Extending an atomistic Fedora-Commons object model to facilitate image segmentation and enhance discovery

Presentation Proposal
Open Repositories 2013

David Lacy
Villanova University

We have hundreds of thousands of scanned pages from books, manuscripts, letters, theses, newspapers, scrapbooks, etc. How can we further describe this material logically, while optimizing it for research discovery? This presentation aims to provide a stack-agnostic strategy for extracting new Fedora objects from existing content, and illustrate how to incorporate this new material into existing discovery layer schemes.

Topics include:

- Organize pages from books logically, into chapters
- Create new objects out of articles extracted from newspapers, snippets from scrapbooks, and other items from complex image data
- Find specific image data objects, resources, and hierarchical folders through the discovery layer, without cluttering the query results

Background

The existing hierarchical model contains 3 basic models: *Core*, *Collection*, and *Data* (fig. 1). The *Core* model describes the datastreams and methods common among all objects (`THUMBNAIL`, `PARENT-LIST`, etc).

The *Collection* model contains datastreams and methods used to describe the object's members. Helper models exist for *Collections*, that further denote the purpose of the object. *Folders* are used to organize objects within a hierarchy, and *Resources* signify an object as it exists in the physical form. *Resources* can be Books, Manuscripts, Newspapers, Letters, etc.

The *Data* model contains `MASTER` and `MASTER-MD` datastreams, used to store the file and technical metadata respectively. Other helper models exist that further extend the *Data* model to include additional derivatives of the `MASTER` datastream, and other methods dependent on the type of file contained in the `MASTER` datastream.

These models are then applied to objects within the collection and related using the `rel:isMemberOf` assertion (fig. 2).

Apache Solr[1] is used for the discovery layer and our index contains records representing *Resource* object types.

1 http://lucene.apache.org/solr/

This presentation will address 3 extensions made to our Fedora-Commons object model and Solr schema that greatly expand the research possibilities of our collections.

Extension 1

An additional helper model (*List*) was created for *Collection* objects, and it is used to add groups of members beneath each *Resource* object (fig. 3). We moved the raw image data objects into one *List*, and created another *List* (Table of Contents) to represent the logical order of the work. *Image* objects contained in the "Raw Image Data" list contain multiple `rel:isMemberOf` relationships, linking them to multiple lists.

Extension 2

A new *Segment* model was created to extend the *Image* object to support a new `COORDINATES` datastream and `genSegment` method (fig. 4). The new *Segment* object maintains a relationship with a source *Image* object (`rel:isPartOf`) and its `MASTER` datastream is the result of the `genSegment` dissemination which takes the image-mapped `COORDINATES` of the *Image* source's `MASTER` file as arguments.

These new *Segment* objects can then be modeled like other *Resources*, and grouped with the source materials using *List* models (fig. 5)

Extension 3

We currently index *Resource* objects into Solr. The *Resource* members' (*Image* objects) `OCR-DIRTY` data is aggregated and included to provided full-text search capabilities. Our new strategy adds the *Folder* and *Image* objects to the Solr index. Each record contains the object's parent information, which will facilitate hierarchical browsing and collection-specific searching. The *Image* object records are suppressed from the search results utilizing Solr's field collapsing[2]. This results in page-specific `OCR-DIRTY` matches in the discovery layer, without cluttering the search results with thousands of *Image* objects (fig. 6).

