

Google Summer of Code 2008 Collection Workflow

Title: New collection workflow system

Student: De Schouwer Bram

Mentor: Lieven Droogmans

Co-Mentors: Bram Luyten, Richard Rodgers, ???

Abstract

In the current DSpace implementation, a workflow system is present that allows content submitters to ingest an item in the repository, that can be reviewed by simply accepting the submitted content, changing some of the submitted metadata or rejecting the item so it can be altered by the submitter. The current collection workflow system contains up to three steps for the submission of files. Except for the metadata changes, there are no possibilities for changing any of the submission content during the different steps of the workflow system.

In more demanding situations these few steps and editing functionalities can be insufficient to suit the requirements of the repository managers. Therefore the workflow system should be transformed into a series of different steps which can be individually defined, and can be configured on a per collection basis. It should be possible to configure the workflow system as a modular system. This way, the repository managers can choose the number and order of different workflow steps for each collection and configure each of these steps for its specified function. Also, it should be possible to change more and different content in each step of the submission. This means that it is possible to define steps in which it is possible to change not only the metadata but also for example the submitted files. This way the repository manager can maintain some interaction with the submitter during the submission process. Changes to the submitted files or info by the submitter can be requested between different steps of the process. This approach gives a lot more freedom and can result in better submissions that are optimally suited for the repository.

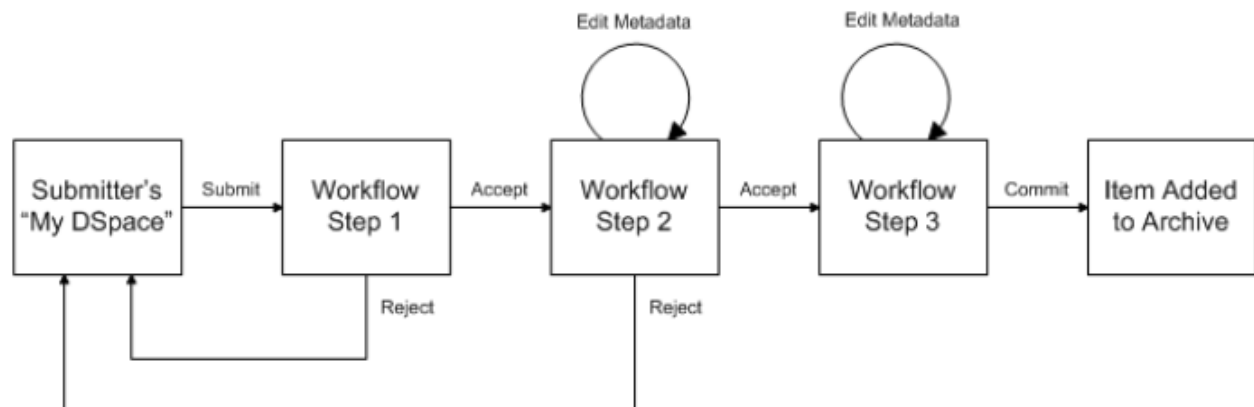
An example of a more elaborate workflow is a peer reviewing workflow in which reviewers are invited who each have to review the given item, so that afterwards the appropriate manager can either approve or decline an item based on the individual reviews. Differences in the editing possibilities could be that a submission initially just contains a short description and some metadata. After the approval of these contents, extra files and information can be added to the submission and the reviewing process can continue.

Of course the existing functionalities of the workflow system will be preserved. These contain the notification of the groups assigned to a step in the system, editing control and many more. The new system will be completely backwards compatible with the current.

Broad Goals

Current workflow system

In the current DSpace implementation, a workflow system is present that allows content submitters to ingest an item in the repository, that can be reviewed by simply accepting the submitted content, changing some of the submitted metadata or rejecting the item so it can be altered by the submitter. The current collection workflow system contains up to three steps for the submission of files. Except for the metadata changes, there are no possibilities for changing any of the submission content during the different steps of the workflow system.



Workflow Step	Possible actions
1)	Can accept submission for inclusion, or reject submission.
2)	Can edit metadata provided by the user with the submission, but cannot change the submitted files. Can accept submission for inclusion, or reject submission.

3)

Can edit metadata provided by the user with the submission, but cannot change the submitted files. Must then commit to archive; may not reject submission.

Modular workflow

Therefore the workflow system should be transformed into a series of different steps which can be individually defined, and can be configured on a per collection basis.

