GSoC 2011 - DSpace Submission Enhancements

Summary

Project	DSpace Submission Enhancements
Student	Gaurav Kejriwal
Mentors	• Mark Diggory • Scott Phillips
Technologies	Submission UI, etc.
Proposal	Melange
Location for project	• GitHub: https://github.com/gauravkl/dspace • JIRA: https://jira.duraspace.org/browse/DS-962 • Screenshots / Mockups: UI Mockup-Submission enhancements

Project Description

Of all configuration DSpace has, item-submission.xml is one which end-users would continuously want to be changing because different repository administrators would want different submission workflow .The logic to do so is simple enough that they could do it. But, being an xml file on the server gets in the middle of that. At the moment, repository admins can edit their metadata registries, bitstream format registries, etc. but still have to wait for the system administrator when they want to map a collection to a new workflow in item-submission.xml for workflow modification.
This project seeks to get rid of these xml files and make the equivalent with Database tables and User Interfaces for the end-user.

Original Proposal

Introduction:

This is a proposal to further continue the work on the topic on which I worked in GSOC in 2009 summer and take it to the point that it can be integrated and brought into the next version of Dspace.

In 2009 summer,I completed successfully the DB management of input-forms. My plan is to completely integrate it this time to Dspace and also further work on DB management of submission-workflow.

The wiki page of that project is at http://wiki.dspace.org/index.php/Google_Summer_of_Code_2009_Submission_Enhancements .

This will enhance the powers of the Administrator to manage their collection and reduce their dependency on the system administrator.

Problem:

Of all configuration DSpace has, item-submission.xml is one which end-users would continuosly want to be changing because different repository administrators would want different submission workflow.The logic to do so is simple enough that they could do it but being xml file in the server gets in the middle of that. At the moment, repository admins can edit their metadata registries, bitstream format registries, etc. but still have to wait for the system administrator when they want to map a collection to a new workflow in item-submission.xml for workflow modification.

Proposed Solution:

We must get rid of these xml files and make the equivalent with Database tables and User Interface for the end-user. At the moment Dspace uses these xml files to map different workflow for each collection. We can change it to allow administrators to do this via web UI. This would greatly reduce the need for system-administrators to touch DSpace installation . It would automate the cumbersome process that uses information which is much more accessible to the machine than to the human. I think that this should be an ordinary editable attribute of the collection, so that the Dspace administrator can do it without bothering/waiting for the site administrator.

Mark Diggory : We are currently preparing a contribution of the Configurable Reviewer Workflow for DSpace 1.8. This includes breaking up the Reviewer Workflow into Configurable Steps similar to the Submission Workflow. An Important Part of this project should be to adopt a similar strategy for the Submission Workflow where Custom Steps and Actions can be defined in a Spring Configuration and registered within a "Workflow Service". Database Persistence of Submission Workflow should involve capturing the relationships between these Steps and Actions. The Reviewer Workflow Does Not attempt this because its configuration of the defined workflow steps still happens in an xml file.

For solving the problem of modification of submission workflow we can have a *'Edit Workflow Page' *where the admin can select the list of steps and their order(priority) to be taken for that collection through a UI. Here, also we can give an option for admin to upload the desired workflow in xml format which will ultimately be updated in the database.

1. We will need to define a Domain Model for representing the Submission Workflow and its steps in such a way as to allow it to be persistable in the database. An initial Model would have JAVA classes defined in Spring Workspace Service.
    a. Step: A physical Step with logic currently defined in Java code.
    b. Action: One or more Actions are the possible outcomes of the execution of the step and represent Tasks that may be completed prior to the Item transitioning to another step.
    c. Role: A Group of Individuals that are alerted to and participate in the completion of a Step.
    d. UserSelectionMethod: A View that allows the user to interact with the Step Executing Actions.
2. Reviewing the Workflow, we find that the primary aggregate root of the Domain is a configuration that arranges the relationships between these 4 Domain objects.

```
<step id="{step.id}" nextStep="{next.step.id}" userSelectionMethod="{user.selection.bean.id}" role="{role.id}" >
   <!-- optional alternate outcomes, depending on the outcome of the actions you can alter the next step here --
>
   <alternativeOutcome>
      <step status="{integer}">{alternate.step.id}</step>
   </alternativeOutcome>
   <action id="{action.bean.id}"/>
   <action id="{action.bean.id.1}"/>
</step>
```

An initial User interface should provide the user with an ability to select

1. a. One or more steps that will be in a workflow
    b. Associate Specific Outcomes (status codes) With existing Workflow Steps, Identifying the paths through the workflow.
    c. Add/Remove EPeople/Groups from Roles associated with the Step
2. An advanced User Interface may provide
    a. the ability to alter the Configuration of what Actions are Associated with the Step
    b. codes those action may generate.

