Model-Controller Interfacing for Struts-Based Web Applications

Presenter: Deepti Bhardwaj
Advisor: Dr. Chris Pollett
Committee: Dr. Robert Chun
Dr. Agustin Araya
Purpose

- Traditional Desktop-based IDEs such as Eclipse and NetBeans.
  - Installation and configuration are required
  - User responsible for his workspace

- Existing Web-based IDEs are Mozilla Skywriter or Bespin, and Aurorasdk IDE.
  - No installation is required
  - Can be used anywhere with internet connection
Purpose

- Limitations of existing Web-based IDE
  - Need to write a lot of code
  - Need to set up the database schema for database-driven Web applications
Introduction – StrutsHib IDE

- Our StrutsHib Web-based IDE allows users to develop Struts based web applications with minimal hand-coding.

- It is developed using Java technologies: Struts, Hibernate framework, JSP and JavaScript.

- We have implemented jQuery Javascript framework for client-side features.

- We have used the CKEditor which is a Web-based WYSIWYG text editor.
Technologies Used

**Struts**
- a Model –View-Controller (MVC) design pattern based framework
- an open source Java framework and used to develop web applications

**Hibernate**
- a framework for mapping an object-oriented domain model to a traditional relational database
- an object-relational mapping (ORM) library for the Java language
Technologies Used

JQuery
- an open source JavaScript framework
- simplify the client-side scripting
Preliminary Work

- File Creation application

**Goal:** understand MVC architecture
get comfortable with Struts Framework
Preliminary Work

- Web-based IDE “Aurorasdk”
  **Goal:** to study the architecture of a web-based IDE

**Technologies Used:**
Google Web Toolkit (GWT), Servlet and MySQL

**Limitations:**
users require to do hand-coding
does not provide cross-browser compatibility
does not provide collaboration feature
Preliminary Work
Design

Features of our StrutsHib IDE

- Automatic setup of files, databases, and database tables
- Creating New Model or Controller
- Associating Models
- Interfacing Model and Controller
Design

- **Directory Structure** – Struts Application
Design

- **Struts Component**

  Action Class Files: stored under `com.webide.struts.action`

  Action Form Files: stored under `com.webide.struts.beans`

  `struts-config.xml`: placed under the `Web-INF` directory

  View Resources: placed under the `Web Root` directory
Database Design

- MySQL database name `web_ide` is used for StrutsHib IDE

- Database tables
  - Users (id, Name, Password, UserName)
  - Projects (id, projectname, projectpath, user_id)
  - Models (id, modelname, modelpath, project_id)
  - Models_Association (id, model_id1, model_id2, asso_type)
  - Controllers (id, controllername, controllerpath, project_id)
When a user creates a new project, three subfolders get created on the server.
- Model
- Controller
- View
Implementation

• When a user creates a new project, three subfolders gets created on the server i.e. Model, Controller and View. Returns the physical path on the server:
  `getServlet().getServletContext().getRealPath()`

• The folder is traversed stored in a string buffer:
  `Request.setAttribute("ProjectUserName", Buffer);`

• In client side values can be retrieved by using:
  `<%StringBuffervalue=(StringBuffer)request.getAttribute(" ProjectUserName"); %>`
Hibernate Instantiation

- SessionFactory is a immutable and thread-safe object for creating new Hibernate sessions

- Session is intended to last as long as the logical transaction on the database.

```java
static {
    try {
        configuration.configure(configFile);
        sessionFactory = configuration.buildSessionFactory();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```
Hibernate Initialization

- Implemented SessionFactory using Singleton pattern
- Lazy initialization is done

```java
public static Session getSession() throws HibernateException {
    Session session = (Session) threadLocal.get();
    if (session == null || !session.isOpen()) {
        if (sessionFactory == null) {
            rebuildSessionFactory();
        }
        session = (sessionFactory != null) ? sessionFactory.openSession(): null;
        threadLocal.set(session);
    }
    return session;
}
```
Struts Configuration

- Binding information for the different components of the Struts framework.

```xml
<action-mappings>
  <action name="UserForm" path="/User"
    type="com.webide.struts.action.UserAction" validate="true"
    input="Welcome.html">
    <forward name="success" path="/index.jsp"/>
  </action>
</action-mappings>
```
StrutsHib IDE

Web Based IDE

Please enter your username and password

Username

Password

Submit  Cancel

New User? Forgot Password?
StrutsHib IDE Main Page
Design Mode – Edit Model

- Right-click on model file gives an option to edit the model schema.

<table>
<thead>
<tr>
<th>Design Mode</th>
<th>Edit Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>integer</td>
</tr>
<tr>
<td>name</td>
<td>string</td>
</tr>
<tr>
<td>address</td>
<td>string</td>
</tr>
<tr>
<td>phonenumber</td>
<td>integer</td>
</tr>
</tbody>
</table>

Add more fields  Save Model Schema
Associating Models

- Models can be interfaced by creating associations between them
- hasMany, hasOne, and belongsTo are the association types
Middle Panel – Edit Mode

```java
public class TestProject {
    @Id
    @GeneratedValue
    private Integer ID;
    public Integer getId(){
        return ID;
    }
    public void setId(Integer ID){
        this.ID = ID;
    }
}
```
Add Project, Models or Controllers

- New project, model and controller can be created
- On clicking “Create” button, JavaScript will send an Ajax call to the controller along with user inputs.
- The controller will create the files on server and store the information in database.
Right Panel

- Interactive Help panel is associated with the user action.
Performance

![Performance Graph]

- **Load Time**
- **Project Creation**
- **Create File**
- **Open File**
- **Save File**

**Graph Explanation:**
- **Thousands**
- **Eclipse Time (ms)**
- **StrutsHib Time (ms)**
- **Aurorasdk Time (ms)**
Usability Testing

- Six users are asked to perform following tasks:

**Task 1.** Create a user account and login using username and password.  
**Task 2.** Create your new project ‘TestProject’ and browse the files structure created by the StrutsHib IDE.  
**Task 3.** Create a new model ‘TestModel’ under Model directory and open the file in the design mode.  
**Task 4.** Edit the model schema by adding new fields to it using design mode editor.  
**Task 5.** Create association between ‘TestModel’ model component and default model ‘TestProject’ using design mode editor.  
**Task 6.** Create interfacing between ‘TestModel’ model and controller component ‘TestController’ using drag and drop feature.
Usability Testing

Expert Users
Novice Users

Task 1  Task 2  Task 3  Task 4  Task 5  Task 6
Usability Testing

- Users found that our StrutsHib IDE simplifies the process of creating a project with its models and controller components.

- Users evaluated it against other IDEs and found it easy to use and requires less learning curve.

Observation

- Less hand-coding increases productivity.
Conclusion

• Achieved the goal of the project by implementing the IDE that automates the processes of creating Web application.

• Our StrutsHib web-based application is user-friendly and it has features which help the user in rapid development of their projects.

• Overall, users found our StrutsHib IDE easy to use.

• StrutsHib IDE can be enhanced to support team collaboration.
Questions?
Thank you