It should be possible to configure the workflow system as a modular system.

- repository managers should be able to choose the number and order of different workflow steps for each collection
- repository managers should be able to configure each of these steps for its specified function
- it should be possible to change more and different content in each step of the submission.
 - possible to change not only the metadata but also for example the submitted files.
 - it should be possible to maintain some interaction with the submitter and/or reviewers during the submission process.
- Changes to the submitted files or info by the submitter can be requested between different steps of the process.

Example workflow

An example of a more elaborate workflow is a **peer reviewing workflow** in which reviewers are invited who each have to review the given item, so that afterwards the appropriate manager can either approve or decline an item based on the individual reviews.

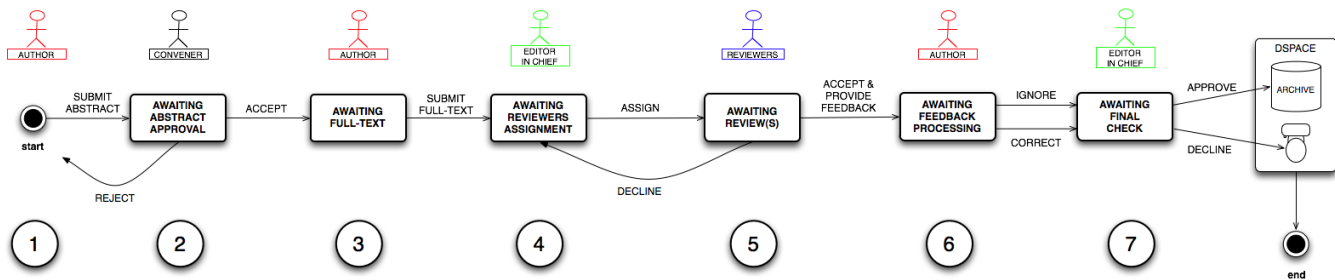
Differences in the editing possibilities could be that a submission initially just contains a short description and some metadata.

After the approval of these contents, extra files and information can be added to the submission and the reviewing process can continue.

Of course the existing functionalities of the workflow system will be preserved.

These contain the notification of the groups assigned to a step in the system, editing control and many more.

The new system will be completely backwards compatible with the current.



Goal Breakdown

Modular workflow system

Different steps

- Different basic categories of steps
 - Metadata edit
 - Acceptance/rejection
 - File edit
 - Insert feedback
 - Assigning reviewers
 - Assigning submissions to reviewers
 - Submission
 - Abstract
 - Complete submission
 - Remove item
 - Archivation of an item
 - ...
- Steps can be combinations of steps
 - Looped
 - Combined functionalities (Example Metadata edit + acceptance/rejection)

Different roles

- Submitter/Author
- Editor in chief
- Reviewer
- ...

Different actions

The actions that can be carried out are defined by the step configuration.

Combine the different parts

- Assign the necessary role to each step.
- Make a full workflow system available as a combination of steps.
- Carry out the necessary actions after each step

Assumptions

- One step can only be carried out by one role.

Configuration

Configuration for a simple collection workflow (submit/accept-reject/accept-reject-edit metadata):

```
<actions>
  &#09;<action id="editMetadata" name="Edit Metadata" />
  &#09;<action id="acceptRejectAbstract" type="acceptReject" />
  &#09;<action id="uploadFullAbstract" type="uploadDocument" />
  &#09;<action id="uploadFullText" type="uploadDocument" />
  &#09;<action id="..." type="..." />
</actions>

<roles>
  &#09;<role id="submitter" name="Author" />
  &#09;<role id="convener" name="Convener" />
  &#09;<role id="editorInChief" name="Editor-in-chief" />
</roles>

<workflow>
  &#09;<step id="StartSubmission" name="Submit paper">
    &#09;&#09;<action id="editMetadata" />
    &#09;&#09;<action id="uploadAbstract" />
  &#09;</step>
  &#09;<step id="acceptRejectAbstractStep" name="Awaiting abstract approval">
    &#09;&#09;<>
  &#09;</step>
</workflow>
```

