GSoC 2011 - DSpace WebMVC

Summary

Project	WebMVC (Freemarker) UI development
Student	Robert Qin Zhengquan
Mentors	• Graham Triggs (lead) • Peter Dietz • Stuart Lewis
Technologies	WebMVC (Freemarker) UI
Proposal	Melange, Original Proposal (pdf)
Location for project	• Main Project Code (Master Branch in GitHub) • Student Forked Code

Ongoing work for this project (after GSoC 2011) is at: WebMVC (Freemarker) UI

Project Description

Problem Statement: Separation of processing logic from presentation layer at the Web UI component in the DSpace application layer.

Problem Elaboration: Although the DSpace three-tier architecture follows the model-view-controller (MVC) architecture, with most of the logic encapsulated in the logic layer, there is still a fair amount of processing logic involved in the Web UI component at the application layer. For example, in the “full.jsp” file under the browse folder of DSpace JSP-UI, there is a thick chunk of scriplet logic embedded from lines 30 to 164.

Solution: *The current proposal is to adopt the *Spring Web MVC framework as a means to enforce the MVC framework architecture and the use of the Freemarker template engine as a means to enforce the amount of logic at the presentation layer.

Freemarker offers an alternative to JSP and JSTL tags in reducing scriplet logic via the use of more data model flexibility, providing the use of customized template snippets via #macro definitions and etc... Freemarker provides a lethal combination with the Spring MVC framework in reducing presentation layer logic by the following breakdown of goals:

Breakdown of goals within Spring MVC framework

Views: Freemarker restricts dynamic content display to the use of Expression Languages variables. ${output} or keys and certain logic/sequence operations via the use of its unique syntax that are ignored by the web browser. Eg: #if, #assign #list tags.

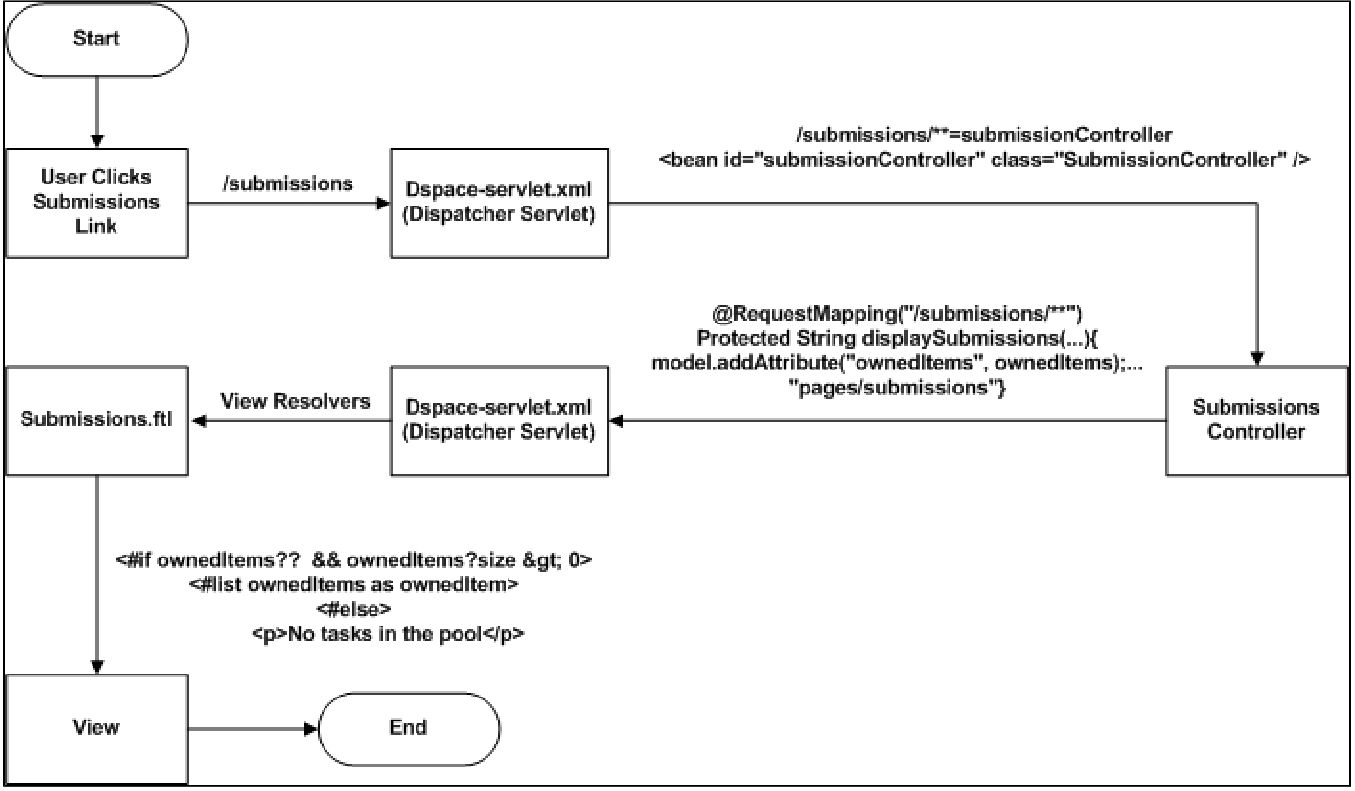
Dispatcher Servlet: Dispatcher servlets act as the "front gate" role in identifying urls, and delegating the appropriate controller class to handle the request via bean tags definitions.

