Enhanced E-theses Project

Deliverable 9

Managing compound objects within Fedora

Document Description

Project	
Title:	Enhanced E-theses
Start date:	1st December 2007
Funding Agency	Knowledge Exchange
Document	
Deliverable number:	D9
Deliverable title:	Managing compound objects within Fedora
Actual Date of Delivery:	January 2009
Author:	Chris Awre (University of Hull)
Editor(s):	Martin Slabbertje (Utrecht University Library)
Workpackage:	WP9
Workpackage title:	Management and preservation of complex digital objects within Institutional Repositories
Workpackage leader:	University of Hull
Version/Revision:	1.1
Draft/Final:	Final
Total number of pages: (including cover)	10
File name:	Enhanced Etheses project_deliverable 9_v1.1.doc

Section 1: Fedora infrastructure

Introduction

The Fedora digital repository system originated in the mid 1990s with an academically-based project to explore how digital content could be organised, if you were starting from a blank sheet. The focus was on the organisation of any type of digital content, both simple and rich, and recognised at an early stage that the management of such content required a flexible and open approach. An underpinning element of the Fedora architecture, therefore, is the digital object model that determines how digital objects can be organised and managed within the same system. This model encompasses both the content itself and the metadata describing it on an extensible basis; as circumstances change over time requirements for describing digital objects may change but you may not wish to lose how it was previously described. The ability to record relationships between digital objects, and to a lesser extent between different parts of an individual digital object, is also built into the digital object model.

In considering how compound ETDs can be managed within repository systems from a Fedora perspective it is necessary to understand the digital object model the system uses. It provides the basis for being flexible as the structure of digital objects evolves and can, given the trade-off between what you would like to do and what is practicable with available resources, provide options for addressing the current need. This section will consider this model and other aspects of current Fedora infrastructure and functionality that can be applied to compound objects. I am grateful to the information freely available via the Fedora wiki for informing this document.

Fedora digital object model

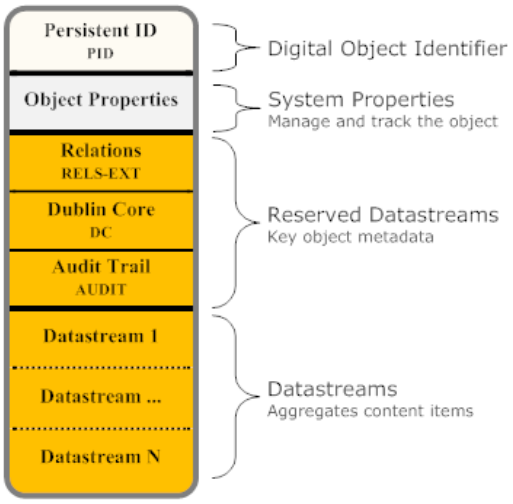

Figure 1: Fedora digital object model

As can be seen from Figure 1, the Fedora digital object model is made up of a container with four constituent parts, with each part also having its own elements. These are described below. The whole digital object model is expressed using XML.

Persistent ID – As the name says, an identifier that is persistent, and also unique, within the repository. This can be assigned automatically by the repository or user-defined.

Object properties – System-defined properties about the object to allow the repository to manage and track it.

Reserved datastreams –

- **RELS-EXT** (Relations): A datastream that is specifically used to express the relationships between digital objects, using RDF as the language of expression. A default set of relationships is available, though this can be refined or extended as required.
- **DC** (Dublin Core): A datastream that records a basic Dublin Core record for the digital object, but which is primarily aimed at use for system purposes rather than for user-oriented functionality (though it can be used for this purpose and some Fedora clients (e.g., Muradora) use it by default).
- **AUDIT** (Audit trail): A datastream that maintains a record of all changes to a digital object. This is maintained by the system automatically and cannot be edited. Examples include information on versioning, checksum changes and logs from batch processing.

Datasteam 1-n – Individual datastreams hold the content of the digital object and metadata to describe this for different purposes. Each digital object may hold more than one content file, or each digital object may hold one content file, and have a relationship expressed to one or more other digital objects. Datastreams for metadata may be used for descriptive metadata (e.g., DC or MODS), technical metadata (particularly useful for images or other multimedia), and/or preservation metadata (e.g. PREMIS), whatever is required.

Aside from the RELS-EXT datastream Fedora also allows for the creation of a RELS-INT datastream that can be used to record relationships between content files or datastreams within the digital object, again using RDF. This feature was fully introduced in the version 3.3 release in January 2010. It will be valuable to see how this gets used or for there to be more exploration of how it could be used, particularly in the context of managing ORE Resource Maps.

