Preservation Metadata in Fedora

A.R.D. Prasad
Documentation Research and Training Centre,
Indian Statistical Institute,
Bangalore

Death strikes each day,

we behave as we are immortal

-- Jataka Stroy

Nothing is permanent (Kshanabanguravada)

-– Buddha

Preservation Metadata

Preservation metadata spans-

- administrative,
- technical, and
- structural metadata elements

with emphasis on-

- Rights,
- Digital Provenance, and
- Relationships among different objects within the preservation and other factors.

PREMIS with METS

- One of the probable best practices for Digital Preservation Metadata is a symbiosis between METS and PREMIS
- These two metadata standards are known to possess many of the desirable features for preservation of digital content.

METS- Metadata Encoding and Transmission Standard

- METS provides a means to convey the metadata necessary for
  - both the management of digital objects within a repository and
  - exchange of such objects between repositories
  - or between repositories and their users.

METS Structure

- A METS record has different sections for various types of metadata elements categorized as-
  - Structural,
  - Descriptive, and
  - Administrative metadata
  - Behavior Metadata, a means to link digital content with applications or computer programming code

Structural Metadata

- Structural Metadata
  - describes,
  - anchors and
  - organizes

the components of a digital object together.

- It helps to retain the integrity of the digital object, even when its components are stored in different places.

Structural Metadata...

- Tools meant to render and display the object (e.g.: a page turning software) can also make use of structural metadata.
- The Structural metadata of the METS document constitute of
  - Structural Map,
  - Files Section, and
  - Structural Links

Descriptive Metadata

- The descriptive metadata section <dmdSec> records metadata associated with the digital object encoded with METS, to facilitate **discovery** of that object.
- Elements of a selected descriptive metadata standards can be inserted within this section.
- The chosen metadata can be either embedded within the METS document; or
- Alternatively it can be stored in a separate file, and a reference can be given to its location.

Descriptive Metadata...

Descriptive metadata can be expressed according to many current content standards viz-

- MARC,
- MODS,
- Dublin Core,
- TEI Header,
- VRA,
- or a locally produced XML schema.

Administrative Metadata

- The Administrative Metadata Section <amdSec> contains the administrative metadata pertaining to the
  - digital object,
  - its component data files and
  - any original source material from which the digital object is derived.

Administrative Metadata …

- The <amdSec> is separated into four sub-sections
  - Technical metadata <techMD>,
  - Intellectual Property Rights metadata <rightsMD>,
  - Source metadata <sourceMD>, and
  - Provenance metadata <digiprovMD

Administrative Metadata...

- METS does not define a vocabulary or syntax for encoding administrative metadata.
- Administrative metadata can be expressed within a METS file according to any community current content standard or a locally produced XML schema.
- It can also be stored externally in another file and can be referenced.

Administrative Metadata...

The administrative metadata schemes endorsed by the METS Board include:

- NISO Technical Metadata for Digital Still Images (NISOIMG)
- TextMD: Schema for Technical Metadata for Text
- MIX XML Schema
- LC-AV (Audio / Video Technical Metadata Schema)
- METS RightsMD Schema

PREMIS- Preservation Metadata Implementation Strategies

- PREMIS comprises of a 'core' set of preservation metadata elements (Data Dictionary) along with strategies for
  - encoding,
  - packaging,
  - storing,
  - managing, and
  - exchanging the metadata.

PREMIS Data Model

- The PREMIS working group developed a simple data model to organize the concepts defined in the Data Dictionary.
- This Data model includes
  - Entities,
  - Properties of Entities (semantic units),
  - Relationships between Entities.
- The container element <premis> may be used to keep all PREMIS metadata together.

PREMIS in the METS Container

- Much of the current thinking on repositories in digital preservation community is based on OAIS (Open Archival Information System) reference model.
- The PREMIS Framework is an elaboration of the OAIS information model, explicated through the mapping of preservation metadata
- So the combination of METS and PREMIS can be a good strategy for preservation metadata.

