Chimera project

Content metadata

Added by Lynn McRae, last edited by Lynn McRae on Dec 20, 2009 (view change)
Labels EDIT LABELS
(None)

This page describes Stanford's approach to recording an object's structural information and associated metadata -- a specification for simplified, unified *contentMetadata* datastream in our Digital Object Registry (DOR). It combines and simplifies information contained in the fileSec/fileGrp and StructMap segments of METS. It carries a consistent set of summary technical metadata, while more variable and detailed technical information will be managed in the *technicalMetadata* datastream.

In the first version of DOR we had addressed the understanding of an object and its parts through a variety of structural cues -- strong naming conventions for datastreams (e.g., descMetadata, rightsMetadata, etc.), or specific content models for child objects (e.g., "googleScannedPage"). But we also resorted to co-opting datastream labels to hold specific information by implicit convention and we found places in Fedora metadata to tuck this information away. Such solutions were ad hoc and would not carry consistently across objects. We want three things from a new design:

- The ability for a complete representation of all the data and metadata being managed. This content metadata is our opportunity to capture the object physical description in a canonical way independent of the specific (i.e. Fedora) technology being used to manage that object. It becomes a simple basis for describing the object to other consumers such as shelving processes for the access environment, or during transfer to the preservation repository.
- A better general approach to capturing core/common information about objects along with more specific information germane to a specific type or colleciton of data
- A rational way of identifying what the parts are and the role they play within the object

The implementation will involve two expressions of this data -- a *contentMetadata* datastream within the object describing content files only, and a fuller XML expression for object transfers that include content metadata for the metadata files that are taken out of Fedora and included as additional object files.

contentMetadata

In this datastream we are describing all the parts that make up the object as it is *stored* and *managed* in DOR, including both data and metadata parts. This is a departure from a METS layout where all the metadata has articulated sections within one file, and metadata found in fileGrp and structMap describes only the files associated with that metadata. It means we can describe the metadata parts as well, because those datastreams become opaque files alongside other files when gathered together in a directory for transfer. We want access to, say, descriptive metadata to be the same as access to any other file that constitutes the object. METS also provided this when the METS file itself was saved as an additional content file, but with some awkward self-referencing issues. The main difference here is that multiple metadata parts azre broken out and remain as discrete resources in the object.

Note that some DOR operational/management datastreams (e.g., workflow data), or application specific data (e.g., Hydra UI properties), will not be described in this datastream.

Note too that we may not have to materialize what is described here as an explicit datastream. In our ETD work we are distributing the resource data illustrated here among child objects, but in principle this content metadata could be dynamically assembled. Further, this data should be easily transformed into various "contentMap" outputs that can present a subset of this information in useful views, e.g., page content for indexing vs file content for preservation.

Files

The basic unit of content metadata is a file description:

```
<file>
   <xxxData>
   <location>  (repeatable)
   <checksum>  (repeatable)
</file>
```

this unit contains

- basic file descriptors (id, format, mimetype, size, etc)
- one of a set of specialized tags associated with format types, e.g., an imageData tag
- one or more locations, urls or file paths, to support multiple access paths
- one or more checksum tags (we intend to record MD5 and SHA1 checksums)

Resources

Files are contain in and organized as "resources". A resource describes what a file is in relation to the object. You can have a simple resource consisting of just one file ...

```
<resource id=pdf>
    <file>
 </resource>
```

... relate multiple files together when they share common descriptions ...

```
<resource id=page>
   <file>             (page image)
   <file>             (OCR text)
</resource>
```

... or nest resources to create meaningful groupings (here, gathering individual page resources together as a "pages" resource facilitates delivering just page content to a client):

```
<resource id=pages>
   <resource id=page1>
      <file>
      <file>
   </resource>
   <resource id=page2>
      <file>
      <file>
   </resource>
         :
<resource>
```

Note that any resource hierarchy expressed here is just a view onto the files as managed in a directory. The directory itself may or may not be organized the same way. We will tend towards a simple flat directory of files for simplicity over a hierarchy of directories and sub-directories in our storage, e.g., for Phoenix, but this is a project by project decision.

