Content Model (object architecture) for Books **June 27, 2006**

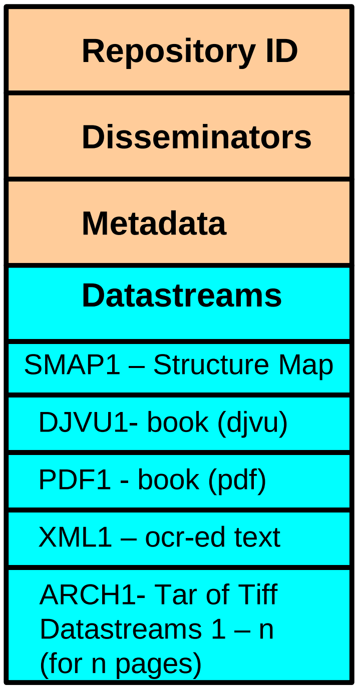
This document describes the object architecture (content model) for books to be deposited in RUcore. Most of what appears here is extracted from previous draft documents. This document reflects, to the best of my knowledge, the approach we are using in the current release of Rucore.

Object Architectures and Content Models

Most of the Fedora community is using the vocabulary “content models” for what we have been calling object architecture. We should begin using the “content model” phrase rather than object architecture in our discussions. This will not only clarify communication with Fedora partners, but will also help in further architectural discussions. Specifically, in the next release, Fedora will come out with a new capability referred to as Content Model Disseminator Architecture (CMDA). This architecture will vastly improve our ability to handle disseminators for well defined objects, specifically by not binding the disseminator definition into the object.

Book Content Model (CM)

The book CM has five datastreams as shown in the following diagram. Since many of our books our outsourced for scanning, we decided to allow for ingest of the previously created pdf, djvu, and xml (ocr text) datastreams. This situation would occur when a vendor can easily provide these files along with the tiffs. Alternatively, for inhouse scanned materials, we also have the option for the pipeline to create the pdf, djvu, and xml files from the scanned tiff images.



Book Content Model

In earlier meetings, we spent considerable time discussing how to handle the archival master and the tiff images. We discussed the advantages and disadvantages of placing each tiff image in a separate datastream. In the end, we decided that tar-ing all the images in a single tar file would promote ease of handling without reducing access flexibility.

Structure Map

The structure map for a book is used by the book disseminator to display the table of contents. An excerpt from a typical structure map is shown below. It is also possible to use a default structure map for the book in which case there would be no table of contents.

```
<structMap TYPE="logical">
<div ID="div1" LABEL="New Sweden Colony" TYPE="book">

<structMap TYPE="logical">
<div ID="div1" LABEL="Orientation Handbook for Indochinese Refugees"
TYPE="book">

<div TYPE="Cover" LABEL="Orientation Handbook for Indochinese Refugees"
ORDER="1">
<area FILEID="" BEGIN="1" END="1"/>
</div>

<div TYPE="Prologue" LABEL="Prologue" ORDER="2">
<area FILEID="" BEGIN="4" END="4"/>
</div>

<div TYPE="Acknowledgements" LABEL="Acknowledgements" ORDER="3">
<area FILEID="" BEGIN="5" END="5"/>
</div>

<div TYPE="Table of Contents" LABEL="Table of Contents" ORDER="4">
<area FILEID="" BEGIN="6" END="8"/>
</div>

<div TYPE="Chapter" LABEL="Chapter I: Laws, Regulations and Programs
Affecting Your Life and Work" ORDER="5">
<area FILEID="" BEGIN="9" END="16"/>
</div>


.
.
.

<div TYPE="Resources" LABEL="List of Other Refugee Assistance Programs"
ORDER="14">
<area FILEID="" BEGIN="43" END="43"/>
</div>

</div>
</structMap>
```

Architecture-related Metadata

The architecture related fields that must be included in the metadata are: 1) The typeOfResource is “image”, 2) the objectArchitecture is “book”, and 3) objectArchitecture is replicated in the first level div TYPE attribute of the structure map (see below).

```
<structMap TYPE="logical">
<div ID="div1" LABEL="New Sweden Colony" TYPE="book">
```

Summary Discussion

Our current content model for books is what can be called a minimal datastream model in that we have few datastreams (a total of five). There are at least two other models that various institutions are using or considering. A more complex model includes a datastream within the object for every tiff image. This approach makes it easier to access or manipulate a single page tiff image. For example, you may want to include a single page in a learning management system or apply a signature to each page. The other more complicated model is the network or graph model in which each page is rendered as a single object and there is a parent object for the book that brings all pages together. This model might be of some benefit for special applications, but it doesn’t have any apparent advantages for the multi-disciplinary repository.

rcj – 06/27/2006