Developing publishing process support system with Fedora and Orbeon Forms - A case study

Matti Varanka, Ville Varjonen, Tapio Ryhänen

University Oulu Library, Finland

08.07.2010

Table of Contents

1. Introduction
2. Autogeneration of XForms editor from XML metadata
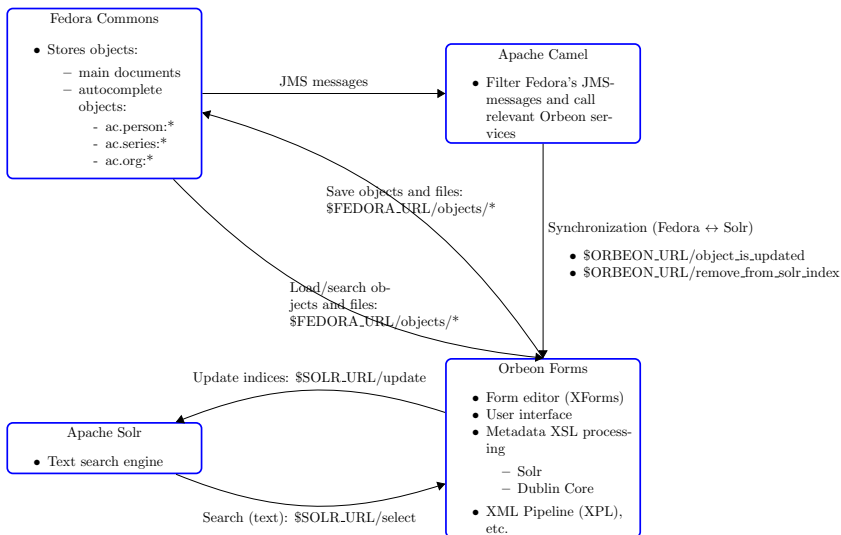3. Autocompletion of metadata blocks

Introduction

Motivation

In University of Oulu, doctoral theses are mainly published on the local Acta Universitatis Ouluensis series.

This process is complex:

- many people:
    - dissertant,
    - series editor,
    - press,
    - secretary, etc.

- lots of metadata,

- lots of communication,

- multiple file versions.

$\rightarrow$ Some kind of supporting software for this process is needed.

Complex publication process (in Finnish, sorry!)
Rectangle=document, oval=people/operation.

Searching for good platform

During development, following requirements arose:

- support for hierarchical metadata,
- support for multiple languages,
- versioning of the metadata and files,
- ability to link multiple documents to each other,
- support for author, series, and organization identifiers.

$\rightarrow$ Fedora.

Convergence towards Fedora & Orbeon Forms

Used architecture is almost XRX (XForms - Rest - ~~XQuery~~XSLT):

Lines represents REST-calls between components

About XForms

In this presentation, we focus on the metadata input process.

We are using heavily XForms with Orbeon Forms.

XForms:

- XForms 1.1 W3C Recommendation (October 2009),
- allows to create complex input forms and other interfaces in XHTML,
- almost a programming language,
- metadata can be stored and processed in XML (no conversions!).

Pros & cons of used software

Pro:

- rapid creation of UI and metadata editors with Orbeon Forms,
- interoperability between components is easy with REST-style data transfers,
- XML is used thoroughly in whole system,
- modularisation and expandability.

Con:

- steep learning curve,
- XForms documentation is not yet complete,
- XML is used thoroughly in whole system,
- some unanswered questions:
  - edit locking,
  - authorization,
  - heavier processing,
  - different roles.

Autogeneration of XForms editor from (almost arbitrary) XML metadata

Editor for complex metadata?

The amount of metadata for doctoral thesis is quite large:

- information about dissertants, custos, reviewers, other people,
- dissertation event, series information,
- publication info.

The used metadata format is hierarchical and fine grained to avoid comma-separated values.

Complexity of metadata
$\rightarrow$ Form editor is complex
$\rightarrow$ Autogenerate editor from given XML metadata.

Generating XForms editor

We created an XSL transformation to generate XForms editor from given metadata:

Metadata XML

```
<metadata>
  <person>
    <first_name/>
    <last_name/>
  </person>
</metadata>
```

XSL
$\longrightarrow$

XForms code

```
<xhtml:head> ...  <xforms:model>
  <xforms:instance id="metadata">
    <metadata xmlns="">
      <person>
        <first_name/>
        <last_name/>
      </person>
    </metadata>
  </xforms:instance>
</xforms:model> ...
</xhtml:head><xhtml:body>
...
  <xforms:input ref="instance('metadata')/first_name">
    <xforms:label>first_name</xforms:label>
  </xforms:input>

  <xforms:input ref="instance('metadata')/last_name">
    <xforms:label>last_name</xforms:label>
  </xforms:input>
...
</xhtml:body>
```