---

2 http://wiki.apache.org/solr/FieldCollapsing

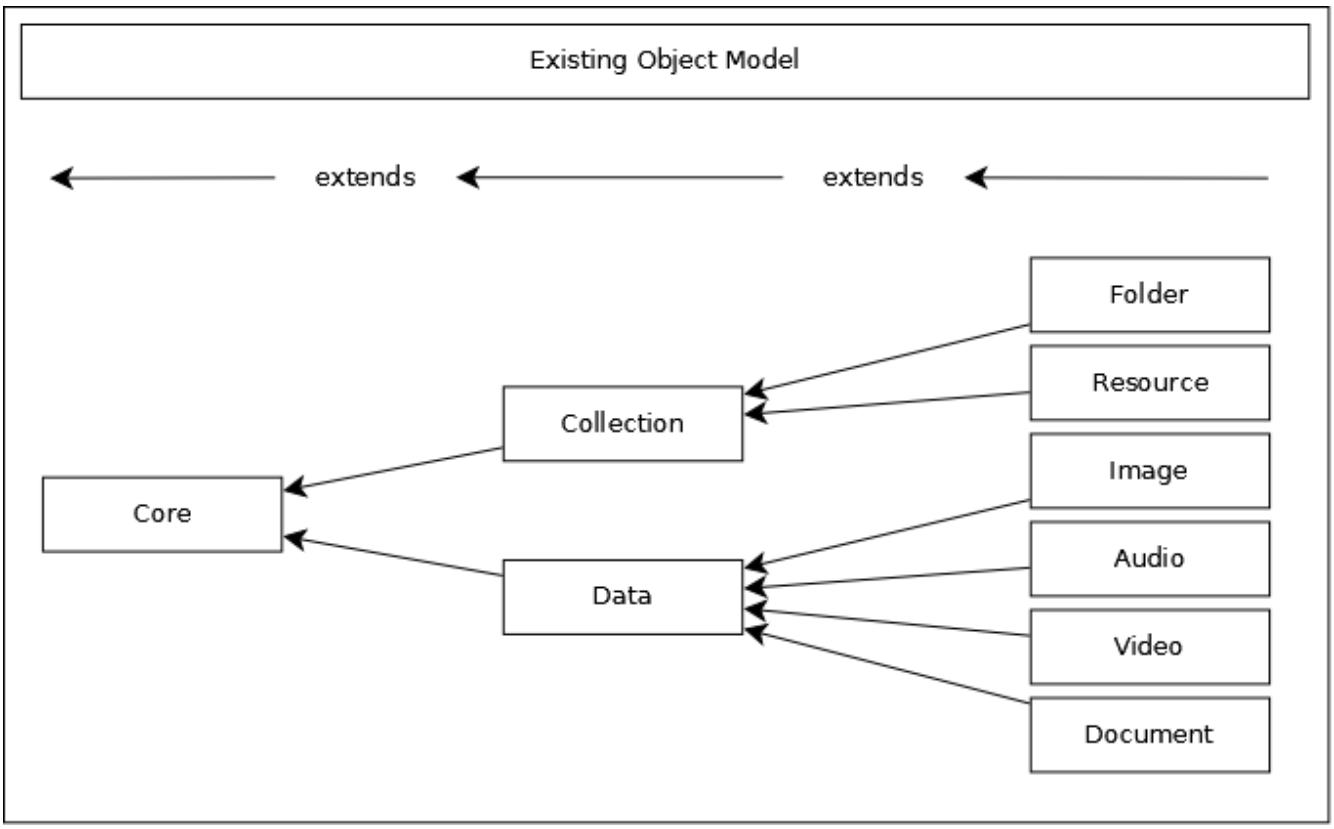