Atomistic, compound and complex digital objects

As indicated in the previous section, digital objects may have one or more content files within them, and may also express relationships with other digital objects. The ARROW project in Australia defined two approaches as below:

- Atomistic: a digital object with one or more content datastreams that are all considered primary to the object. This may include a digital object for an image that contains separate content datastreams for different resolutions
- Compound: a digital object consisting of multiple content datastreams that are not all primary to the object. This may include a piece of text alongside one or more images and possibly a video, as in learning materials

The choice of whether to take an atomistic or compound approach to take is flexible and will depend on identified requirements by the repository managers or within the community within which the repository will play a part. ARROW decided on a compound approach for practical reasons, though other repositories have adopted an atomistic one (which has been argued as a purer approach if one that is likely to require more configuration and processing time).

Alongside compound objects, the Fedora digital object model also allows for complex objects, where a parent aggregation object is related to multiple child digital objects, each one with a single content datastream. This highlights that defining which approach to take is not just dependent on the content itself and how this is structured, but also on the relationships, or potential relationships, that exist between the content files.

ETDs (or enhanced publications generally) – compound or complex?

Given the different means of structuring digital objects within Fedora described, which approach is likely to best suit ETDs? There is nothing specific about ETDs that places any limits on how Fedora is used to manage them, but their role and the requirements that people and organisations have for them is likely to influence this. Options for structuring ETDs will include the following (others may be possible or desired):

Atomistic – The ETD, whether made up from a single or multiple text files, is made into a single file of a preferred format. This approach is how much ETD management is taking place currently, with PDF as the preferred file format. It can be used to manage a compound ETD where the ETD is made up of multiple text files, though cannot take account of ETDs that include files of different formats easily.

Compound – The ETD is made up from multiple content files, which may be of different formats. For example, the ETD may have a text file as the main body of the thesis, but will also have associated images and/or datasets that exist in a non-

textual native format. Fedora is able to construct a digital object that can encompass all of these content files together and associate relevant metadata about the ETD with them. Where metadata about individual content files is required this can be stored as well, and the relationship recorded in the RELS-INT field. However, such relationships may be non-standard and impact upon future interoperability due to the current status of this datastream within Fedora. As such, there are likely to be some limitations to using this approach as ETDs become more complex. Where component content files of the ETD will be used in other circumstances managing them as separate digital objects may also be of value.

Complex – The ETD is described using a network of digital objects within the repository. A parent object represents the ETD as a whole, but has no content datastreams. Each content file is managed through its own digital object, with each having its own metadata as required. The RELS-EXT datastream is used to record the relationships between the parent and child objects, which may take the form of RDF relationship statements coming from either direction. This approach offers the maximum level of flexibility in structuring and managing a ETD that comprises of multiple content files, and is an extension of the atomistic approach taking advantage of the RELS-EXT capability.

There is no fixed best practice. Notwithstanding the advantages that a complex object approach suggests, it is also the approach that is likely to require greater resource in order to keep track of all the objects and ensure the ETD does not get corrupted. Ingest will also be more complex itself. As with the ARROW decision, a compound object approach may suffice.

Functionality of relevance to compound ETDs in Fedora

I. Ingest

Digital objects can be ingested into Fedora in one of three formats, as described below. Compound ETDs can be structured using any one of these for ingest where all the constituent content and metadata files are known and available.

- FOXML – this is Fedora native 'Fedora Object XML' and was created to enable a number of system-related features in Fedora. It is a derivative of METS and is currently at version 1.1 (though version 1.0 is still usable for backwards compatibility).
- METS – notwithstanding the derivation of METS used by Fedora internally, Fedora can also accept a METS 1.1 file for ingest. However, due to the generic nature of METS, there are also some Fedora-specific rules that need to be adhered to ensure ingest is successful.
- ATOM – The ATOM Syndication Format, in conjunction with ATOM Threading Extensions, can also be used to structure a digital object for

creation, using different elements to describe the objects themselves and the datastreams within them.

Where not all the content files and/or metadata are available digital objects can be created individually and metadata created for them through an appropriate client (the exception being the DC datastream which is created by default when a digital object is created). Content files can be stored within the Fedora repository or referenced elsewhere as and when they are added to the digital object.

II. Export

The same formats as for ingest can be used as the formats for exporting objects from a Fedora repository. However, in order to facilitate the use of the exported digital object(s) a CONTEXT is set as follows:

- Migrate – this can be used if the primary purpose of the export is to migrate the object(s) to another Fedora repository.
- Public Access – this can be used if the primary purpose of the export is to make the object(s) widely available through another means
- Archive – this can be used where the primary purpose is to create a self-contained archive of the object(s) that can be used at some point in the future without loss of integrity

III. Access

Access to digital objects within a Fedora repository is available through a variety of methods; for example, systems can make use of the web service APIs, API-A or API-M. Fedora also offers the use of disseminators, which can carry out actions on a piece of content as part of delivering it. This might be to package up a digital object as an ORE Resource Map, for example, as an ATOM feed. Web clients are not a standard deliverable with Fedora and a number of local and more widely distributed alternatives are available to select from. Access to compound ETDs through these clients will be affected by the exact functionality that the client makes available and the reader is directed to sources of information for those clients for further information.

