Building flexible workflows with Fedora, the University of York approach

Julie Allinson and Yankui Feng

Abstract

In 2008, the University of York embarked on a project to build a multimedia Digital Library underpinned by Fedora Commons. In the long-term, the York Digital Library (YODL) plans to meet not only multimedia requirements, but multi-disciplinary, institutional, multi-user and multiple access control needs. In order to do this, we needed a flexible, scalable approach to fulfil the following three strands of our roadmap:

- An ‘administrative’ workflow, including metadata creation forms, automatic extraction of metadata and data/resource transformation for images, video, music, audio and text resources to be extensible as new resource types are identified.
- A self-deposit workflow for non-administrative users to deposit to YODL, White Rose Research Online (WRRO) and other targets as appropriate.
- Bulk ingest tools and procedures, to include a desktop deposit tool.

This paper will outline current and future work at York which builds on Fedora Commons, initially drawing on the Muradora interface and access control layer with a SWORD-enabled simple deposit tool in development and future plans for making this more flexible with Mura-independent applications.

Requirements

From the outset, YODL has been designed for multimedia. This brings with it, not only a range of (ever-growing) resource types, but also a variety of metadata requirements. In the initial phase of the project we have focussed on still images and, in particular, on the needs of our History of Art Department. From an analysis of requirements, spanning across resource types, usage, access control requirements and metadata, we devised a ‘content model’ [1], a document which lays out all of our requirements and decisions around images from which the workflow could be developed. Current use of Muradora [2], coupled with some of the bespoke work we have undertaken based around Muradora, means that we are tied to Fedora 2.2.4 currently. Although a version of Mura is available for Fedora 3x, we have taken the decision to build a new interface and de-couple workflow elements from Muradora by late 2010. Current work on content models is being done with the future use of Fedora content models in mind.

Taking images as an example, the requirements analysis concluded that we should recommend a set of common image media types with the promise that these will be processed fully, a set of other known image types which will be treated as images but without additional bespoke processing, and the promise that anything else would be stored without processing. There is more on the technical development of this tripartite workflow below. Regarding metadata, it was clear from speaking with users that Dublin Core is not sufficiently rich to meet the needs of multimedia searching. For History of Art and Archaeology, for example, location is crucial. In some cases this refers to the repository or gallery location of the artwork, in others to the site of an archaeological dig, or the location of a piece of architecture. Dublin Core in its simple form (that used as a base metadata format in Fedora) cannot capture these distinctions. This left us with a decision to make about metadata formats. There were essentially three options: create our own local bespoke schema, create a Dublin Core application profile or build on the existing Dublin Core application profile for images, or use VRA Core 4. After surveying each option we selected the latter, for three main reasons: firstly, it is a standard format and will thus increase the re-usability and interoperability of our metadata, secondly, it has been designed by experts and meets the needs of images well and, thirdly, it is fully documented with an XML Schema in place.

In order to keep the creation of different workflows manageable, we are aiming to use bespoke metadata format for each major content type (or content model) and to keep these to a relatively small number. Currently these are images, audio and unsupported (our catch-all for any other resources). In the medium term, we will be adding workflows for collections, video and theses, with the possibility of a small number of additional content models.

Technical implementation

Muradora has a submission wizard to facilitate the process of creating new digital objects. Users with appropriate permissions can create a new object in three steps: (1) selecting parent collection and object content model, (2) uploading/specifying resources, and (3) entering metadata. However, the drawbacks of Muradora's submission workflow cause problems in it being used in a production environment, particularly for multimedia resources. In the current Muradora deposit workflow, the depositor has to wait while uploading files as uploading is the pre-step of entering metadata. Therefore, the current Muradora workflow is not efficient especially when uploading large files. As a result, two separate asynchronous processes for uploading/processing resources and submitting metadata would be a better choice in terms of efficiency and performance. Bespoke workflow is another requirement for specific media types and users. Continuing with images as an example, some depositors have agreed to a small 'Preview' image being made public whilst restricting the full sized image for University users, and in another case very large archival quality TIFF images must not made available to users but need to be stored with the object. Therefore, a preview image should be generated from the original image when an image is submitted, which is not implemented in muradora's workflow.

