Maintenance and Java Server Pages

Jeff Offutt
http://www.cs.gmu.edu/~offutt/

SWE 642
Software Engineering for the World Wide Web

sources: Professional Java Server Programming, Patzer, Wrox, 14

JSP Maintenance Problems

• Presentation and content are not always well separated
  – Java mixed with the HTML can be very hard to understand

• Most developers are not good at establishing levels of abstraction in JSPs

• Books, articles, and web resources focus on JSP syntax, not style and design
  – When they do, they seldom explain why something should be done
First Rule of Formatting JSP

• JSP is somewhat messy (like JavaScripts)
  – Hard to read
  – Hard to debug
  – Hard to get right
  – Hard to maintain

• Strategy:
  Keep a minimum of Java in the JSP, do most of the programming with separate Java:
  • Servlets
  • Beans
  This allows separation of concerns – good OO design

JSP : Readable HTML

• Make JSP look like HTML with Java calls, not Java with some HTML
• Move all of the business logic out of the JSP
• Java that generates HTML is hard to maintain:
  – Humans have trouble viewing HTML as “normal text”
  – The quotes (“"”) are very hard to read
• Let HTML developers write HTML, and Java developers write Java

The system design must support these goals
**J2EE Assumptions about Data**

- Data values: The contents of memory
- Data structure: Types, organization and relationships of different data elements
- Data presentation: How the data is shown to humans

J2EE assumes that data:
- values change very frequently (during execution)
- presentation changes occasionally
- structure changes very infrequently

---

**JSPs in a Multi-Tier Architecture**

Each software layer communicates only with adjacent layers.
Beans and Java classes that are used directly by the JSPs

Parsing, cookie handling, ... library classes that are used by web software
JSPs in a Multi-Tier Architecture (4)

JSPs

- JSP resources
  - JSP beans
  - Tag extensions

Swing

Web-specific library classes

Non UI-specific presentation layer
- Models
  - Library classes

Business logic layer
- (Session EJBs)

Enterprise resource abstraction layer
- (Entity EJBs)

Enterprise resources
- Databases
  - Legacy systems

Presentation logic in Swing that produces output that is independent of UI.

JSPs in a Multi-Tier Architecture (5)

JSPs

- JSP resources
  - JSP beans
  - Tag extensions

Swing

Web-specific library classes

Non UI-specific presentation layer
- Models
  - Library classes

Business logic layer
- (Session EJBs)

Enterprise resource abstraction layer
- (Entity EJBs)

Enterprise resources
- Databases
  - Legacy systems

Presentation logic that produces output that is independent of UI (an alternative to Swing)
JSPs in a Multi-Tier Architecture

- JSPs
  - JSP resources
  - JSP beans
  - Tag extensions

Web-specific library classes

Swing

Non UI-specific presentation layer
  - Models
  - Library classes

Business logic layer
  - (Session EJBs)

Enterprise resource abstraction layer
  - (Entity EJBs)

Enterprise resources
  - Databases
  - Legacy systems

Business logic processing not concerned with presentation or data storage

Classes used to interface with the resources level, often EJBs
JSPs in a Multi-Tier Architecture

If the system is designed well, there is little code here.

Non-java resources for data storage, including databases.

1. JSPs
2. JSP resources
3. JSP beans
4. Tag extensions

Web-specific library classes

Swing

Non UI-specific presentation layer
Models
Library classes

Business logic layer
(Session EJBs)

Enterprise resource abstraction layer
(Enterprise EJBs)

Enterprise resources
Databases
Legacy systems

Databases

Legacy systems
Multi-Tier Architecture Summary

• This model allows very clean separation of the software that handles the data values, structure, presentation, and storage

• In small applications, some levels can be skipped

• The need for this separation is hard to see with small applications – maintenance is only hard when systems get big