PREMIS in the METS Container…

- METS Board endorses PREMIS Data Dictionary for administrative Metadata.
- METS and XML-encoded PREMIS could be combined in several ways.
  - Either the whole PREMIS can be embedded as a single unit in METS <amdSec>

OR

  - PREMIS data can be distributed among the subsections of METS <amdSec>.

PREMIS in the METS Container…

- While using all PREMIS units together, the entire package goes in the Provenance metadata <digiProvMD> with the <premis> element as a container.

- But there might be advantages splitting data for maintenance and reuse purposes.

PREMIS in the METS Container…

- PREMIS first level data elements should be used in the METS sections as described below :
  - Object entity in technical Metadata section <techMD> or in Provenance metadata <digiProvMD>
  - Event entity in Provenance metadata <digiprovMD>
  - Agent entity in <digiprovMD> or <rightsMD>
  - Rights entity can be used in Intellectual Property Rights metadata <rightsMD> section.

Object entity in technical Metadata section <techMD>

```
<mets:amdSec>
 <mets:techMD ID="object1">
  <mets:mdWrap MDTYPE="PREMIS:OBJECT">
   <mets:xmlData>
    <premis:object xsi:type="premis:file"xsi:
     schemaLocation="info:lc/xmlns/premis-v2
     http://www.loc.gov/standards/premis/v2/
     premis-v2-0.xsd">

     <premis:objectIdentifier>
        <premis:objectIdentifierType>hdl
        </premis:objectIdentifierType>
        <premis:objectIdentifierValue>
         http://hdl.handle.net/1849/267
        </premis:objectIdentifierValue>
      </premis:objectIdentifier>
        ..................
    </premis:object>
   </mets:xmlData>
  </mets:mdWrap>
 </mets:techMD>
</mets:amdSec>
```

Event entity in Provenance metadata <digiprovMD>

```
<mets:amdSec>
 <mets:digiprovMD ID="event1">
  <mets:mdWrap MDTYPE="PREMIS:EVENT">
   <mets:xmlData>
    <premis:event>
     <premis:eventIdentifier>
      <premis:eventIdentifierType>
      FedoraRepository
      </premis:eventIdentifierType>
      <premis:eventIdentifierValue>e-
      ing0211</premis:eventIdentifierValue>
      </premis:eventIdentifier>
      <premis:eventType>ingestion </premis:eventType>
      <premis:eventDateTime> 2008-02-12T01:12:03.004
      </premis:eventDateTime>
.............
    </premis:event>
   </mets:xmlData>
  </mets:mdWrap>
 </mets:digiprovMD>
</mets:amdSec>
```

Agent entity in <digiprovMD> or <rightsMD>

```
<mets:amdSec>
 <mets:digiprovMD ID="agent1">
  <mets:mdWrap MDTYPE="PREMIS:AGENT">
   <mets:xmlData>
    <premis:agent>
        ............
      <premis:agentName>Prasad, A.R.D.</premis:agentName>
      <premis:agentType>person</premis:agentType>
    </premis:agent>
   </mets:xmlData>
  </mets:mdWrap>
 </mets:digiprovMD>
</mets:amdSec>
```

Similarly, Rights entity can be used in Intellectual Property Rights metadata <rightsMD> section, though it is not so common

Disadvantages of Single METS File

- Encoding all the metadata about an archive in a single METS file will result in a complex METS file with enormous size.
- It will be difficult to work with it especially when new Representations, Files, events and agents are created through migrations.
- Additionally it demands careful planning and control over the assignment of identifiers and linking mechanisms.
- A better approach is to split up the archive into numerous, smaller, METS files and use the linking mechanisms in METS and PREMIS to connect these.

Smaller METS Files Linked

- Separate METS files can be created for:
  - Intellectual Entities,
  - Representations of Intellectual Entities,
  - Files,
  - Agents,
  - Events and
  - Rights.
- This helps to reduce METS file sizes
- It avoids unnecessary duplication of metadata.

