Chapter 17
Web-Based User Interface Design and Programming

© Pearson Education Limited 2005

Topics
- Terminology and technologies
  - Enabling technologies for web client tier
    - HTML
    - scripting languages
    - applets
  - Enabling technologies for web server tier
    - servlets
    - beans
    - tags
    - JSP
    - Struts
- Additional references

Concepts and technologies
- Web-based application – an Internet browser manages the rendering of UI content, but the business logic and database state exists on a server
- HyperText Markup Language (HTML)
  - A HTML-formatted web page is a mix of the presentation content (e.g. some text) and rendering instructions (e.g. font size)
  - HTML can be used to apply rendering instructions on a text in standard UI components, including Swing components
- JavaScript – a scripting language used in HTML document to perform quick validation of inputs, simple animations, and other activities to make the web page "alive"
- Applet – technology that makes the UI dynamic (in layout and content) within a browser
- <object> component in Swing provides a UI container that can be placed in a web page
- HTML uses the <object> tag to download an object (e.g. an applet) that lives in the URL space to the client
- Applet lives in a sandbox environment – it has limited and restricted access to system resources
- Applet can be digitally signed – it can access system resources if an explicit permission is given to it → trusted applet

JavaScript by example
- JavaScript can access the current web page via its objects
- This is shown on Line 6 where the script checks the value of text field named percent (declared on Line 20 – next slide)
- The value is returned as a string which needs to be converted into an integer via the call to a built-in function parseInt()
- If the input value is valid (Lines 8 – 9), then the script returns true, otherwise it displays a message box indicating the wrong input and returns false

Calling JavaScript from HTML
- The server URL, which the form will call, is placed in the action attribute
  - The form processor is handled by a servlet called SearchMovie, located on the root of the server (indicated by /)
  - The checkIt function (Line 5) checks the value of input text, presented by Line 26, to see if it has the correct value (as demanded by the lower and upper parameters of checkIt())

© Pearson Education Limited 2005
Applet in a sandbox

- **Applet**
  - program that runs inside a web browser
  - runs within a sandbox determined by Java run-time security system
  - built using an application framework (library) rooted at `JApplet`

- **Applet is not allowed:**
  - read from a file system of the host machine (including files, properties, etc)
  - write or delete a file
  - connect to a network port on any machine except the HTTP server it comes from
  - execute or load another program/library/DLL code

Applet by example

- Applets are not required to have a `main()` – you put any startup code in `init()`
- The `init()` method is responsible for putting all the components on the form using the `add()` method
  - In this program, the only activity is putting a text label on the applet, via the `JLabel` class
  - Swing requires to add all components to the "content pane" of a form, and so you must call `getContentPane()` as part of the `add()` process

```java
public class Applet1 extends JApplet {
    public void init() {
        getContentPane().add(new JLabel("Applet!");
    }
}
```

Applet method summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>void destroy()</code></td>
<td>Called by the browser or applet viewer to inform this applet that it is being reclaimed and that it should destroy any resources that it has allocated.</td>
</tr>
<tr>
<td><code>void init()</code></td>
<td>Called by the browser or applet viewer to inform this applet that it has been loaded into the system.</td>
</tr>
<tr>
<td><code>void start()</code></td>
<td>Called by the browser or applet viewer to inform this applet that it should start its execution.</td>
</tr>
<tr>
<td><code>void stop()</code></td>
<td>Called by the browser or applet viewer to inform this applet that it should stop its execution.</td>
</tr>
</tbody>
</table>

Running applet inside browser

- Unfortunately not as simple as:

```
<applet code=Applet1 width=100 height=50>
</applet>
```

- Because of the browser and language wars, Sun could not expect browsers to support the correct flavor of Java, and the only solution was to provide some kind of add-on that would conform to a browser’s extension mechanism
  - With Internet Explorer, the extension mechanism is the ActiveX control, and with Netscape it is the plug-in
  - In your HTML code, you must provide tags to support both, but you can automatically generate the necessary tags with the `HTMLconverter` tool that comes with the JDK download

Running applet from command line

- To create an applet that can be run from the console, command line, you simply add a `main()` to your applet that builds an instance of the applet inside a `JFrame`
  - `Applet1b.java` modified to work as both an application and an applet (the applet is created and added to a `JFrame` so that it can be displayed):  

```java
public class Applet1c extends JApplet {
    public void init() {
        getContentPane().add(new JLabel("Applet!");
    }
    public static void main(String[] args) {
        JApplet applet = new Applet1c();
        JFrame frame = new JFrame("Applet1c");
        frame.setTitle("Applet1c");
        frame.setSize(100,50);
        applet.init();
        applet.start();
        frame.setVisible(true);
    }
}
```

Servlet (Server Applet)

- The basic computational unit that a Java-based Web server uses to handle multiple clients (HTTP requests)
  - request is processed and data returned in a stateless transaction (normally in HTML output)
  - There is no client code as such in the servlet. The client is a HTML page or other pages, such as JSP or ASP.
**Beans**

- JSP is able to use JavaBeans and Enterprise JavaBeans (EJB)
- JavaBeans are a set of Java classes that follow predefined rules to allow inspections and modifications on their properties.
- EJB is similar to JavaBeans but it supports the capability of accessing shared business logic and a shared database. EJB is fully managed by a server container (usually an application server).
- Beans conventions:
  - Each attribute of the bean that will be exposed publicly should have at least a method called get().
  - The get() method should return the same type value that the set() method takes as an argument.
- JSP supports an introspection mechanism on beans that allows form values to be automatically populated into beans from a JSP page by using the jsp:setProperty tag.