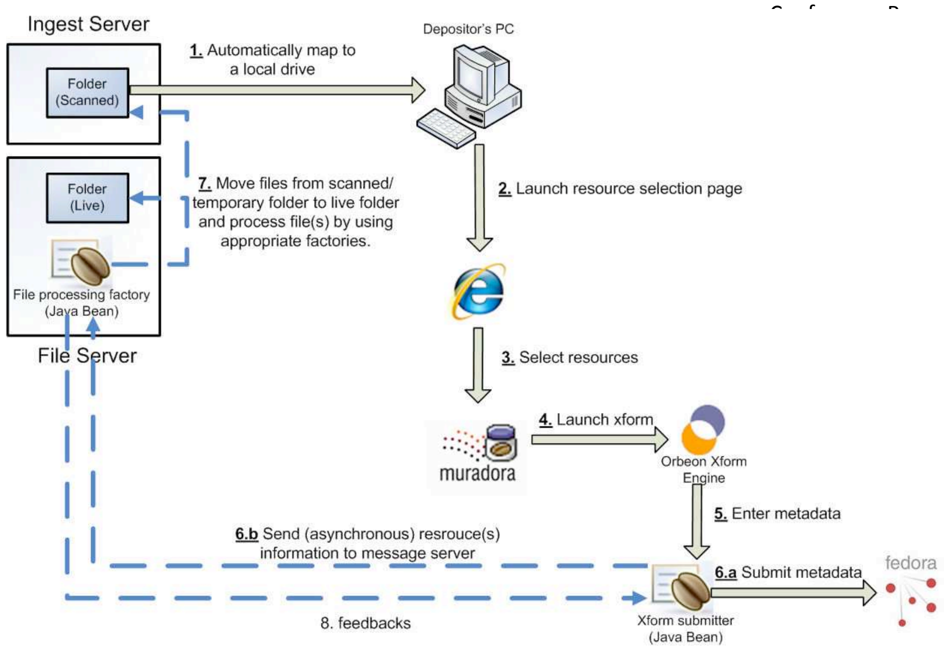
In summary, the new work flow is able to deposit in a more efficient way and is able to deposit any type of file, which can be divided into three categories as shown below:

Fully supported files (e.g. TIFF/JPEG images, WAV audio files, and ISO CD/DVD images): the corresponding processing for each type of file is defined individually in the workflow. For example, a TIFF image file is transformed to a full-size JPEG file, to a preview JPEG file, and to a thumbnail JPEG file. The original TIFF image and all three generated image files are ingested into Fedora as data streams.

Partly supported files (e.g. BMP/PNG images): for these files, generic processing logic is defined. For example, GenericImage for any declared partly supported images, GenericAudio for any declared partly supported audio files.

Unsupported files (e.g. AVI file for now): for these files, a more generic ('Generic of generic') process is defined. For example, when an AVI file is selected, the file is ingested into Fedora as a data stream under a pre-defined fixed name and a pre-defined thumbnail image is used for any unsupported file.

As shown, to support the asynchronous deposit process, an ingest server can be used by University wide users as a temporary storage for resources to be ingested into YODL. Depositors can specify resources via various ways, e.g. select resources from a mapped drive of ingest server in their own PC, or upload resources from their local drive, or point to a URL either as 'redirect' or 'external' links. All resources are mapped to a URL and are ready for ingestion. Based on the content model and editor selected by the depositor, the appropriate metadata entry form is launched. These forms use the XForm [3] technology. Currently, a VRA [4] XForm editor has been developed for images and a customized MODS editor is under development for audio. After submitting an XForm, VRA metadata is saved into Fedora directly and transformed into Dublin Core; RELS-EXT and RESL-INT datastreams are also created. At the same time, an asynchronous process is used to process pre-prepared