A brief case study of the Muradora client is given here. This client allows all three of the digital object structures to be used. An object can be viewed with all its datastreams. Alternatively, a parent object can also be created and child objects related to it, and this is displayed as a collection with the constituent members. From a user perspective the atomistic/compound object approach displays all the relevant information on one screen: for complex objects each content file needs to be accessed separately, requiring more clicks. In contrast, Muradora supports full-text indexing. When a search is carried out and results are found in both compound and complex objects the user is able to know in exactly which content file the search

term was found, whereas in compound objects (s)he will need to explore the available content files to identify the exact hit.

IV. OAIS – Open Archival Information System

One of the main reasons for moving digital objects in and out of a repository is for preservation purposes. The options within Fedora and its underlying model provide much of the functionality that is required to support such preservation practice. The system maps to the OAIS Reference Model as shown in Figure 2, which summarises the different capabilities Fedora has for ingest, internal management and access/export.

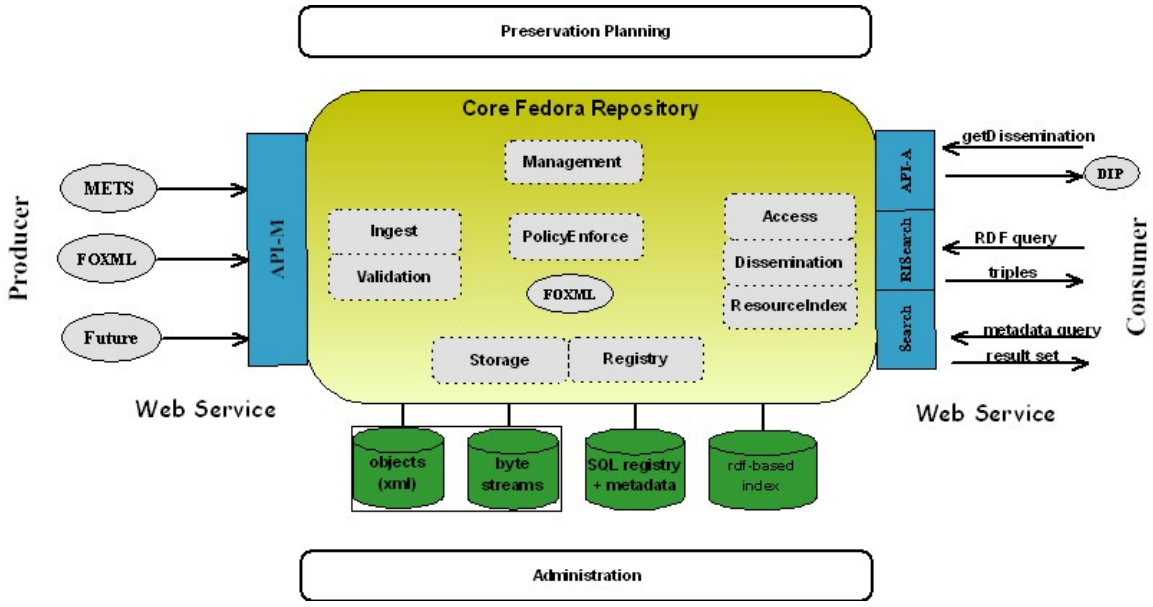

Figure 2: Mapping Fedora onto the OAIS

Section 2: Fedora compatibility with the Object Reuse and Exchange (ORE) specification

Introduction

The Object Reuse and Exchange specification version 1.0 was released in October 2008, though its widely available pre-releases have attracted extensive interest and testing during its development over the previous two years. The specification has been released under the auspices of the Open Archives Initiative, but can be considered a community standard due to the wide level of input and suggestion made to the Editors (Herbert Van de Sompel and Carl Lagoze) and the associated ORE Technical Committee. An introductory primer to the ORE specification can be found on the web pages of the Open Archives Initiative. This paper will not seek to replicate this information here, but will use an example to provide context for the discussion of ORE and Fedora.

ORE summary and example

When we browse the Web we collect resources, either through bookmarked links or collections of multimedia (e.g., images). However, whilst we make use of the individual content files, we rarely make active use of the collections themselves. Technically this is largely because there is no easy way to identify the collection or define its boundaries or what is in the collection. ORE provides a way of achieving these capabilities and allowing collections to be structured from content files and references all over the Web, not just within an application.