Controller classes: To handle request urls from the dispatcher servlet, delegate the appropriate method/class method calls via the use of request /response annotations, correspond in parsing data values to the model component for logic processing, and delegating the appropriate response views. The full path url of the views will be automatically resolved by the dispatcher servlet by gearing it with the appropriate prefix and suffix.

Model and View: Represents a Class to represent a model and view returned by a handler used by a DispatcherServlet. The view can take the form of a reference to a View object, or a String view name which will need to be resolved by a ViewResolver object. The model is a Map, allowing the use of multiple data objects keyed by name. These keys will be utilize by the #assign syntax of Freemarker to provide dynamic content for views.

Model: To ensure that complex logic algorithms computation is handled entirely at the business logic layer. For example, session beans, web services.

Example Scenario: User clicks submission link


Start
User Clicks Submissions Link
/submissions
Dspace-servlet.xml (Dispatcher Servlet)
Submissions Controller
Dspace-servlet.xml (Dispatcher Servlet)
View Resolvers
Submissions.ftl
View
End

```
/submissions/**=submissionController
<bean id="submissionController" class="SubmissionController" />
```

```
@RequestMapping("/submissions/**")
Protected String displaySubmissions(...){
model.addAttribute("ownedItems", ownedItems);...
"pages/submissions"}
```

```
<#if ownedItems?? && ownedItems?size &gt; 0>
<#list ownedItems as ownedItem>
<#else>
<p>No tasks in the pool</p>
```

Project Timeline

Before April 20:

- Installation of DSpace with netbeans
- Initiate contact with the Mentor to confirm the specific milestones required in the project development
- Initiate contact with the Mentor on the specific skill sets required for this project
- To code a complete Spring Web MVC application with Netbeans
- To read up the DSpace API documentation

April 20 – May 23 (Before the official coding time):

- To remain in contact with the mentor via email and IRC and to clarify existing doubts in the process of learning and discovery about DSpace.
- Continue reading up on DSpace API documentation, especially the specific required components
- To re-model previous enterprise applications based on the Spring MVC framework
- Using of FreeMarker engine for presentation layer code generation
- Understanding the webmvc module
- Understanding GIT functionalities to commit code to GitHub.

May 23 – June 18 (Official coding period starts):

- Understanding the project objectives of the first phase deliverables
- Understanding webmvc module
- Coding on the first phase deliverables
- Assisting in webmvc system documentation
- Weekly pull request to commit forked copy from robertqin/webmvc to dspace/webmvc
- Recoding all logic from the JSP (Method call handlers) into relevant Servlets and applying the use of the Freemarker syntax.
- System documentation
- Reporting of any system bugs and errors on https://jira.duraspace.org/browse/DWMVC
- Weekly scrum with mentor on progress statuses including clarification on any existing doubts

June 18 – July 5:

- Progressive unit testing on the previous milestone deliverable
- Post implementation review of the previous milestone deliverable
- Coding on the next mile stone deliverable
- Weekly pull request to commit forked copy from robertqin/webmvc to dspace/webmvc
- Assisting in webmvc system documentation
- System documentation
- Reporting of any system bugs and errors on https://jira.duraspace.org/browse/DWMVC
- Weekly scrum with mentor on progress statuses including clarification on any existing doubts

JULY 6th MID TERM EVALUATION

July 6 – July 15:

- To prepare a presentation report to update developers on the developed functionalities, exceptions and bugs encountered
- Make further changes in the code to improve the Functionality, Exception handling, Bug Removal.
- Weekly pull request to commit forked copy from robertqin/webmvc to dspace/webmvc
- To maintain contact with the developers and to update them on the progress
- Consultation with the mentor on the next phase of milestones deliverables
- Assisting in webmvc system documentation
- Coding on the second phase deliverable
- Phase 2 system documentation

July 15 – July 25:

- To be in constant touch with the developers and to let them know about our progress.
- Progressive unit testing on the previous milestone deliverable
- Post implementation review of the previous milestone deliverable
- Coding on the next mile stone deliverable
- Assisting in webmvc system documentation
- Most of the time will be consumed for rigorous testing and bug fixes.
- Coding on phase three deliverables
- Phase 3 system documentation

July 25 – July 31:

- Post implementation review

A Buffer of two weeks has been kept for any unpredictable delay.

NOTE: Require webmvc documentation and update dspace 1.7.0 API library.

Student Background

Self description

I am a year 3 **Information Systems Student** specializing in **Service Science Systems** at the **National University of Singapore (NUS)**. I have previously been involved in two enterprise applications development projects.

1. **Online Tourist Portal**
2. **Online Video Rental Portal**
3. **Embedding of external SDK into iPhone/iPad apps**

Online Tourist Portal

In the online tourist portal project, my main scope of responsibilities involved the creation of a tourist map application that retrieves the various stakeholders' meta-data information and displays them via the use of the**Google Map API**. I was also involved in the creation of a tourist analytics platform that uses data-mining techniques to search through the various purchasing and customer demographics information and to identify various sales and customer behavioral patterns and present them to the end-user via the use of* JFree charts*.

Online Video Rental Portal

In the online video rental portal project, I took my enterprise application development education to the next level by embarking on the use of web services to encourage interoperability via the use of auto-generated WSDL and SOAP XML contracts. My scope responsibility was to create a platform for users to upload videos, add inventory, performed sales analysis of the various inventory items and to allow customers to watch online videos.

Extracts of my source code are available at the following links:

Online Tourist Portal: http://max.mailbigfile.com/aeefd9149fd1a61fd248e59f1f01c2b6/listFiles.php

Online Video Rental Portal: http://max.mailbigfile.com/2d5a34530f1a7d5fa8aacc3fb6154219/listFiles.php

Embedding of external SDK (Open source) into iPhone apps

I had a one year internship stint with a software development company in Beijing, China. I was mainly involved in project management to coordinate the work responsibilities between the developers and designers in the deliveries of an iPhone application that is embedded with our company's SmarTots API. Part of my responsibilities includes embedding the external API method calls (Local and Remote) into the developers' software code. I had to manually look through every developer's code to understand the specific portion to embed the required API calls.

The table below depicts my skills sets and competency level.

Technical Skill set	
Type	**Competency Level**
Java	**Advance**
HTML, CSS	**Beginner**

Javascript	**Intermediate**
JSP, Servlets	**Advance**
Google Maps API	**Beginner**
JFree charts	**Advance**
Enterprise Javabeans	**Advance**
Web services	**Intermediate**
XML	**Beginner**
C	**Beginner**
SQL	**Intermediate**
Development Environment Skill set	
Netbeans	**Advance**
Glassfish/Tomcat	**Advance**
Spring Web MVC Framework	**Beginner**
Freemarker	**None**
None Technical Skill set	
English (Written)	**Excellent**
English (Spoken)	**Excellent**
Chinese (Written)	**Average**
Chinese (Spoken)	**Average**
Presentation	**Excellent**

Reasons for joining this project

In my experiences of building enterprise applications (Tourist Portal and Video rental system) my JSP pages were often “peppered” with scriptlets containing certain business logic. For example, within the JSP page itself, my scriptlets would contain method calls, like (get methods) to ensure the required content is displayed on the page. JSP pages should not contain too much logic as it creates tight coupling scenarios and can create obstruction to web design works. For example, the combination of javascript, scriptlet logic and html code could complicate web design development at the front end layer. Scriptlet logic should best be confined within the Servlet and JSP should mainly contain HTML code.

I am interested in taking on this project because I have witnessed for myself how logic at the presentation layer could create tight coupling scenarios and obstruct the evolvement of the front layer coding, thus impairing the quality of my code. Above all, I wished to utilize the opportunity of the GSOC program to brush up on the quality of my software development skills by working with expert developers in the industry to develop real world applications. I strongly believe the experience gained from the GSOC program would put me in good standing with my future employers when I transit to the workforce upon graduation from NUS.