HTML Cockpit Specification Review Struts Actions 1.0 Component Specification

1. Design

As we progress in upgrading the Direct application, the next thing we are focusing on is the complete flow of launching a contest. This includes everything from entering the basic data for a contest to payment, spec review, provisioning, etc. We currently have a basic flow for launching a contest created in the service layer and being assembled for the HTML version of Direct. This contest will dig deeper into the process of launching a contest to address the requirements introduced and clarified by the requirements that are referenced in this contest.
This component provides the struts actions to start specification review, retrieve specification review scorecard for display and resubmit specification review scorecard.

1.1 Design Patterns

Facade pattern – Various services are used as facades of the back end.

MVC pattern – Struts Actions are the model and part of the controller in the MVC pattern.

DTO – ViewSpecificationReviewActionResultData and SaveSpecificationReviewCommentActionResultData are DTOs that hold result data.

Strategy pattern – Each action defines the concrete strategy in execute method used by Struts framework context.

1.2 Industry Standards

Struts 2.1.8.1, Spring 3.0

1.3 Required Algorithms

None.

1.4 Component Class Overview

SessionAwareAction
This abstract action provides the session injected by Struts built-in interceptor to the implementing classes.

SpecificationReviewAction
This abstract action provides common request parameters (contestId and studio) to the implementing classes. It also provides convenience methods to get the TCSubject from session and to store result data, exceptions and validation errors to the AggregateDataModel#data map of AbstractAction. The session key used to retrieve the TCSubject and the keys used to store result data, exceptions and validation errors in AggregatedDataModel#data map are all configurable via injection setter, though default values are already defined for them.

StartSpecificationReviewAction
This action is responsible for handling the request to start a specification review. It extends SpecificationReviewAction.

SpecificationReviewService and ContestServiceFacade are used to perform the request.

ViewSpecificationReviewAction

This action is responsible for handling the request to view a specification review. It extends SpecificationReviewAction.

SpecificationReviewService and SpecReviewCommentService are used to perform the request.

The result data is aggregated into ViewSpecificationReviewActionResultData.

SaveSpecificationReviewCommentAction
This action is responsible for handling the request to add or update a specification review comment. It extends SpecificationReviewAction.

SpecReviewCommentService is used to perform the request.

The result data is aggregated into SaveSpecificationReviewCommentActionResultData.

ResubmitSpecificationReviewAction
This action is responsible for handling the request to resubmit a specification review. It extends SpecificationReviewAction.

SpecificationReviewService is used to perform the request. Also, content should be configured using injection, as it is not a request parameter.

ViewSpecificationReviewActionResultData
This class is a simple DTO that holds the result data of ViewSpecificationResultAction.

It holds the retrieved specification review, its status and its comments.

SaveSpecificationReviewCommentActionResultData
This class is a simple DTO that holds the result data of SaveSpecificationReviewCommentAction.

It holds the saved user comment.

1.5 Component Exception Definitions

SpecificationReviewActionException
This exception is never created directly and serves as a top level exception for this component.

StartSpecificationReviewActionException
This exception is created and used by StartSpecificationReviewAction if any error occurs during action execution. It is not thrown by the action, but just stored in the model map as required by Struts Framework component.

ViewSpecificationReviewActionException
This exception is created and used by ViewSpecificationReviewAction if any error occurs during action execution. It is not thrown by the action, but just stored in the model map as required by Struts Framework component.

SaveSpecificationReviewCommentActionException
This exception is created and used by SaveSpecificationReviewCommentAction if any error occurs during action execution. It is not thrown by the action, but just stored in the model map as required by Struts Framework component.

ResubmitSpecificationReviewActionException
This exception is created and used by ResubmitSpecificationReviewAction if any error occurs during action execution. It is not thrown by the action, but just stored in the model map as required by Struts Framework component.

1.6 Thread Safety

This component is thread safe. Actions are mutable and not thread safe. But when mutable attributes are not modified after the initialization (i.e. when beans are managed by Spring container or request parameters are set by Struts) and entity instances passed to these classes by the caller are used in thread safe manner, these classes are also used in thread safe manner.

The business services used by this component are reasonably thread safe and are responsible for handling transactions.