Details

Resource descriptors

These attributes describe the nature of a resource within an object, what role it plays in the object. In METS, the fileGrp schema offers similar functionality through complementary GROUPID and USE attributes:

- For Phoenix, GROUPID recorded values like "MasterJPEG2000", "HOCR", and "AuxiliaryTar"
- For Phoenix, USE captured descriptive phrases there like "Archive Masters", "OCR Creation" and "Container for Data Provider Source Files"

These are not duplicated precisely in the proposal below, but similar descriptive attributes are provided.

The <resource> element

attribute		description
id="*identifier*"	1..1	A unique identifier for the resource in the context of this object.
contains="content|metadata"	1..1	Does the resource represent the contents of the work itself or metadata about the work? Note that some "content" may take the form of metadata, e.g., vector data used to create an online map. But if it defines the the "work", it is content. This attribute would likely be used on outermost (not nested) resource descriptions to characterize the basic parts of an object.
type="*type*"	0..1	Indicates what type of resource it is in terms usefule to managing the obkect and its resources. For example, a "page" resource describes the files that make up a Google scanned page. For ETDS, this can be used to distinguish the main PDF vs supplemental files vs permission letters. The type descriptions would be short codes, a usage key, not long descriptive phrases. In some cases naming conventions for the resource might serve adequately to distinguish resources, but *type* could be used independently to supplement this. For instance, with ETDs, the resources might be identified as supplemental01, supplemental02, etc, to retain input order, while the type value for all such resources could be just "supplemental".
sequence="*n*"	0..1	An integer value that can be used to designate an ordering of resources across the object or within a containing resource.

Resource sub-elements

element		description
<attr name="*name*">*value*	0..n	Optional, multiple. Attribute/value pairs germane to the resources, as defined by individual projects, e.g., google page tags and page numbering.
<file>	1..n	Described below.

The <file> element

These describe the physical file itself. One can think of these as attributes that exist as part of the file independent of the object it is in.

attribute		description
id="*filename*"	1..1	The id/name of the file as stored in the file system, relative to the root directory. This is the unique identifier for a file, and should be the file name itself plus any path within the containing directory that leads to the file, e.g., "000001.html" or "page1/image.jp2".
format="*format*"	1..1	Values generally as returned by JHOVE with some adjustments for consistency.

mimeType="*mimetype*"	1..1	Values generally as specified by IANA (http://www.iana.org/assignments/media-types/) and as generally returned by JHOVE validation services.
dataType="*type*"	0..1	.This tells what format and file type can't, such added information like some XML is conveying MODS, that some HTML is conveying hOCR, etc. It is simply a form of tagging for tracking and counting.
size="*integer*"	1..1	File size, in bytes.
preservation="yes"	0..1	Indicates a file should go to the preservation repository. Files not marked would typically be resources created for the access environment, such as display derivatives.

File sub-elements

element		description
<*xxxx*Data ... >	0..1	Format-specific details, e.g., <imageData>. Details below.
<sourcefile source="*sourceid*" filename="*source-filename*">	0..1	This is an operational attribute to help flag resources that came directly from a depositor or source and might need to go back intact, with an optional original file name if different.
<location type="url|path">*value*	1..n	Access URLs for the file. Provides capability of recording stacks, workspace, and other locations.
<checksum type="md5|sha1">*value*	1..2	Checksum values for the file.

Note: this design does not convey internal file management information like create/update timestamps, or version information where applicable. This information could be added later.