**JSP by example – bean**

A simple bean (Animal.java) that implements a few properties of animal:
```java
public class Animal {
    String commonName = null;
    String speciesName = null;
    float adultWeight = 0;
    float adultHeight = 0;
    int topSpeed = 0;
    String description;

    public String getSpeciesName() {
        return this.speciesName;
    }

    public String getCommonName() {
        return this.commonName;
    }
}
```

**JSP by example –– input form**

This page, after it’s filled out, has been shown on Slide 16. When the submit button is clicked, the values are sent to a second page (next slide).

**JSP by example –– HTML page**

This page is actually straight HTML, defining a standard form that takes the various properties of an animal.
The display code uses both the `jsp:getProperty` tag and the raw `getX()` calls to the object itself. We need to use the raw calls to compute the English unit equivalents of the metric values.

```jsp
<s:body><jsp:getProperty name="animal" property="*"/></s:body>
```

The result of this is that the newly created animal bean is populated with the values from the previous page.

```jsp
&animal.getTopSpeed() * 0.621<BR>
&animal.getAdultHeight() * 3.28<BR>
&animal.getAdultWeight() * 2.2<BR>
```

To reuse common functionalities that you repeatedly have to code in Java on the JSP page, you might need to present metric values in English units, as in the previous example. JSP syntax can be extended by adding new custom tag libraries to JSP.

To define a Java class that extends `BodyTagSupport` that simply takes the values in `Address.java` and displays them.

```jsp
<s:setProperty name="animal" property="*"/>
<s:useBean id="animal" scope="request" class="demo.Animal"/>
```

Struts builds applications that are compliant Web applications (Webapps), which means:

- A standard directory structure
- Certain standard configuration files (web.xml and so on)
- Dynamic functionality deployed as Java classes and .jsp pages
- A standard Web Archive (.war file) format for deployment

Struts is a Model-View-Controller architecture that makes it easy to build flexible applications and a set of JSP custom tags for building JSP pages.
**Struts – tag libraries**

- Used for creating View components.
  - The Struts tag libraries provide a set of JSP custom tags that are generally understandable by both JSP developers and page designers.
    - The custom tags have names like `<html:text>` and `<logic:iterate>`.
- The tags include:
  - HTML tags
  - Bean tags
  - Logic tags
    - The HTML custom tags are used in JSP files for:
      - Generating HTML elements.
      - Coordinating form processing.
      - Linking the JSP pages (View components) into the rest of the Struts framework.
- It is possible (though not recommended) to use Struts without using any of the Struts custom tags.
- In time, JSTL (Java Standard Tag Library) and JSF (JavaServer Faces) will make the proprietary Struts tags obsolete.

**Struts – MVC for webapps**

- Model components (beans) provide access to the information from which the user presentation is built.
- View components are the JSP files that render the HTML to be sent to the user.
- A Controller component performs several primary activities:
  - Validates that the data entered by the user was valid.
  - Makes decisions about which Model components need to be accessed or updated, and manages these activities.
  - Collects the data that the View component will need for display.
  - Decides which View component should be displayed to the user.

**Struts – implementation overview**

- A web-based application means that an Internet browser manages the rendering of UI content, but the business logic and database state exists on a server.
- In other than applet technology, the presentation logic is also on the server.
- Scripting languages can be used in HTML documents to enrich user interactions.
- Servlet is a dedicated service running on a server machine to serve multiple clients.
- Java Server Pages (JSP) is a variation of the servlet technology. JSP separates the business logic from the presentation.
- JSP supports reuse of components from previously implemented web pages via the Custom Tag Library.
- Struts – an open-source project from Apache – provides an additional level of decoupling between the presentation, the business logic and the data obtained from business objects.

**Summary**

- A web-based application means that an Internet browser manages the rendering of UI content, but the business logic and database state exists on a server.
- In other than applet technology, the presentation logic is also on the server.
- Scripting languages can be used in HTML documents to enrich user interactions.
- Servlet is a dedicated service running on a server machine to serve multiple clients.
- Java Server Pages (JSP) is a variation of the servlet technology. JSP separates the business logic from the presentation.
- JSP supports reuse of components from previously implemented web pages via the Custom Tag Library.
- Struts – an open-source project from Apache – provides an additional level of decoupling between the presentation, the business logic and the data obtained from business objects.