DTOs are mutable and not thread safe. This is not an issue because each action instance will be used by one thread only, and will create/use only one DTO instance at a time. Also, this is not an issue at the client side because it is single-threaded.

Hence, this component is effectively thread safe.

2. Environment Requirements

2.1 Environment

Development language: Java1.5

Compile target: Java1.5

2.2 TopCoder Software Components

Base Exception 2.0 – is used by custom exception defined in this component.

Struts Framework 1.0 – provides AbstractAction and other classes used by this component.

Contest Service Façade 1.0 – provides a service facade and other classes used by this component.

Specification Review Service 1.0 – provides a service facade and other classes used by this component.

Spec Review Comment Service 1.0 – provides a service facade and other classes used by this component.

Security Manager 1.1 – provides the TCSubject class.

Project Service 1.2 – provides StudioCompetition and SoftwareCompetion classes.

Project Services 1.1 – provides the FullProjectData class.

Studio Service 1.3 – provides ContestData and other classes used in this component.

NOTE: The default location for TopCoder Software component jars is../lib/tcs/COMPONENT_NAME/COMPONENT_VERSION relative to the component installation. Setting the tcs_libdir property in topcoder_global.properties will overwrite this default location.

2.3 Third Party Components

Struts **2.1.8.1** (http://struts.apache.org/2.1.8.1/)

NOTE: The default location for 3rd party packages is ../lib relative to this component installation. Setting the ext_libdir property in topcoder_global.properties will overwrite this default location.

3. Installation and Configuration

3.1 Package Name

com.topcoder.direct.actions

3.2 Configuration Parameters

All parameters are configured using IoC (eg: Spring). Example configurations for both struts and Spring are shown below.

Sample application.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:context="http://www.springframework.org/schema/context"
    xmlns:tx="http://www.springframework.org/schema/tx"
    xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-2.5.xsd
                              http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context-2.5.xsd
                              http://www.springframework.org/schema/tx
http://www.springframework.org/schema/tx/spring-tx-2.5.xsd">

    <bean id="mockActionInterceptor"
        class="com.topcoder.direct.actions.MockActionInterceptor">
    </bean>

    <bean id="contestServiceFacade"
        class="com.topcoder.direct.actions.MockContestServiceFacade">
    </bean>

    <bean id="specificationReviewService"
        class="com.topcoder.direct.actions.MockSpecificationReviewService">
    </bean>

    <bean id="specReviewCommentService"
        class="com.topcoder.direct.actions.MockSpecReviewCommentService">
    </bean>

    <!-- =========================== -->
    <!-- action beans                -->
    <!-- =========================== -->

    <bean id="startSpecificationReviewAction"
        class="com.topcoder.direct.actions.StartSpecificationReviewAction">
        <property name="specificationReviewService" ref="specificationReviewService"></property>
    </bean>

    <bean id="viewSpecificationReviewAction"
        class="com.topcoder.direct.actions.ViewSpecificationReviewAction">
        <property name="specificationReviewService" ref="specificationReviewService"></property>
        <property name="specReviewCommentService" ref="specReviewCommentService"></property>
    </bean>

    <bean id="saveSpecificationReviewCommentAction"
        class="com.topcoder.direct.actions.SaveSpecificationReviewCommentAction">
        <property name="specReviewCommentService" ref="specReviewCommentService"></property>
    </bean>

