SYLI

An external workflow system for DSpace

Samu Viita & Janne Pietarila

Structure of the presentation

1. What it SYLI
2. Motivation
3. Features
4. Demo
5. Tech
What is SYLI

- External tool for librarians to manage submission, workflow and collections in DSpace
- Especially useful in shared repositories (=one DSpace instance is used by several organizations)
- Has its own user and workflow database and connects to DSpace via SimpleREST interface
Motivation
Motivation to implement SYLI

- National Library of Finland hosts DSpace based repositories for over 40 public organizations
- Organizations have diverse needs especially in submission processes
- These needs are hard to fulfil with DSpace build in features
Doria repository contains almost 70,000 items:
- Theses
- Scientific publications
- Digitized heritage materials:
  - Books
  - Images
  - Maps
  - Audio
  - Sheet music
- Reports
- Catalogs
Problems in Doria

- Community admin role offered by DSpace is not enough
  - Cannot access user management features -> need help from the main administrators (us)
  - Cannot add field to metadata registry -> need help...
  - Cannot define submission forms by themselves -> need help...
- Problems with workflows: Defined in every collection explicitly, no inheritance from the upper level
Theseus contains over 70,000 full text items, mostly thesis.

- Same submission process and metadata for all organizations
- Thesis submitter use input form once in a lifetime
- Heavily tailored, external input form
- Over 200 submissions per day in rush seasons

Shared repository Theseus
Problems in Theseus

- Local librarians still use DSpace tools and need help in many situations like when new collection is added.
- Workflow doesn’t include edit metadata step since it doesn’t support multilingual metadata (e.g. `description.abstract fi`).
  - Has to edit item after it has been published.
- External input form is not easily customizable or utilizable in other situations.

...so we decided to do generic external submission and workflow system for dspace based on our customers needs...
Commonly needed features - requirements for SYLI

- Input form configuration improvements
  - Support for multilingual metadata:
    - e.g. description.abstract (fi, sv, en)
  - Input form definition improvements:
  - Copy fields: e.g. dc.contributor -> dc.publisher
  - Define constant fields
  - Licence definitions
  - Localization of input instructions and hints
  - Automatically adds metadata field to metadata registry, if field is missing
  - No need to restart DSpace when new input definition is made

- No community-collection separation, just collections
- Inheritable role and workflow definitions at any level in collection tree
Syli features
Syli input forms

- Defined with custom XML format
- Format inspired by DSpace input_forms.xml
  - Introduces new tags to implement new features
  - Includes localisation
Input forms XML: <field>

<field>
  <schema>dc</schema><element>description</element><qualifier>abstract</qualifier>
  <label>description_abstract_KEY</label>
  <multiple>true</multiple> true | false
  <required>false</required> true | false
  <input_type>text</input_type> line | text | personname | title | dropdown |
                        date | onki_ysa | onki_puho | onki_afo | onki_tero
  <language>*</language> * | - | <language code>
  <hint>description_abstract_hint_KEY</hint>
</field>
Input forms XML: dropdown

<field>
  <schema>dc</schema><element>contributor</element><qualifier>other</Qualifier>
  <label>contributor_other_KEY</label>
  <multiple>true</multiple>
  <required>false</required>
  <input_type value_pairs_name="contributor">dropdown</input_type>
  <language>*</language>
  <hint>contributor_other_hint_KEY</hint>
</field>
Input forms XML: dropdown

```xml
<field>
  <schema>dc</schema>
  <element>contributor</element>
  <qualifier>other</qualifier>
  <label>contributor_other_KEY</label>
  <multiple>true</multiple>
  <required>false</required>
  <input_type value_pairs_source="http://example.com/contributor.xml">dropdown</input_type>
  <language>*</language>
  <hint>contributor_other_hint_KEY</hint>
</field>
```
Input forms XML: <copy_field>

<copy_field>
    <from>
        <schema>dc</schema><element>organization</element><qualifier></qualifier>
    </from>
    <to>
        <schema>dc</schema><element>publisher</element><qualifier></qualifier>
    </to>
</copy_field>
Input forms XML: <constant_field>

<constant_field>
    <value>Our publishing company</value>
    <to>
        <schema>dc</schema><element>publisher</element><qualifier/>
    </to>
</constant_field>
Input forms XML: <fetch_field>

<fetch_field>
    <from>http://urnsource-kk.lib.helsinki.fi/cgi-bin/urn_generator.cgi?type=nbn</from>
    <to>
        <schema>dc</schema><element>identifier</element><qualifier>urn</qualifier>
    </to>
</fetch_field>
Input forms XML: <attr_field>

<attr_field>
  <schema>dc</schema><element>programme</element><qualifier></qualifier>
  <up>0</up> top | <non negative integer>
  <attr>short_description</attr>
</attr_field>
Roles

- Anonymous
- Member
- Editor
- Administrator
Workflow: simple

Draft → Publish

Delete

Published

Deleted
Workflow: normal

- Draft
  - Submit
  - Reject
  - Delete

- Pending
  - Accept
  - Delete

- Published

- Published
  - Publish
Workflow: complex
Demo
Tech
Syli is implemented using:

- Python
- Grok (based on Zope)
SimpleREST resource URLs (1 / 2)

/rootcommunities
/community/{communityId}
/community/{communityId}/logo
/community/{communityId}/communities
/community/{communityId}/collections
/collection/{collectionId}
/collection/{collectionId}/logo
/collection/{collectionId}/items
SimpleREST resource URLs (2 / 2)

/item/{itemId}
/bundle/{bundleId}
/bitstream/{bitstreamIdDotFormat}
/metadataschemas
/metadataschema/{metadataSchemaId}
/metadatfields
/metadatfield/{metadataFieldId}
SimpleREST code

https://github.com/anis-moubarik/SimpleREST

(Syli code is still located in our local version control system, but it will be published in GitHub.)
Thank you!

Contact information:

samu.viita@helsinki.fi
janne.pietarila@helsinki.fi