Other changes which I am interested to work on to improve the submission process is enabling tool-tip on submission fields.

Mark: The Inputforms work done in the previous GSoC project should be reevaluated and presented to the duraspace-gsoc list so that we can be refamiliarized with it. There may be significant recommendations in this area that emerge from GSoC 2011 - DSpace SKOS Authority Controls Project. More specifically, we may want to begin to rethink the level at which authorities and other sources like Controlled Vocabs and Lists are associated with the Metadata Field. This might include creating a metadatafieldtype Domain model that is assigned to the metadatafield allowing the presentation to be refined to just be skinning one or more metadatafields with forms/widgets int eh User Interface.

I am open to ideas from the mentors to mould the project goals in a better manner. Apart from configuring the submission-workflow we can look for changes in the submission process which may ease the whole submission process for the end-user.

I followed the discussions which happened on the devel-mailing lists since a few days on this topic and agree with the point that the genreal UI changes need to be separate than the more concrete structural changes .It can be decided within the community what will be better to work on in scope of 3 months and I will be willing to go with that.

I will conduct a Usability survey on the Dspace-mailing list to find the common pains users' face during the submission process in Dspace.Also I will go through the dspace-mailing lists archives to see the common problems faced by the users during the submission process.

List Of tasks :

I'll achieve the database management of workflow by moving in a parallel framework to that of**metadata schema registry.**

I'll move the item-submission from file based storage to the database and make them manageable via the UI.

I'll be creating separate db tables for each purpose like tables for storing the processing class and JSPUI binder.

For doing this in JSPUI interface I'll have to create a JSP file which will diplasy the form so that

the admin can enter the various properties like JSPUI binder,processing class,steps.order etc.

I can write servlets with code to extract the values entered by the admin in these forms which can be used by functions of other classes.

I can write classes to extract the workflow configuration from the database .

When the user will start the submission process the workflow steps will be extracted from DB and performed in that order.

I'll also include functions for sanity checks to ensure that the user doesn't pass null values for elements and its attributes and to prevent other such faults.

For doing this for XMLUI interface,I'll develop in a similar framework to that which is present for editing a metadataschema registry or creating a new one. I'll create classes similar to the editmetadata schema & metadataregistryMain class in the authorizations aspect which'll generate the DRI document of the desired format. The XMLUI interface will be given more consideration as I believe XMLUI is the primary interface which will be used more in the future.

Timeline:

First I plan to start with completing the integration of the integration task of “input-form DB management “.I will take the suggestions of the community by May end .I will conduct a Usability survey on the Dspace-mailing list to find the common pains users’ face during the submission process in Dspace and will discuss the basic project goals with the mentor accordingly.

I will end the integration task by June end.After this I will start the “DB configuration of workflow”.I will complete the API by midterm evals and start the work of UI development after that.

I will complete the UI development related work by June end after which I will work to remove bugs and to integrate the work to Dspace main branch.

I will give weekly status updates to my mentors.

I have no other obligations this summer till August as I will be joining MS (Computer Science) course at **Texas A&M** university this fall. So I am totally determined to make this project a success and take it to the point of integration into the base Dspace code.

We will want to review this timeline in duraspace-gsoc to assure it is tractable. I believe that the tasks outlined are a bit ambitious and we want to use caution in fixing them prior to having a mentor/student dialog about goals of the project.

Rough Design -

Here I am outlining the rough sketch of design which I arrived at for the project.

I studied the implementation ways of the upcoming Configurable reviewer workflow and made a design with a similar flow.

So,the major difference with the reviewer workflow will be that we will pe persisting the workflow steps in the DB here eliminating the need of item-submission.xml

For this preliminary draft I am currently assuming we have collection based submission-workflow though I am looking for suggestions as to how to extend it for making type-centric(like having multiple workflows for a particular collection based on content type) and dynamic workflows.
I am a little unclear on if we need the concept of "roles" in case of submission-workflow as there will be only one submitter.

I welcome feedback from the community on how can this design be modified to make it better.

Basic Architecture of API--

SubmissionProcess-

A SubmissionProcess object will represent one submission process generated from the DB.
and will contain a number of submission steps that the workspace-item should go through.
Multiple collections can use the same submission process
and for those, the same SubmissionProcess object will be used .
The main responsibility of the SubmissionProcess object will be managing its steps

Step-

A Step will be a child element of the SubmissionProcess object and will contain a number of actions that will sequentially be executed by the submitter once the workspaceitem goes through the step.Each step has the following important properties:

actionConfigsMap: The actionconfigsmap contains a mapping of the action-identifiers used in the spring config file to the actual object which performs the execution of that action.
outcomes: The step also contains a list of alternative outcomes. In case one of the actions in the step returns an outcome different from 0, this list will be used to lookup the next step to activate.