Example configuration for an elaborate (peer-reviewing) collection workflow:

```

<actions>
  &#09;<action id="editMetadata" type="EditMetadata">
    &#09;&#09;<inputforms value="default" />
    &#09;&#09;<continuationaction id="uploadAbstract" />
  &#09;</action>
  &#09;<action id="uploadAbstract" type="uploadDocument">
    &#09;&#09;<uploadrequired value="true" />
    &#09;&#09;<continuationaction id="acceptLicense" />
  &#09;</action>
  &#09;<action id="acceptLicense" type="acceptLicense">
    &#09;&#09;<nextstep id="acceptRejectAbstractStep" />
  &#09;</action>
  &#09;<action id="uploadFullText" type="uploadDocument" />
  &#09;<action id="assignReviewersForItem" type="assignRoleMembers">
    &#09;&#09;<selectFromRole value="reviewer" />
    &#09;&#09;<addToRole value="assignedReviewer" />
    &#09;&#09;<assignmentMethod value="manual" />
    &#09;&#09;<assignmentMethod value="SomeClass.biddingAlgorithm()" />
  &#09;</action>
  &#09;<action id="acceptAbstract" type="accept" />
  &#09;<action id="returnAbstract" type="return" />
  &#09;<action id="commitToArchive" type="archive" />

</actions>

<roles>
  &#09;<role id="submitter" name="Author" />
  &#09;<role id="metadataeditor1" name="Metadata Editor (1)" />
  &#09;<role id="metadataeditor2" name="Metadata Editor (2)" />
  &#09;<role id="metadataeditor3" name="Metadata Editor (3)" />
  &#09;<role id="reviewer" name="Reviewer" />
  &#09;<role id="assignedReviewer" name="Assigned Reviewer" />
  &#09;<role id="convener" name="Convener" />
  &#09;<role id="editorInChief" name="Editor-in-chief" />
</roles>

<workflow id="default">
  &#09;<step id="StartSubmission" name="Submit paper" role="submitter">
    &#09;&#09;<action id="editMetadata" nextStep="acceptRejectAbstractStep"/> <!-- list of actions to be chosen
    from -->
  &#09;</step>
  &#09;<step id="acceptRejectAbstractStep" name="Awaiting abstract approval" role="metadataeditor1"
  defaultNextStep="editMetadataStep">
    &#09;&#09;<action id="acceptAbstract" nextstep="EditMetadataStep" /> <!-- list of actions to be chosen from -->
    &#09;&#09;<action id="returnAbstract" nextstep="StartSubmission" />
  &#09;</step>
  &#09;<step id="editMetadataStep" name="Awaiting abstract approval" role="metadataeditor2">
    &#09;&#09;<action id="editMetadata" /> <!-- list of actions to be chosen from -->
    &#09;&#09;<action id="acceptAbstract" nextStep="..." />
  &#09;</step>
  ...
  &#09;<step id="assignReviewers" name="..." role="convener">
    &#09;&#09;<action id="assignReviewersForItem" nextstep="reviewFullText" /> <!-- list of actions to be chosen
    from -->
  &#09;</step>
  &#09;<step id="reviewFullText" name="Awaiting abstract approval" role="assignedReviewer">
    &#09;&#09;<action id="editMetadata" /> <!-- list of actions to be chosen from -->
    &#09;&#09;<action id="uploadDocument" />
    &#09;&#09;<action id="acceptFullText" waitforAll="assignedReviwer" nextStep="..." />
  &#09;</step>
  &#09;<step id="commitArchive" name="Add to archive" role="convener">
    &#09;&#09;<action id="commitToArchive" />
  &#09;</step>
</workflow>

<workflow id="123456789/6546">
</workflow>

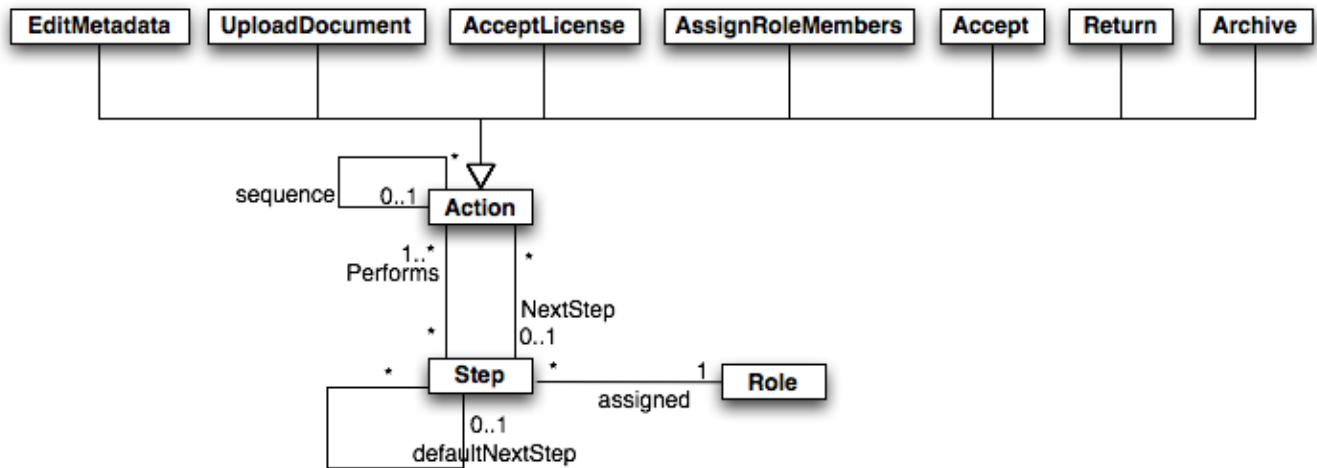
```