The following file formats are supported by the JHOVE toolkit and cover most of our current needs except for the un-constrained filetypes that may be uploaded as part of ETDs.

category	example	JHOVE format	DOR format	mimetype	encoding	dataType
image	jpeg2000 image file	JPEG 2000	JPEG2000	image/jp2		
	JPEG image file	JPEG	JPEG	image/jpeg		
	TIFF file	TIFF	TIFF	image/tiff		
	GIF file	GIF	GIF	image/gif		
text	Text file	UTF-8, ASCII	TEXT	text/plain	UTF-8, ASCII, etc	ocr
	Web page	HTML	HTML	text/html	UTF-8, etc.	hocr, alto
	Metadata XML file	XML	XML	application/xml (1)	UTF-8, etc.	mods
audio	AIFF audio file	AIFF	AIFF	audio/x-aiff		
	MP3 audio file	MPEG? (3rd party)	MPEG	audio/mpeg		mp3
	WAVE audio file	WAVE	WAVE	audio/x-wave		
other	PDF file	PDF	PDF	application/pdf		
	ZIP file	ZIP? (3rd party)	ZIP	application/zip		

(1) JHOVE returns **text/xml**, which is largely deprecated, while **application/xml** is generally preferred.

Some additional files and how we would treat them; format simply derived from file extension?

category	example	format	mimetype	dataType
other	Word document	DOC	application/msword	
	Powerpoint slides	PPT	application/mspowerpoint	
	Excel spreadsheet	XLS	application/msexcel	
	TAR file	bitstream	application/x-tar	

Elements for format specific attributes

element	attributes
<imageData>	height="*pixels*" width="*pixels*"

Examples

What googleScannedBook *contentMetadata* might look like:

```
<contentMetadata type="googleScannedBook">

   <resource id="page1" sequence="1" type="page">
      <attr name="pageType">Title</attr>
      <attr name="pageNumber">3</attr>
      <attr name="pageLabel">ii</attr>
      <attr name="googlePageTag">IMAGE_ON_PAGE,IMPLICIT_PAGE_NUMBER</attr>
      <file id="00000001.jp2" format="JPEG2000" mimetype="image/jp2" size="169627" preservation="yes"">
         <imageData height="800" width="1200">
         <location type="url">http://service/druid/00000001.jp2</location>
         <location type="path">/dor/workspace/...</location>
         <checksum type="md5">56dd37697f05073168b9b58ddaccad0a</checksum>
         <checksum type="sha1">884b7650725011de8a390800200c9a66</checksum>
      </file>
      <file id="00000001.html" format="text" mimetype="text/html" encoding="UTF-8" dataType="hocr" size="734" preservation="yes">
         <location type="url">http://service/druid/00000001.jp2</location>
         <checksum type="md5">60dd37697f05073168b9b58ddaccad0a</checksum>
         <checksum type="sha1">324b7650725011de8a390800200c9a66</checksum>
      </file>
   </resource>
   <resource id="page2" ...
          :

   <resource id="googleMETS" contains="metadata" source="google">
      <file id="34521242324.xml" format="XML" mimetype="application/xml" encoding="UTF-8" dataType="METS" size="55911" preservation="yes">
         <location type="url">http://service/druid/fedora/ds/googleMETS</location>
         <location type="path">/dor/workspace/...</location>
         <checksum type="md5">50dd37697f05073168b9b58ddaccad0a</checksum>
         <checksum type="sha1">424b7650725011de8a390800200c9a66</checksum>
      </file>
   </resource>
   <resource id="descMetadata" contains="metadata">
      <file id="descMetadata.xml" format="XML" mimetype="application/xml" encoding="UTF-8" dataType="MODS" size="4391" preservation="yes">
         <location type="url">http://service/druid/fedora/ds/googleMETS</location>
         <location type="path">/dor/workspace/...</location>
         <checksum type="md5">20dd37697f05073168b9b58ddaccad0a</checksum>
         <checksum type="sha1">774b7650725011de8a390800200c9a66</checksum>
      </file>
   </resource>
```

- Instead of "groups" or "aggregates" or "divs", we describe the parts as embedded "resources"
- <file> is always contained/described as a resource sub-element, in order to support metadata about how the file relates to the object
- units for file size (bytes?)
- units for height/weight (pixels?)