In discussing the use of Fedora to manage compound ETDs in this document, an assumption has been inherent that the ETD will be contained within Fedora. Content files do not have to be held within the repository to be part of a Fedora digital object, but the underlying management structure is the Fedora repository and how this determines you can work. ORE offers an alternative viewpoint. It starts the discussion about compound ETDs from the perspective of the ETD itself, the aggregation of content files and metadata. Fedora can be used to manage this aggregation, but the aggregation, the compound ETD, is not bound by it.

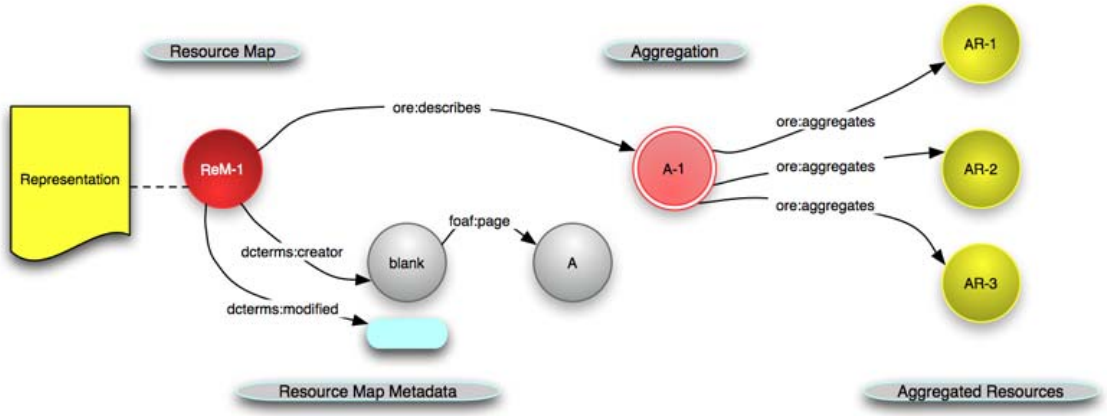

Figure 3: An example ORE model

The aggregation A-1 comprises three aggregated resources (AR-1, AR-2 and AR-3): these may have additional relationships between them. The aggregation is described using a Resource Map (ReM-1), which has metadata of its own. The Resource Map is what it says, a map that describes the constituent parts of the aggregation and their relationships. The Resource Map can be represented through an appropriate serialisation, of which ATOM is the most widely known, though varieties of RDF can also be used. Systems that can interpret these representations can be used to manage Resource Maps and the aggregations they describe.

Fedora and ORE

Given the relative newness of the ORE specification there has not been time for Fedora to formally adopt the capability to work with the specification. And yet it

may be that such formal adoption is not absolutely required. The Oxford Research Archive, for example, makes ORE Resource Maps available in a variety of RDF serialisations through its public interface, and there has been a flurry of activity across the repository community in general to demonstrate its usefulness during the latter part of 2008; for example, the JISC-funded TheOREm project at the University of Cambridge has taken a specific look at how ORE might be used for compound chemistry ETDs that include chemical structural datasets.

The specification fits snugly with the Fedora digital object model. ORE is based around the description and identification of resources and the relationships between them: within Fedora the RELS-EXT datastream is used for relationships and each digital object is uniquely identifiable on the Web. The most obvious parallel is with the complex object approach in Fedora, though compound objects can also be linked to within an ORE aggregation.

Within the Fedora community there has been some activity. Oskar Grenholm released an initial oreprovider plugin in March 2008 that made use of the RELS-EXT relationships to generate the Resource Map. Eddie Shin at Fedora Commons also wrote a prototype ORE provider plugin: both used ATOM as their default serialisation, although both were released prior to changes in the ORE specifications in September 2008 that altered the way ATOM should be used. Fedora is formally testing how ORE can be best used with other partners (including DSpace, under the DuraSpace umbrella, and Microsoft). One key possible advantage of using ORE is to help structure the RELS-INT datastream for compound objects, allowing more fine-grained relationships to be expressed in a standard way.

Section 3: Recommendations for the roadmap

The following recommendations are made for further enabling the use of Fedora for the management of compound e-theses using ORE:

- The development of tools to facilitate ingest of ORE Resource Maps and their constituent parts into a Fedora repository, recoding the relevant relationships as required
- The development of more tools to facilitate the generation of ORE Resource Maps (which may be Fedora-related disseminators or other) by end-users, extending the initial work carried out in 2008 at the time of the launch of this protocol
- The development of clients that can navigate ORE Resource Maps intelligently to allow them to be interpreted correctly and clearly

It is acknowledged that some work in these areas does exist, but that additional effort is merited to achieve wider adoption.