Orbeon
Forms
$\longleftarrow$

Metadata editor

Choose Form Language English ▾

person

person:	Add
first_name:		Add
last_name:		Add

Example: Dublin core editor

Metadata

```
<rdf:RDF xmlns:rdf=...>
  <rdf:Description rdf:about="...">
    <dc:creator/>
    <dc:creator/>
    <dc:title ouf:type="textarea"/>
    <dc:description
       ouf:type="textarea"
       ouf:html_editor="true"/>
    <dc:date ouf:type="xs:date"/>
  </rdf:Description>
</rdf:RDF>
```

Example: Dublin core editor

Metadata

```
<rdf:RDF xmlns:rdf=...>
   <rdf:Description rdf:about="...">
      <dc:creator/>
      <dc:creator/>
      <dc:title ouf:type="textarea"/>
      <dc:description
         ouf:type="textarea"
         ouf:html_editor="true"/>
      <dc:date ouf:type="xs:date"/>
   </rdf:Description>
</rdf:RDF>
```

Features:

1. Multilanguage

2. Repeating fields

3. Different input fields

4. HTML editor

Features, cont.

Metadata

```
<metadata xmlns=...>
  <person>
    <first_name/>
    <ouf:label elementname="first_name">
      First name
    </ouf:label>
    <last_name/>
    <ouf:hint elementname="last_name"
      lang="eng">Display name is generated
      automatically
    </ouf:hint>
    <xforms:bind ref="//display_name"
      calculate="concat( //last_name, ', ',
      //first_name)"/>
    <display_name calculate="true"/>
    <organization show="false">
      <org_name/>
    </organization>
  </person>
  <publication>
    <pub_title/>
  </publication>
</metadata>
```

More features:

1. tabs,

2. translations,

3. calculate values,

4. helps & hints,

5. hide/show blocks.

How about more complex metadata?

The created XSL can take (almost) arbitrary XML and create a somewhat usable metadata editor. For example, <u>XMetadiss</u>:

Autocompletion of metadata blocks

Autocompletion

Some of the inputted data repeats itself in many cases:

- organizations,
- series,
- personnel.

Because our metadata is fine-grained, each entity contains many fields

   $\rightarrow$ data input is tedious

      $\rightarrow$ why not automate this?

Autocomplete example

1. Blocks (ie. organization) can be autocompleted from drop-down list,

2. List is created from Fedora objects (ac.org:*),

3. Whole block is replaced,

4. Updated data can be submitted back into the Fedora object.

Autocomplete example

1. Blocks (ie. organization) can be autocompleted from drop-down list,
2. List is created from Fedora objects (ac.org:*),
3. Whole block is replaced,
4. Updated data can be submitted back into the Fedora object.

Autocomplete example

1. Blocks (ie. organization) can be autocompleted from drop-down list,

2. List is created from Fedora objects (ac.org:*),

3. Whole block is replaced,

4. Updated data can be submitted back into the Fedora object.

Autocomplete example

1. Blocks (ie. organization) can be autocompleted from drop-down list,

2. List is created from Fedora objects (ac.org:*),

3. Whole block is replaced,

4. Updated data can be submitted back into the Fedora object.

Implementation of autocomplete

Metadata (Fedora object, e.g. thesis:2)

Fedora object ac.org:23

```
<publication_metadata>
<pub_info>
  ...
    <pub_title>Test publication</pub_title>
  ...
</pub_info>
<creator_info>
  ...
  <person_org_info ac_choice="ac.org:23">
    <fedora_id>ac.org:23</fedora_id>
    <org_id org_id_schema=""/>
    <org_univ_name lang="eng">University of Oulu</org_univ_name>
    <org_univ_name lang="fin">Oulun yliopisto</org_univ_name>
    <org_faculty_name lang="eng">Faculty of Science</org_faculty_name>
    <org_faculty_name lang="fin">Luonnontieteellinen
tiedekunta</org_faculty_name>
    ...
    <org_address show="false">
      ...
      <street_name lang="eng">P.O.Box 3000</street_name>
      <post_number lang="eng">FI-90014</post_number>
      ...
    </org_address>
  </person_org_info>
  ...
</creator_info>
<thesis_info>...</thesis_info>
</publication_metadata>
```

1. Metadata contains all information about publication,

2. Each autocomplete block is also stored as a separate object in Fedora by using xforms:submit-method.

Autocompletion

Benefits of block-level autocomplete:

- reduces typing,
- reduces errors,
- publication contains the publication-time information,
- object database contains the newest information,
- e.g. organization lists are easy to generate.

Conclusion

- Currently, the input framework and the basic functionality is implemented.
- Rapid development with Orbeon Forms & Fedora
  - steep learning curve,
  - small amount of code,
  - XML-based,
  - good interoperability.
- Generation of XForms editor
  - fast prototyping,
  - allows to use fine-grained and complex metadata,
  - editor can have sophisticated features,
  - block-level autocomplete,
  - hopefully the source code for this autogeneration will be released on some day on our wiki:
    https://wiki.oulu.fi/display/UusiJultika/Home.

That's all, folks!

Thank you!

Questions?