Figure 1

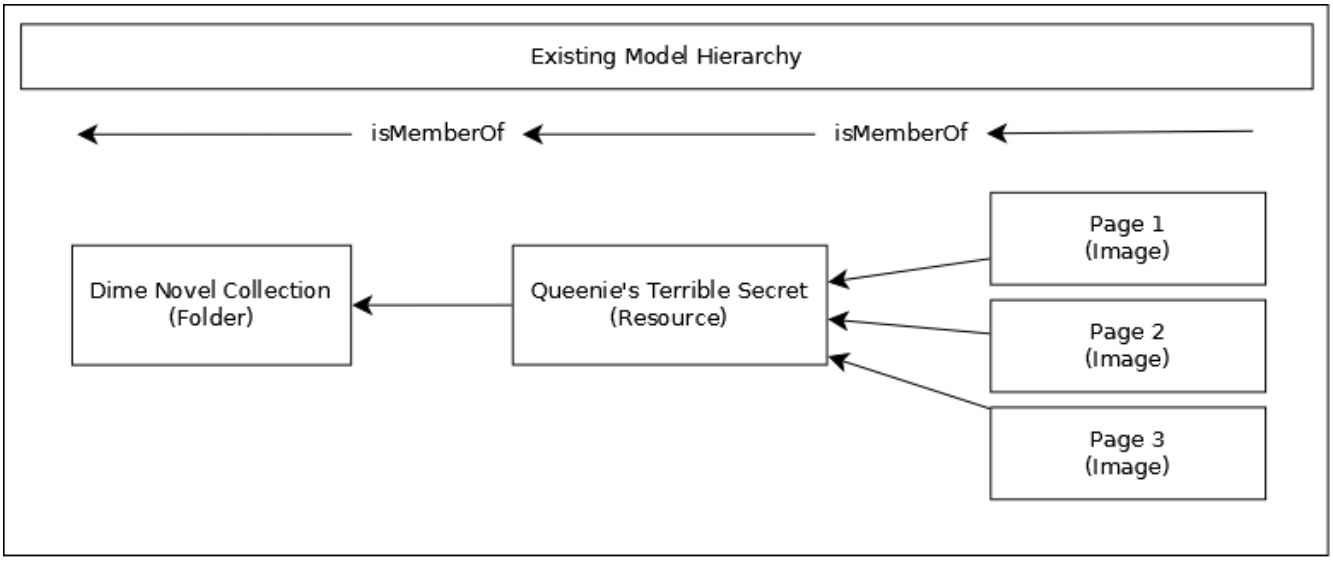

Figure 2

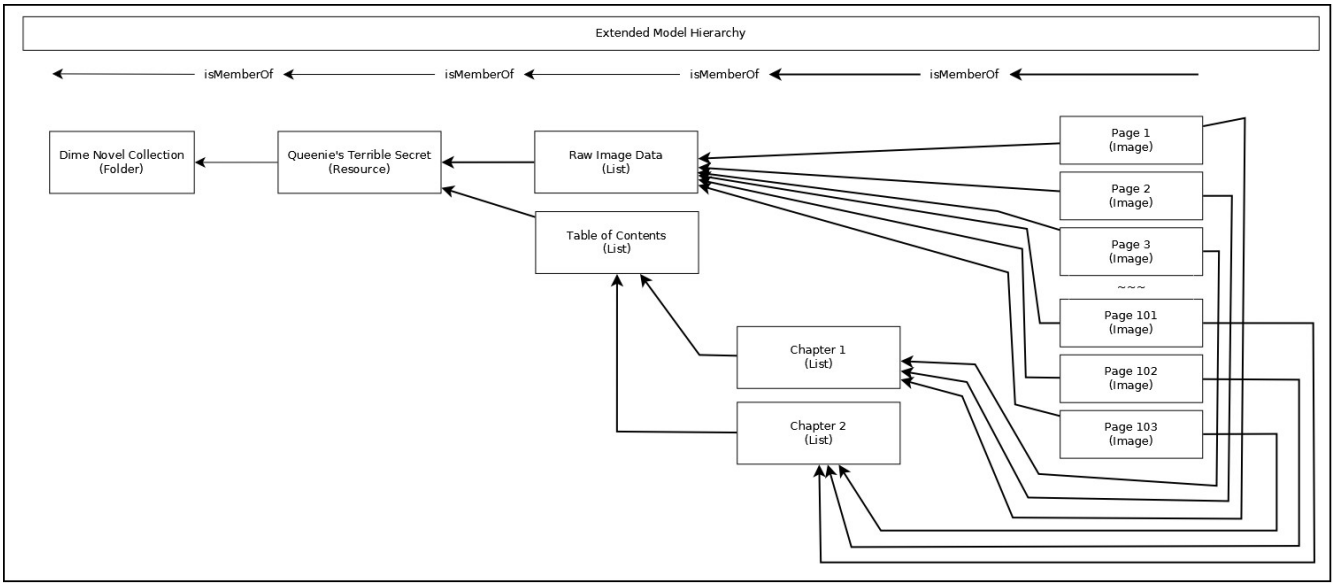