SubmissionActionConfig-

SubmissionActionConfig objects will be children of Step object. The actions are the different tasks that can be executed as part of one Step.These are responsible for the execution of the different tasks in that step.

The submission framework will use these SubmissionActionConfig object to separate the user interface classes from the API classes.
Therefore, each SubmissionActionConfig object contains an Action object and a property that indicates whether the Action requires interaction with the user interface. The Action object is the API part of the action and is responsible for the actual execution of the action.

Action

An Action object will be responsible for the actual execution of the task for which it is responsible.
The Action object will a child of a SubmissionActionConfig object.

SubmissionFactory

The SubmissionFactory will be responsible for managing the different submission-process objects
that are being used by the different collections in DSpace.
The SubmissionFactory will contain a mapping from collection handles or the "default" options to SubmissionProcess-objects.
Each SubmissionProcess object will represent one submission-process containing a number of different steps.
The SubmissionFactory will contain all the required functions for loading the API bean classes that are required by
the Submission framework.

Architectural Design - UI

SubmissionXMLUI Factory

The SubmissionXMLUIFactory will be responsible for creating the required UI actions that are used by
the submission framework.The WorkflowXMLUIFactory will have a method that will create an instance of the required
class by using the Spring configuration file .

Action Interface

Each object returned by the SubmissionXMLUIFactory will be an implementation of the ActionInterface object .
These objects will used by the Transformer in order to
generate the different user interface pages that are used by the submission-framework.

Apart from these there will be classes to generate the UI for allowing of Editting the submission-workflow by
the administrator.Similar to the "Edit Metadata Registry" there will be options to edit the various attributes
of any particular submission-process like the steps it will contain and their order.

Configuration-

We will be needing 2 spring configuration files for mapping out the specific Processing classes and UI
generation classes for a particular action-id.

API config-

This configuration file describes the different Action Java classes that will be used by the submission framework.
This config file will contain Spring configuration.

This file contains the beans for the actions and user selection methods referred to in the workflow.xml. In order for the workflow framework to work properly, each of the required actions must be part of this configuration.

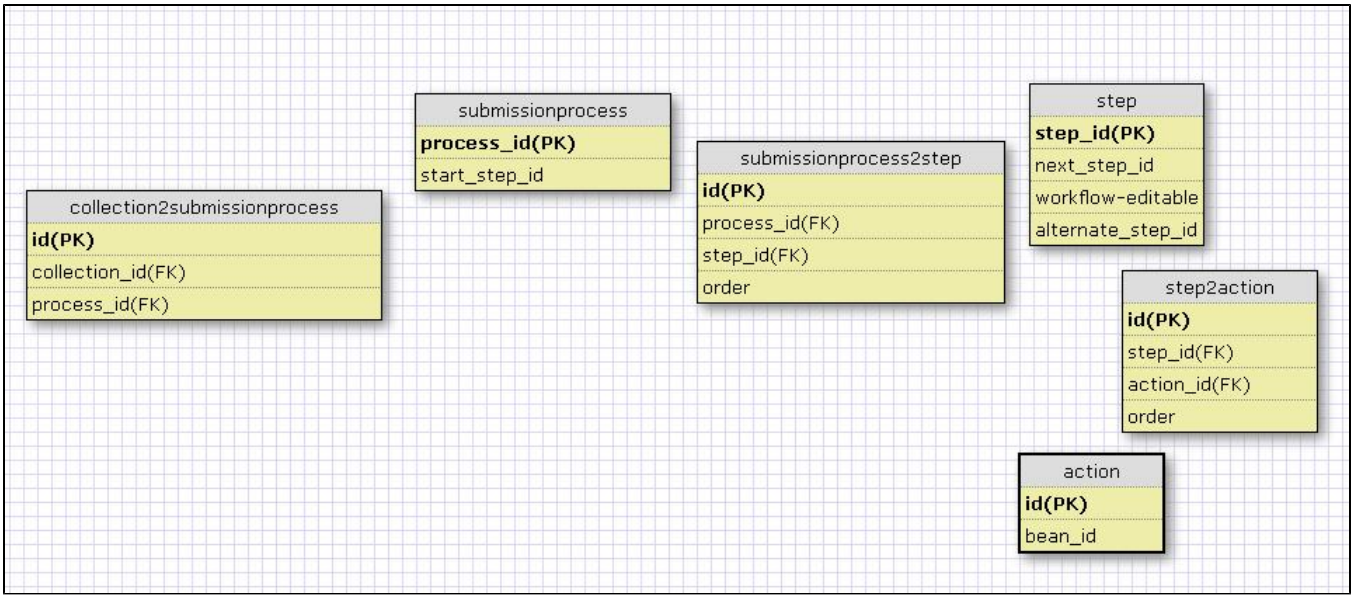
```
<bean id="{action.api.id}" scope="prototype"/>
   <bean id="{action.api.id.2}" scope="prototype"/>

<!-- Below the class identifiers come the declarations for out actions -->

<bean id="{action.id}" scope="prototype">
    <constructor-arg type="java.lang.String" value="{action.id}"/>

    <property name="processingAction" ref="{action.api.id}"/>
    <property name="requiresUI" value="{true/false}"/>
</bean>
```