What ETD *contentMetadata* might look like:

```
<contentMetadata type="etd">

   <resource id="main" type="main-original" data="content">
      <file id="mydissertation.pdf" format="PDF" mimetype="application/pdf" size="758621" preservation="yes">
         <location type="url">http://service/druid/mydissertation.pdf</location>
         <location type="path">/dor/workspace/etd/</location>
         <checksum type="md5">b54ccd1075c511de8a390800cc927150</checksum>
         <checksum type="sha1">af73868075c511de8a390800d99ff700</checksum>
      </file>
   </resource>
   <resource id="main" type="main-final" data="content">
      <file id="mydissertation-final.pdf" format="PDF" mimetype="application/pdf" size="751418" preservation="yes">
         <location type="url">http://service/druid/mydissertation-final.pdf</location>
         <location type="path">/dor/workspace/etd/</location>
         <checksum type="md5">17bfe54075d811de8b3c0440200c7c66</checksum>
         <checksum type="sha1">1b204e5075c621de8a390803300c9a66</checksum>
      </file>
   </resource>
   <resource id="supplement-1" type="supplement" data="content">
      <file id="datafile.xls" format="XLS" mimetype="application/ms-excel" size="83418" preservation="yes">
         <location type="url">http://service/druid/datafile.xls</location>
         <location type="path">/dor/workspace/etd/</location>
         <checksum type="md5">17bfe54075c611de8a390800200c9a66</checksum>
         <checksum type="sha1">1b204e5075c611de8a390800200c9a66</checksum>
      </file>
   <resource id="permissions-1" type="permissions" data="content">
      <file id="xyz-permission.txt" format="TEXT" mimetype="application/text" encoding="UTF-8" size="341" preservation="yes">
         <location type="url">http://service/druid/xyz-permission.txt</location>
         <location type="path">/dor/workspace/etd/</location>
         <checksum type="md5">17bfe54075c611de8a390800200c9a66</checksum>
         <checksum type="sha1">1b204e5075c611de8a390800200c9a66</checksum>
      </file>
   </resource>
```

For reference, we have two earlier forms of expressing file contents separate from the METS fileGrp and StructMap:

- Current Simple Content Map service
- Richard's fileAggregateList used by the robots en route to creating the METS metadata

See also the various forms (Atom, RDF, XML) of the ORE Resource Map. These are not simple models, but may be a form of output (along with METS) that we will need to produce from our data someday (aka "just in time").

A snippet of the two models highlights the differences:

- Content Map
  - Has object level identifying data
  - Embeds child object in strongly typed container (<page>)
  - Could vary somewhat in details across types of data (i.e. maybe not generic enough)
  - designed for output, possibly one of several forms of output
  - could vary its content for an audience (e.g., file-locaiton info for internal shelving robot)

```
<contentMap id="dr:xx123yy1234" type="pageMap">

   <description>
      <title>The title from the DC datastream<title>
      <author>The author from the DC datastream<author>
      <identifier>catkey:36105036338734</identifier>
  </description>

   <page sequence="1" id="dr:wk847bs9372">
      <pageType>Title</pageType>
      <file datastream="image" format="image/jp2" checksum="3218728144" height="800" width="1200" url="http://service/druid/image1.jp2" />
      <file datastream="text" format="text/plain" checksum="3827715263" url="http://service/druid/image1.txt" />
   </page>

   <page ...
```

- fileAggregationList
  - only file-specific data, including grouping
  - generic substructure
  - contains more complete

```
<om:fileAggregateList name="pages">

   <om:fileAggregate type="Page" druid="dr:qg126rs0427" subTypes="FRONT_COVER,IMAGE_ON_PAGE,IMPLICIT_PAGE_NUMBER">
      <om:file id="FILE_00001_dr_qg126rs0427.jp2" groupid="MasterJPEG2000" format="JPEG2000" path="dr_nf452zq6060/00001_dr_qg126rs0427.jp2"
            size="169627" created="2007-08-28T00:00:00" checksum="60dd37697f05073168b9b58ddaccad0a"
            originalPath="36105061216029/00000001.jp2" admid="AMD_FILE_00001_dr_qg126rs0427.jp2 C1"/>
      <om:file id="FILE_00001_dr_qg126rs0427.html" groupid="HOCR" format="HTML" path="dr_nf452zq6060/00001_dr_qg126rs0427.html"
            size="734" created="2007-08-28T00:00:00" checksum="8884e49cdac7065a4333733f54b85f62"
            originalPath="36105061216029/00000001.html" admid="AMD_FILE_00001_dr_qg126rs0427.html "/>
   </om:fileAggregate>

   <om:fileAggregate ...
```