Figure 3

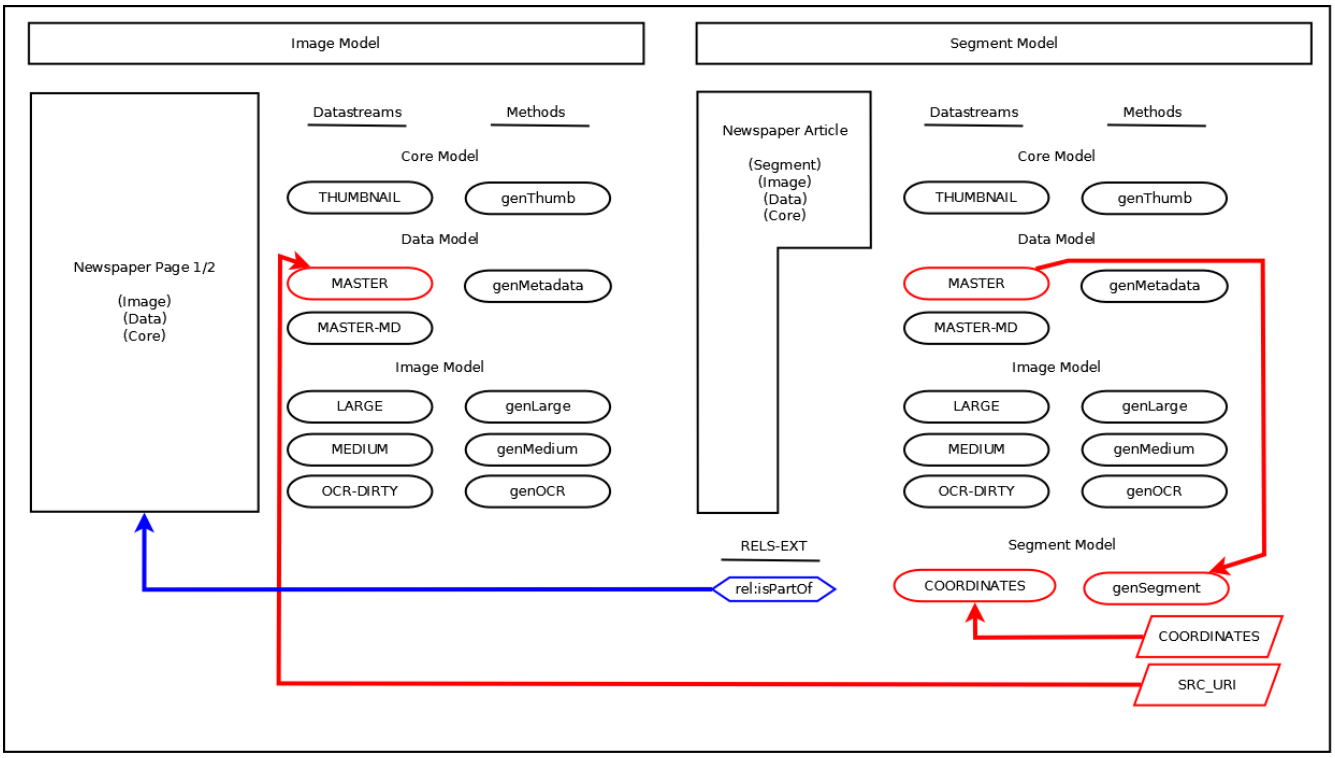

Figure 4