UI configuration -

Each bean defined here will have an id which will be the action identifier and the class will a classpath which links to the xmlui class
responsible for generating the User Interface side of the submission-action.
In case an action requires a User Interface class, the workflow framework will look for a UI class in this configuration file.

```
<beans
   <bean id="{action.id}" scope="prototype"/>
   <bean id="{action.id.2}" scope="prototype"/>
</beans>
```

Brief Outline of DB schema-



The image above depicts a brief sketch of the DB schema which I arrived at.
We will need many new tables and a few of the existing ones like the "workspaceitem" will need to be modified.

1.submission-process--

This will represent a particular submission-workflow.A new entry in this table will
be created when the admin will create a new submission-process via the UI which will be present in the "Administer" options.

The columns will be-
1.process_id(Primary Key)
2.start_step_id(Foreign key)

2.step--

This will represent a particular step in the submission-workflow.
The columns will be-
1.step_id(PK)
2.next_step_id
3.workflow-editable(Boolean)
4.alternate_step_id

3.action-

This will represent a particular action which occurs in a step.
The columns will be -
1.action_id
2.bean_id(which can be referred to in the Spring configuration for the Processing and UI layout class)

4.collection2submissionprocess

This will be a mapping table for mapping a collection to the submission-process to be used for it.Its columns will be
1.collection_id(FK)
2.process_id(FK)

5.submissionprocess2step

This will be a mapping table for mapping a submission-process to a step to be used for it along with the order.
Its columns
will be
1.process_id(FK)
2.step_id(FK)
3.order

6.step2action

This will be a mapping table for mapping a step to an action along with the order.
Its columns will be
1.step_id(FK)
2.action_id(FK)
3.order

Apart from these some existing tables like the "workspaceitem" will need to be modified.

In "**workspaceitem**" we will need to add the following columns-

1.action_id--this will denote the action the submission process has reached till now.
2.step_id--the step which contains this action
3.process_id -- the workflow process which contains the step.

Experience and background:

I am an engineering graduate from Institute Of Technology, BHU, Varanasi which has been recently converted to an IIT. I participated in 2009 for Dspace in Google Summer Of Code . My project was "Submission Enhancement".I feel the project was very much successful and I completed the DB management of input-forms. The wiki page of the project is at http://wiki.dspace.org/index.php/Google_Summer_of_Code_2009_Submission_Enhancements.

The code is also available in the Dspace SVN repository at: https://scm.dspace.org/svn/repo/sandbox/gsoc/2009/kejriwal/

Last year also I worked on Dspace at Hasselt University library and contributed to the Oceandocs project being run by UNESCO. I also did a project on Dspace in summer of 2008 under Dr.ARD Prasad(member of Dspace Governance board) ,Indian Statistical Institute(ISI),Bangalore for Food & Agricultural Organization(FAO),United Nations(UN).The work was related to the customization of DSPACE & making it compliant to the AGRicultural Information System (AGRIS) metadata standard set by FAO.The work involved knowledge of JAVA,JSP,XML,XSLT. I did the task of automatic ARN generation(ARN is Automatic Resource Number defined by FAO which is set for every AGRIS record) and also developed a XML Parser in Java to convert WebAgris to DSpace input format which could help in export and import of AGRISAP metadata.

More information on the project can be found at http://code.google.com/p/dspace-agrisap/

As I have got much experience of working on Dspace I am pretty much familiar with the inner architecture of Dspace and it won't take much time for me to get initiated. I also have 9 months of work experience as a flash-developer at a social gaming startup -Oxylabs in India where I worked on developing flash based isometric game to be launched on Facebook platform.I will be leaving the job by May beginning as I will join TAMU this fall. I have received SUN Certification in JAVA Programming and cleared the exam with 95% marks. I have also done an internship at Deutsche Telekom Labs ,Ben Gurion University,Israel in the field of data-mining which included coding in JAVA.

Contact

email- gaurav.kl@gmail.com

skype -gaurav_kl