    <bean id="resubmitSpecificationReviewAction"
        class="com.topcoder.direct.actions.ResubmitSpecificationReviewAction">
        <property name="specificationReviewService" ref="specificationReviewService"></property>
    </bean>
</beans>
```

Sample struts.xml

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!DOCTYPE struts PUBLIC
    "-//Apache Software Foundation//DTD Struts Configuration 2.0//EN"
    "http://struts.apache.org/dtds/struts-2.0.dtd">

<struts>
    <constant name="struts.devMode" value="true" />
    <constant name="struts.convention.action.disableScanning" value="true" />

    <package name="tcs-default" extends="struts-default"
        abstract="true">
        <!-- Setup interceptors stack -->
        <interceptors>
            <interceptor name="mock" class="mockActionInterceptor" />

            <interceptor-stack name="defaultTCSStack">
                <interceptor-ref name="mock" />
                <interceptor-ref name="defaultStack" />
            </interceptor-stack>
        </interceptors>

        <!--
            Make the default one used for all actions unless otherwise
            configured.
        -->
        <default-interceptor-ref name="defaultTCSStack" />

    </package>

    <package name="default" namespace="/" extends="tcs-default">

        <action name="startSpecificationReviewAction" class="startSpecificationReviewAction">
            <result name="input">/input.jsp</result>
            <result name="success">/success.jsp</result>
        </action>

        <action name="viewSpecificationReviewAction" class="viewSpecificationReviewAction">
            <result name="input">/input.jsp</result>
            <result name="success">/displaySpecReviewComments.jsp</result>
        </action>

        <action name="saveSpecificationReviewCommentAction"
class="saveSpecificationReviewCommentAction">
            <result name="input">/input.jsp</result>
            <result name="success">/displayComment.jsp</result>
        </action>

        <action name="resubmitSpecificationReviewAction" class="resubmitSpecificationReviewAction">
            <result name="input">/input.jsp</result>
            <result name="success">/success.jsp</result>
        </action>
    </package>
</struts>
```

3.3 Dependencies Configuration

See dependencies documentation.

4. Usage Notes

4.1 Required steps to test the component

Extract the component distribution.

Follow Dependencies Configuration.

GUI demo is required for this component, instructions are provided below on how to deploy it.

1. Download JBoss 4.0.2 and install it locally (http://www.jboss.org/jbossas/downloads/).
2. Set your JAVA_HOME to use JDK 1.5 as compiler.

3. Edit build-dependencies.xml and configure jboss.home and junit.jar path.
4. Execute ‘ant deployToJBoss’ within the directory that the distribution was extracted to in order to deploy the demo to JBoss.
5. For instructions on how to use demo, see section 4.3.

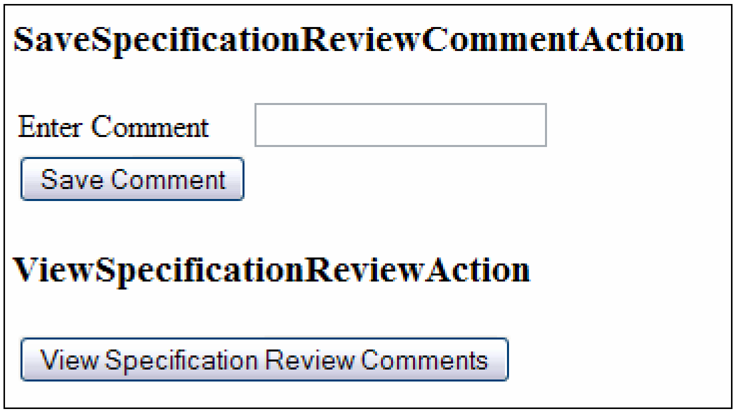
Execute ‘ant test’ within the directory that the distribution was extracted to.

4.2 Required steps to use the component

Please see the demo.

4.3 Demo

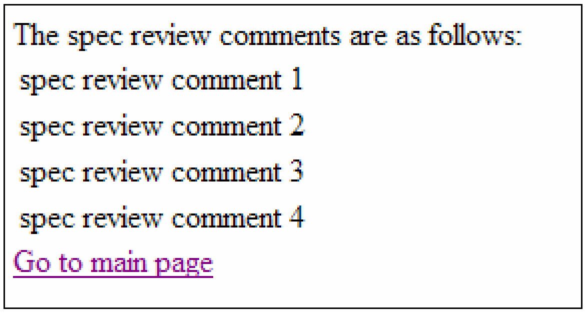
1. Be sure you have deployed demo to JBoss, refer to section 4.1 for details.
2. Start JBoss.
3. Navigate to http://localhost:8080/demo/input.jsp
   a. This assumes you use port 8080 for JBoss, you can modify this URL if you use some other port.
4. You will be presented with the following screen.



