

## Content Models in Fedora 3.0

### Process for Creating Content Models

Although we are still learning about content models, these models and Fedora's 3.0-based CMA (Content Model Architecture) offer several advantages as follows: a) CMA offers a formal method for defining object classes, what we have previously referred to as the object architecture, b) whereas in Fedora 2.x disseminator code must be bound into the object making it awkward to change disseminator behavior, Fedora 3 uses content models to drive disseminators, effectively uncoupling them from individual objects and making it easier to alter their functionality, and c) Fedora 3.0 offers validation services related to content models. For example, we presumably will be able to validate on ingest that an object has the required archival master.

In an earlier software architecture working group meeting, we concluded that content models will be created for every object architecture that is currently supported in WMS. In addition we also concluded that every resource object would be associated with a content model, whether or not the object has a disseminator. For R5.0, the architecture release, content models will be created manually, a process which will be described further in this specification. For R5.1, we may add capability to WMS for the creation and editing of content models. This topic will be discussed further in a future sw\_arch working group meeting. There are basically two major scenarios for content model creation that need to be supported in R5.0: a) content models for objects that are migrated from RUcore/Fedora 2.1 and b) new content models that are created once R5.0 is in production.

### Migrated Content Models

In Fedora 2.1, the notion of a content model was introduced but it was treated only as a simple string and was not controlled by the system in any way. In CMA and Fedora 3.0, a content model is used to establish a set of DTD-like constraints to which the object should conform. Constraints include the mime type, minimum and maximum number of instances of a datastream, and whether multiple instances of a datastream are ordered. The Fedora 3.0 migration capability will automatically create content models based on the structural elements of the XML objects. Since there are slight variations in the structure of an object with a specific architecture, the migration utility creates many content models for a specific architecture. Since this content model creation is based on a heuristic that has little relationship to the RUcore environment, these automatically created content models do not appear to be particularly useful.

#### *Proposal for Creating Content Models.*

1. Establish a base class of content model that is common to all objects. For example, all objects have the following datastream IDs: MODS, DC, SMAP1 and either ARCH or

RARCH substrings. See attachment 1 at the end of this document for sample showing the composite model xml.

2. Using this base model as a template and working with an xml editor, create content models for each object architecture. For example, a photograph would have both a PDF-1 and a DJVU-1 datastream. A video would have an FLV-1 and a MOV-1 datastream. (See Attachment 2 below.)
3. The result of step 2 would be the 15 (or so) content models that represent our current object architectures. (See Attachment 3 below for a list of object architectures that will require content models in R5.0.) The “manifest file” resulting from the migration step (members.txt) could then be edited to point to the newly created content models. An XSLT transformation would then be run to update the object FOXML.
4. At present, the only objects that have disseminators are collection objects, book objects, and dynamic collection objects. The respective content models for these objects would need to be linked to the appropriate bmech objects.
5. There should be an attribute identified in the content model that carries the name of the class, e.g. book, photograph, etc. Proposal: LABEL=”content model”, DC Title should be the generic class, i.e. book, photograph, etc and the object ID can now be of the form “rutgers-lib:Book-cmodel”.
6. Per an earlier sw\_arch discussion, we will add the element “contentModel” to techMD. We also decided to leave the older objectArchitecture elements in place. As an example, for migrated books we would see contentModel=”book” and objectArchitecture=”book”. Newly created objects in R5.x would only have the contentModel element.
7. MODS descriptive metadata will not be needed for content model objects. In this sense, they are very similar to the old disseminator objects (bdefs and bmechs).

### **Creating New Content Models**

After RUcore R5.0 is running in production mode, we expect that new content models will be needed and current ones will probably have to be edited.

*Creating New Models.* Assuming steps 1 and 2 will work, a new content model could be created by using the base model of step 1 above and an xml editor. Ingest could be accomplished via dlr/EDIT. Since any new content model may have impact on the software architecture, these new models and the associated XML should be reviewed in sw\_arch. New content models will include those for annotation objects, policy objects, and playlist objects.

*Editing of Existing Content Models.* Much of the editing of a content model could be undertaken in the xml editor provided by dlr/EDIT. For example, the constraints identified above are represented in the CompositeModel of the content model (see XML of attachment 1). These constraints could all be edited with dlr/EDIT.

