manipulate Content Through**
 - DOM access
 - XSL/XSLT
 - **Encapsulated in Tag Library actions**

Jakarta-Taglibs

- **An Apache project @ Jakarta**
- **Code examples of TagLibs**
- **Goals:**
 - educational
 - basis for cut-and-paste examples
 - experimental ideas
 - useful libraries
- **Not the Standard Tag Library**

JSR Standard Tag Library

- **Goals include:**
 - **Actions for Data Sources**
 - LDAP
 - JDBC queries
 - **Actions for control of presentation**
 - Conditional
 - Iterative

Standard Tag Library

- **Benefits include:**
 - **Widespread availability**
 - Ease of use by authors
 - Better support by tools
 - **High quality**
 - **Mindshare: training, portability, maintainability**
 - **Enables specialized implementations**

JSR Endorsers

- Abobe
- Allaire
- ATG
- BEA
- Bluestone
- Caucho
- Exoffice
- Jakarta (ASF)
- IBM
- Gefion
- KL Group
- K&A
- Macromedia
- NetObjects
- Netscape
- New Atlanta
- Oracle
- Visual Cafe Group
- Orion
- Turbine (ASF)
- Sun Microsystems
- SilverStream

TagLibs in JSP Spec

- **Next JSP & Servlet spec may:**
 - Improve authoring support
 - Improve composition mechanisms
 - Enable customization of libraries
 - Tune mechanism based on experience

Summary

- **Tag Libraries is key JSP 1.1 feature**
- **Encapsulated functionality**
- **High interest in user community**
- **Standard Tag Library has wide industry interest**
- **Practical experience will drive evolution**

Resources

- 1 <http://java.sun.com/products/jsp>
- 1 <http://java.sun.com/products/servlet>
- 1 <http://java.sun.com/j2ee>
- 1 <http://java.sun.com/apm>

- " jsp-interest@java.sun.com
- " servlet-interest@java.sun.com
- "
- " <http://jakarta.apache.org/>