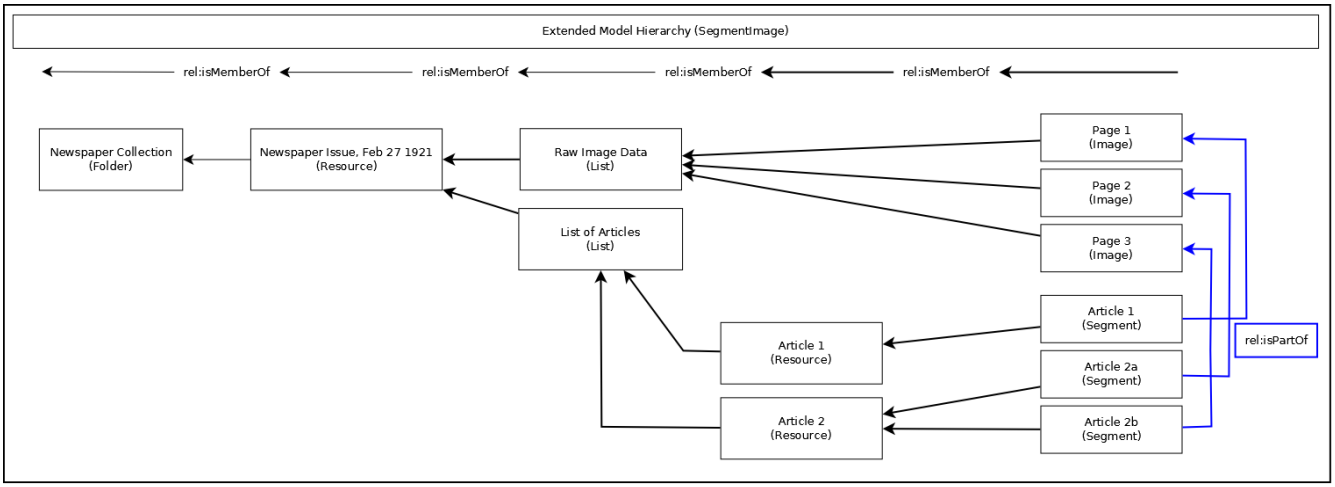

Figure 5

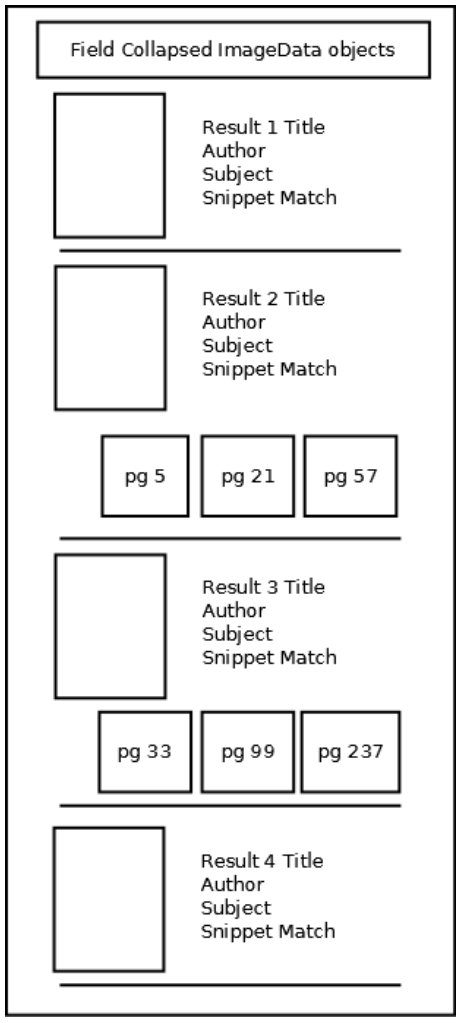

Figure 6