Sample parallel METS fileSec and structMap XML

```
<mets:fileSec ID="FS_dr_jf822ps0564">
      <mets:fileGrp USE="Archive Masters">
         <mets:file ID="FILE_00001_dr_gd459cw5199.jp2" GROUPID="MasterJPEG2000"
                    MIMETYPE="image/jp2"
                    SIZE="168370"
                    CREATED="2006-12-12T00:00:00"
                    CHECKSUM="20b569803bb407bc352c9d57a594cd7c"
                    CHECKSUMTYPE="MD5"
                    ADMID="AMD_FILE_00001_dr_gd459cw5199.jp2 C1"
                    OWNERID="36105049267078/00000001.jp2">
            <mets:FLocat LOCTYPE="URL" xlink:type="simple"
                         xlink:href="dr_jf822ps0564/00001_dr_gd459cw5199.jp2"/>
         </mets:file>
       </mets:fileGrp>
      <mets:fileGrp USE="OCR creation">
         <mets:file ID="FILE_00001_dr_gd459cw5199.html" GROUPID="HOCR" MIMETYPE="text/html"
                    SIZE="4426"
                    CREATED="2006-12-12T00:00:00"
                    CHECKSUM="1f95ba856efb57d17ab9a0ac46fe0ee6"
                    CHECKSUMTYPE="MD5"
                    ADMID="AMD_FILE_00001_dr_gd459cw5199.html "
                    OWNERID="36105049267078/00000001.html">
            <mets:FLocat LOCTYPE="URL" xlink:type="simple"
                         xlink:href="dr_jf822ps0564/00001_dr_gd459cw5199.html"/>
         </mets:file>
      </mets:fileGrp>
      <mets:fileGrp USE="Container for Data Provider Source Files">
         <mets:file ID="FILE_dr_jf822ps0564_auxiliary.tar" GROUPID="AuxiliaryTAR"
                    MIMETYPE="application/octet-stream"
                    SIZE="368640"
                    CREATED="2009-04-13T15:06:32-07:00"
                    CHECKSUM="a2ac8907ce0fd1bd20a16b5ee3e02875"
                    CHECKSUMTYPE="MD5"
                    ADMID="AMD_FILE_dr_jf822ps0564_auxiliary.tar">
            <mets:FLocat LOCTYPE="URL" xlink:type="simple"
                         xlink:href="dr_jf822ps0564/dr_jf822ps0564_auxiliary.tar"/>
         </mets:file>
      </mets:fileGrp>
   </mets:fileSec>
            :
     <mets:structMap TYPE="Physical" ID="SM_dr_jf822ps0564">
      <mets:div TYPE="simpleObject" ADMID="AMD_ObjectLevel_01" ID="simpleObjectLevel_01"
                ORDER="1"
                ORDERLABEL="Current version">
         <mets:div TYPE="sourceFileAggregation">
            <mets:fptr FILEID="FILE_dr_jf822ps0564_auxiliary.tar"/>
         </mets:div>
         <mets:div TYPE="digitalObjectAggregate" ID="Top01" DMDID="DMD_dr_jf822ps0564_MODS">
            <mets:div ID="Level02" TYPE="documentEntire">
               <mets:div TYPE="Page" ORDER="1" CONTENTIDS="dr:gd459cw5199" ID="Level03_Page1">
                  <mets:fptr FILEID="FILE_00001_dr_gd459cw5199.jp2"/>
                  <mets:fptr FILEID="FILE_00001_dr_gd459cw5199.html"/>
               </mets:div>
               <mets:div TYPE="Page" ORDER="2" CONTENTIDS="dr:yk428qf6809" ID="Level03_Page2">
                  <mets:fptr FILEID="FILE_00002_dr_yk428qf6809.jp2"/>
                  <mets:fptr FILEID="FILE_00002_dr_yk428qf6809.html"/>
               </mets:div>
                     :
```

Add Comment