Atrium EAD Content Modelling

Overall Approach

Notre Dame's EAD modelling approach in Atrium is as follows:

1. Model an EAD based collection at the atomic level
   a. Each object contains only a snippet of EAD from the entire EAD finding aid for a collection
   b. The EAD is stored in the descMetadata datastream for each object
2. Generate the entire EAD finding aid from its atomic parts when necessary
3. Index atomic parts in solr so they are independently discoverable
4. Some parent metadata is indexed along with children so that children are discoverable via parent metadata.
5. If an EAD finding aid already exists, when ingesting into Fedora it is parsed and multiple atomic objects are generated from it.
6. EAD hierarchical relationships are maintained via RDF in the RELS-EXT datastream for each Fedora object.
7. If desired, an atomic object's EAD metadata may be edited directly in the Hydra head

Content Model for EAD objects

Object hierarchy:

- Collection
  - Child Image(s)
  - Child Component(s)
- Child Image(s)
- Child Component(s)

Collection

There is one Collection object that contains the ead header for the finding aid. A collection can have one or more child component objects. RELS-EXT Links between Collection and immediate child component objects are maintained via the hasMember relationship.

Component

There is a component object for each c# (c01, c02, c03, etc.) elements. Nested component elements are represented by an object independent of any parent/ancestor. A Component can have any number of nested Child Component objects (the EAD spec allows for 12 possible levels), and can also have child Image objects. Even though the original EAD finding aid might contain numbered components such as c01, c02, c03, etc., those components are converted to unnumbered component elements at ingest time. The role a component plays in the hierarchy is then determined by its context through relationships (ie. is it a direct child of the Collection, does it have item level child components) and by its level attribute. The level attribute can have the following possible values:

* collection
* fonds
* class
* recordgrp
* series
* subfonds
* subgrp
* subseries
* file
* item
* otherlevel

If a component has child components, the parent component's level value is used to determine how to map relationships to the children.

Image

An image object is based on Hydra's GenericImage model and can be a child of either a Collection or Component object. It also has (albeit small) portion of EAD xml represented in its own descMetadata datastream. It is mapped to a parent collection or component also via RELS-EXT.

EAD Fedora Object and xml Example

For example, if you have the following EAD xml:
This results in two Fedora objects, both converted to unnumbered components. The descMetadata for each object is below:

**Component 1**

```xml
<did>
  <container type="box-folder" label="Box">1 : 2-4</container>
  <unittitle>Dance programs</unittitle>
</did>
```

**Component 2**

```xml
<did>
  <container type="box-folder" label="Box">1 : 4</container>
  <unittitle>The Feast of Apollo, Brattle Hall, Massachusetts</unittitle>
  <unitdate normal="1925">1925</unitdate>
</did>
```

**Component 1 RELS-EXT:**

Because component 1 is the parent and has level value "series", a relationship is added to the RELS-EXT for Component 1 using the "hasSeriesMember" relationship. To simplify the example, we will give each the following pids: Component:1 and Component:2

```xml
<rdf:RDF>
  <rdf:Description rdf:about="info:fedora/Component:1">
    <hasModel rdf:resource="info:fedora/afmodel:Component"/>
    <hasSeriesMember rdf:resource="info:fedora/Component:2"/>
  </rdf:Description>
</rdf:RDF>
```

There are also two "daoloc" elements representing two images in Component two. So, we will create two Image objects as well with the following metadata:

**Image 1:**

```xml
<daoloc href="CC_1775_05_10_$3.front"/>
```

**Image 2:**

```xml
<daoloc href="CC_1775_05_10_$3.back"/>
```

**Component 2 descMetadata (updated):**
Component 2 RELS-EXT:

Hierarchical Relationship predicates

If Component 1 was level "subseries", the relationship used would be "hasSubseriesMember" instead. Using all possible level attributes the following custom predicates are added to the predicate mappings configuration for EAD hierarchical relationship linking:

*Note: To distinguish an EAD component collection from the overall collection, the relationship is called "hasComponentCollectionMember" instead of "hasCollectionMember" for level value "collection".

For linking to child images two additional relationships are also added:

Note: This text is a continuation of the previous one and includes additional relational predicates for EAD hierarchical relationship linking.