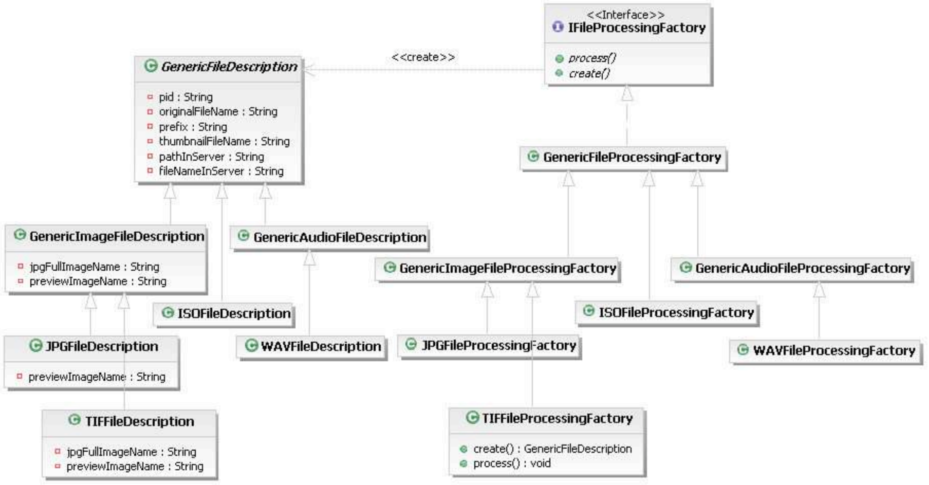

Ingest Server
Folder (Scanned)
1. Automatically map to a local drive
Depositor's PC
Folder (Live)
7. Move files from scanned/ temporary folder to live folder and process file(s) by using appropriate factories.
2. Launch resource selection page
File processing factory (Java Bean)
File Server
3. Select resources
4. Launch xform
muradora
Orbeon Xform Engine
5. Enter metadata
6.b Send (asynchronous) resrouce(s) information to message server
6.a Submit metadata
fedora
8. feedbacks
Xform submitter (Java Bean)

resources. A message containing resource details is sent to YODL server, where a program is running to process all resources, e.g. following an appropriate work flow for a specific resource, and ingest these resources into Fedora server as data streams.

As YODL is expected to support more and more file types in specific ways, it is desirable that the workflow is reusable when the support of a new file format is required. The ideal development scenario when the workflow is asked to deal with a new file format (mime type) is:

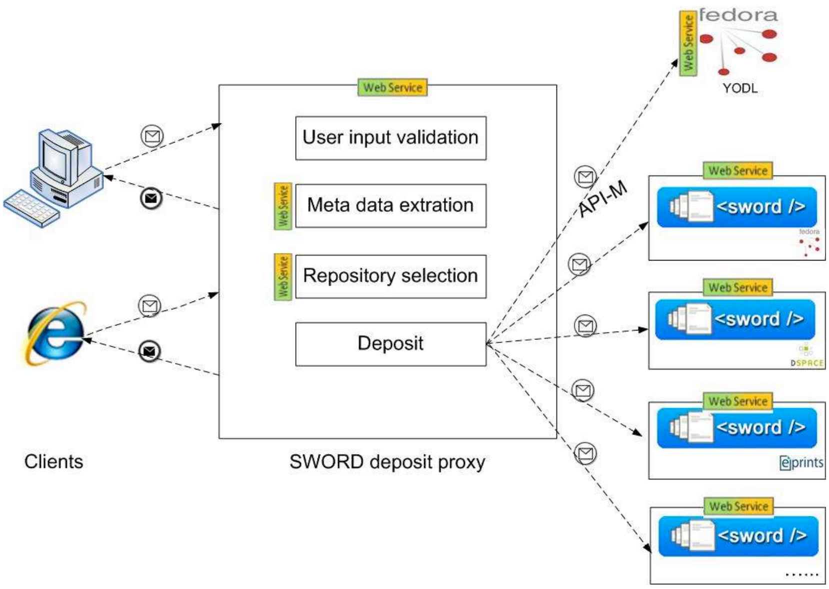
- Add new code to deal with the new file format
- Modify related configuration file(s)
- No need to modify any existing code