SaveSpecificationReviewCommentAction

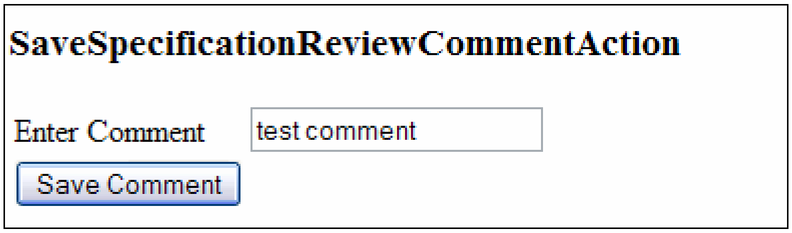
ViewSpecificationReviewAction

5. Press the “View Specification Review Comments” button to see the existing comments, you will get the following screen:



The spec review comments are as follows:
spec review comment 1
spec review comment 2
spec review comment 3
spec review comment 4

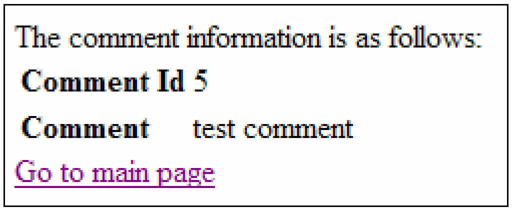
6. Click the “Go to main page” link to return to the main page.
7. In the “Enter Comment” textbox, enter “test comment”.



SaveSpecificationReviewCommentAction

Enter Comment test comment

8. Click the “Save Comment” button and you will be presented with the following screen:



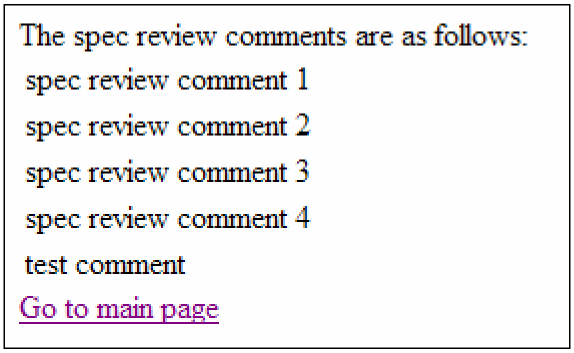
The comment information is as follows:

Comment Id 5

Comment test comment

9. Click the “Go to main page” link to return to the main page.

10. Click the “View Specification Review Comments” button to see the existing comments as well as the new comment that was just added:



The spec review comments are as follows:

spec review comment 1

spec review comment 2

spec review comment 3

spec review comment 4

test comment

4.3.1 *Action usage*

The following JSP was used for the input page. It utilizes the SaveSpecificationReviewCommentAction to save the comments and the ViewSpecificationReviewAction to display them.

Hidden values are used to configure contest id, studio, questionId and action request parameters. In this case, the specification review is about a software component (studio is false) and the comment is being added for the first time (action is “add”).

When the “Save Comment” button is clicked, SaveSpecificationReviewCommentAction is invoked and the configured result page will be displayed showing the new comment ID.

To view the comments, the “View Specification Review Comments” button can be clicked and the ViewSpecificationReviewAction will be used to get the comments. Refer to previous section for details.

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
        pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<%@taglib uri="/struts-tags" prefix="s"%>
```

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Demo</title>

<link href="<s:url value="/css/main.css"/>" rel="stylesheet"
        type="text/css" />

</head>
<body>

<!-- display any action errors -->
<div style="color: #ff0000;"><s:iterator value="actionErrors">
        <span><s:property escape="false" /></span>
</s:iterator></div>

<h3>SaveSpecificationReviewCommentAction</h3>
<s:form action="saveSpecificationReviewCommentAction">
        <table>
                <tr>
                        <td>Enter Comment</td>
                        <td><s:textfield name="comment" /></td>
                </tr>
                <tr>
                        <td><s:submit value="Save Comment" /></td>
                        <td></td>
                </tr>
        </table>
        <s:hidden name="contestId" value="1" />
        <s:hidden name="studio" value="false" />
        <s:hidden name="questionId" value="2" />
        <s:hidden name="action" value="add" />
</s:form>

<h3>ViewSpecificationReviewAction</h3>
<s:form action="viewSpecificationReviewAction">
        <table>
                <tr>
                        <td><s:submit value="View Specification Review Comments" /></td>
                </tr>
        </table>
        <s:hidden name="contestId" value="1" />
</s:form>

</body>
</html>
```

5. Future Enhancements

None.