## Issues and Questions

- The process outlined above requires code changes for R5.0 as follows:

- WMS

In lieu of adding disseminator code to the object, WMS should add a RELS-EXT datastream pointing to the content model. The following is an example of a new RELS-EXT datastream assigning a collection-disseminating content model:

```
<foxml:datastream ID="RELS-EXT" STATE="A" CONTROL_GROUP="X"
VERSIONABLE="true">
<foxml:datastreamVersion ID="RELS-EXT.0" LABEL="Relationships"
CREATED="2008-05-05T20:10:21.901Z" MIMETYPE="text/xml" SIZE="236">
<foxml:contentDigest TYPE="DISABLED" DIGEST="none"/>
<foxml:xmlContent>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:fedora-model="info:fedora/fedora-system:def/model#">
<rdf:Description rdf:about="info:fedora/rutgers-lib:XXXXYYZZZ">
<fedora-model:hasContentModel rdf:resource="info:fedora/rutgers-
lib:collection-cmodel"/>
</rdf:Description>
</rdf:RDF>
</foxml:xmlContent>
</foxml:datastreamVersion>
</foxml:datastream>
```

- There are minor changes to the API required for WMS edit.
    - The WMS should add contentModel to techMD.

- dlr/EDIT

- Ability to edit content models.
- What validation can be accomplished on ingest via Fedora and the content model?
- What guidelines can be provided for system testing content models? A) Objects with disseminators should be given special attention, B) editing of content models, C) We need to develop all content models as per the list in Attachment 3.
- We should explore attaching multiple disseminators to a content model. This capability could enable us to deliver a book in several different ways.
- There is currently a known bug limiting the resource index that shipped with the Fedora 3 beta software, and this could have an impact on our assignment of content models and disseminators. We need to explore this issue further, conducting tests on our server and querying the people at Fedora.

### Attachment 1 – XML excerpt of Content Model – Showing Base Model Template

```

<dsCompositeModel xmlns="info:fedora/fedora-system:def/dsCompositeModel#">

<dsTypeModel ID="SMAP1" ORDERED="false" MIN="1" MAX="1">
<form MIME="text/xml"></form>
</dsTypeModel>

<dsTypeModel ID="DC" ORDERED="false" MIN="1" MAX="1">
<form MIME="text/xml" FORMAT_URIS=""></form>
</dsTypeModel>

<dsTypeModel ID="MODS" ORDERED="false" MIN="1" MAX="1">
<form MIME="text/xml" FORMAT_URIS=""></form>
</dsTypeModel>

<dsTypeModel ID="ARCH1" ORDERED="false" MIN="1" MAX="1">
<form MIME="application/x-tar" FORMAT_URIS=""></form>
</dsTypeModel>

</dsCompositeModel>

```

### Attachment 2 – XML excerpt of Content Model for a Photograph (rutgers-lib:photograph-cmodel)

```

<dsCompositeModel xmlns="info:fedora/fedora-system:def/dsCompositeModel#">

<dsTypeModel ID="SMAP1" ORDERED="false" MIN="1" MAX="1">
<form MIME="text/xml"></form>
</dsTypeModel>

<dsTypeModel ID="DC" ORDERED="false" MIN="1" MAX="1">
<form MIME="text/xml" FORMAT_URIS=""></form>
</dsTypeModel>

<dsTypeModel ID="MODS" ORDERED="false" MIN="1" MAX="1">
<form MIME="text/xml" FORMAT_URIS=""></form>
</dsTypeModel>

<dsTypeModel ID="PDF-1" ORDERED="false" MIN="1" MAX="1">
<form MIME="text/xml" FORMAT_URIS=""></form>
</dsTypeModel>

<dsTypeModel ID="DJVU-1" ORDERED="false" MIN="1" MAX="1">
<form MIME="text/xml" FORMAT_URIS=""></form>
</dsTypeModel>

<dsTypeModel ID="JPEG-1" ORDERED="false" MIN="1" MAX="1">
<form MIME="text/xml" FORMAT_URIS=""></form>
</dsTypeModel>

```

```
<dsTypeModel ID="ARCH1" ORDERED="false" MIN="1" MAX="1">  
<form MIME="application/x-tar" FORMAT_URIS=""></form>  
</dsTypeModel>  
  
</dsCompositeModel>
```

### **Attachment 3 – Current WMS Object Architectures**

Article  
Audio  
Book  
Diary  
Manuscript  
Map  
Newsletter  
OralHistory  
Pamphlet  
Periodical  
Photograph  
Poll  
Record  
Video

rcj/jat – 06/26/2008