As shown, factory design patterns [5, pp.87] is used to maximize the reusability of existing workflow. A matched factory is used to process each file type. Basically, these factories implement the processing logic for each file type. For example, TIFFileProcessingFactory defines the processing logic for TIFF images, e.g. transform TIFF image to JPEG image, and generate preview and thumbnail images. Currently, only a few file types have their customized factories, e.g. JPG, TIFF, ISO, and WAV. It is impossible to have a specific factory for each file type. Therefore, some generic factories are designed to process a general category of files, e.g. GenericImageFileProcessingFactory for generic images including BMP, PNG, and GIF, and GenericAudioFileProcessingFactory for generic audios such as MP3. In addition, a more generic factory namely GenericFileProcessingFacotry is used to process all other file types.


<<Interface>>
IFileProcessingFactory
process()
create()
<<create>>
GenericFileDescription
pid : String
originalFileName : String
prefix : String
thumbnailFileName : String
pathInServer : String
fileNameInServer : String
GenericFileProcessingFactory
GenericImageFileDescription
jpgFullImageName : String
previewImageName : String
GenericAudioFileDescription
GenericImageFileProcessingFactory
GenericAudioFileProcessingFactory
ISOFileDescription
ISOFileProcessingFactory
JPGFileDescription
previewImageName : String
WAVFileDescription
JPGFileProcessingFactory
WAVFileProcessingFactory
TIFFileDescription
jpgFullImageName : String
previewImageName : String
TIFFileProcessingFactory
create() : GenericFileDescription
process() : void

Next steps

Currently in development is a SWORD-based deposit tool. The first iteration of this tool will be ready by summer 2010 and will offer a simple way to deposit a range of item types into YODL or White Rose Research Online (WRRO). The tool will work by allowing users to select a particular content type and upload files. The tool will then pre-select a repository (in the first iteration YODL or WRRO) and



offer a simple metadata creation form. Once metadata is completed, the tool will package up the content and send to the selected repository. In technical terms, the main development will be a SWORD deposit proxy, a client application which will perform a set of actions on a deposit, e.g. file validation, metadata collection and extraction, repository selection, packaging and deposit to specified repository. This is illustrated in the accompanying diagram.

Future iterations of this tool will (if feasible) add new repository targets and implement shibboleth to authenticate users. A batch upload tool, based on the same underlying technology is also currently being specified with a dual-purpose of offering users a simple way to upload batches of resources, whilst offering advanced features to assist repository administrators in loading batches.

Conclusions

The University of York is comparatively small, with a relatively small development team. The Digital Library has a long-term goal to create a range of workflows to meet different requirements. We have a long way to go, but can already say that our repository can accept any content and can perform bespoke actions on a small cohort of content types. In the near future we will integrate with our WRRO partner repository, extend to self-deposit and offer a bespoke workflow for music. Reflecting on the topic of 'the grand integration challenge', York Digital Library is indeed working on integration of workflows: the integration of local and institutional, of subject and cross-disciplinary, and of bespoke and general.

References

1. Content Model for Images https://vle.york.ac.uk/bbcswebdav/xid-448730_3
2. Muradora http://www.muradora.org/
3. XForms http://www.w3.org/MarkUp/Forms/
4. Gamma, E., Helm, R., Johnson, R., Vlissides, J. "Design Patterns: Elements of Reusable Object-Oriented Software" (2001), Addison-Wesley, ISBN: 0-201-63361-2.
5. VRA core 4.0 http://www.vraweb.org/projects/vracore4/