User interface for the usage of the system

- Does not include the configuration setup
- Interface for executing the actions required for each step

Design

Domain model analysis



Action

- These are the basic building blocks of the new workflow system
- Different types of actions
- Can be performed in a step
- Can be linked to a next action for the same user to create a sequence of actions
- Can be linked to a next step

Role

- Needed to perform the action(s) connected to a step
- Granted to users
- Connected to a particular step type

Step

- Part of the workflow system
- Can be looped
- Step combinations are restricted by certain constraints
 - Example: it makes no sense to combine a reviewingstep and an acceptance step with the reviewstep as the last one in the sequence
- Different categories of steps as described in the domain model analysis
- Each step defines a role or number of roles
- Can contain a sequence of actions

Detailed Design

WorkflowPart

- Generalization of Actions and Steps
- Interface used for common parts in the workflow system
 - Selection of the next step/action (depending on the current action and eventually a choice for an action)

Action

- The execution of actions will be implemented using the [Command design pattern](#)
- Action types
 - Actions to choose which action to perform next

- Have different next actions or steps
- Actions to perform a single operation
 - Make it possible to perform an operation
 - Has a pointer to a next action or a next step

Step

- Have a group of users connected to them
- These users can perform the actions linked to this step
- Contains methods for informing the connected users
- Contains a sequence of Actions

Workflow

- Contains the different steps

WorkflowItem

- Will be guided through the different actions
- Will contain a current step
- In the case that a sequence of tasks is part of the step and different epersons can perform the same sequence of actions, there will be a mapping eperson-action to the state of the WorkflowItem for each of these epersons.
 - Example: A review step which contains a group of 10 epersons but expects only 5 reviews. The first 5 epersons who claim the reviewingtask will be able to perform the sequence of actions related to this review step. Since each of these epersons can perform these actions on different moments, the map will contain information about the progress of each of the reviewers.

WorkflowFactory

- Implemented using the Factory Method design pattern
- Used to create actions, steps and workflows
- Will look up the right action type in the configuration xml

WorkflowManager

- Guides the workflow system
- Uses a request to lookup the right action and performs this action
- Contains all nessecary database methods

Implementation

- JAVA 6.0
- DSpace 1.5 with manakin

Open Issues

- Backwards compatibility with old workflow system (when upgrading to the new workflow system)
- Supported DSpace versions
- Web interface workflow configuration
- What about imported items? (how to specify what to do with batch imported items)
- Names for steps and roles should be configured in Messages.properties. The xml should only contain the keys for Messages.properties
- Assign people to role on level of collection, community, repository
- ...

References and Other Research

- Existing open-source workflow engines
 - ...
 - ...
- ...

Project timeline

JUNE

- Domain model analysis
- Start of detailed design

JULY

- Detailed design part 2
- Implementation

AUGUST

- Implementation

Results

Technical info

- The project code has been submitted to the dspace gsoc repository in the collection-workflow branch
- The code is based on a dspace 1.5 implementation
- The SQL commands to perform the necessary database changes can be found in the dspace/etc/collection-workflow-changes.sql file
- The configuration xml describing the used workflow system can be found in the dspace/config/workflow.xml file

Achievements

- The current workflow system has been implemented using the new workflow system
- The system can be configured using a configuration xml
- New types of actions can be added
- Order and number of actions can be configured as needed
- Notification emails can be sent using different email templates
- A basic interface to perform each action has been implemented

Further development

- More action types could be added
- The interface for the existing actions could be improved