PREMIS Relationships in METS

- Even though structural relationships are detailed in the METS Structural Map, PREMIS relationships can also be used for both structural and derivative relationships.

- PREMIS allows relationships between all kinds of objects.

- Event and Agent information can also be stored with a relationship in PREMIS.

- It can link to objects outside of current xml-file.

Where as limited options are available with METS alone.

PREMIS and METS: Issues to be Addressed

- How to deal with overlaps between the schema?
  - for example the overlap between METS attributes, PREMIS elements, descriptive metadata elements and format specific schemes used in technical Metadata section.
- Whether to use ID and IDREFS or embedded identifiers or nested XML elements to tie different sections together?
- Ambiguity regarding the METS sections in which to embed the various PREMIS entities etc…..

PREMIS in METS: Real World Implementations

- There are a few projects implemented the PREMIS in METS combination. Some are:
  - MathArc (Ensuring Access to Mathematics Over Time)
  - Paradigm (Personal Archives Accessible in Digital Media)
  - The ECHO DEPository (The Exploring Collaborations to Harness Objects in a Digital Environment for Preservation)

FEDORA ...

- In addition to digital content item and its metadata in any format, the Fedora repository stores the relationships between digital objects stored

- It also provides functionality to store just the metadata and relationships for the content, which is held either locally or by another organization or system.

- Fedora's flexible and open architecture presents a technological solution for the Digital Preservation dilemmas.

Features of Fedora that support digital preservation

- Powerful digital object model
- Extensible metadata management
- Expressive inter-object relationships
- Web services integration
- Version management
- Configurable security architecture
- OAI-PMH conformance
- Preservation worthy

Preservation Metadata Management: The FEDORA Way

- An ideal digital archiving package should be capable of supporting activities like
  - Metadata editing,
  - Import/ Export,
  - Schema validation and
  - Preserving the relationship among various metadata schema, services and objects.
- These are necessary for the long-time preservation and maintenance of digital objects; and to support interoperability.

Digital Object and Datastream in FEDORA

- A **digital object** is basic unit for information aggregation in Fedora and basically consists of:
  - An identifier or PID (Persistence Identifier); and
  - Metadata that provides the basic description of the digital object.(By default DC)
- A **Datastream** is a component of a digital object that represents data source.(metadata, relationship, etc.)

- A digital object can have just the basic Dublin core metadata set or it can have any number of datastreams associated with it.

Metadata Management in FEDORA

- The metadata datastreams can be associated and managed locally or by some externally referenced data source .

Metadata Management in FEDORA...

1. Internal Management

- Stores a name-spaced block of XML content within the Fedora digital object XML file.
- It is restricted to a well-formed XML, as it will be stored inline with the part of XML structure of the digital object.
- This is the best way to associate metadata with the datastream as in this case the datastream content is stored as part of the XML structure of the digital object itself.
- Thus it is included when the digital object is exported which further helps in archiving.

Metadata Management in FEDORA...

2. Managed datastreams

- Metadata is stored with a particular digital object inside the repository.
- The XML restrictions are not that much with this type of metadata association and the content can be of any mime type.

3. External referenced datastreams

- Third approach of managing metadata is importing it from external source just by providing a reference URL.
- The fetched metadata is stored inside the repository with the digital object.

Metadata Management in FEDORA...

4. Redirect datastreams

- Here, the metadata is not stored inside the repository but at a web accessible location.
- The reference is given by providing the URL of the metadata file

Metadata Management in FEDORA…

5. By using some external or local Service

- A number of external or local services can be associated with Fedora. e.g., Metadata crosswalks.

One could associate an XSLT engine (e.g. Saxon) service with the digital object that processes the base metadata format with a transform XSL document (packaged as a datastream in another digital object) to derive one or more additional formats.

Metadata Management in FEDORA…

- In all of the mentioned methods, XML files containing PREMIS elements in METS container can be associated.
- For Internal XML Metadata an XML schema definition is required for proper validation as it requires formal rules/restrictions

Conclusions

- Fedora employs XML and METS to create digital objects by encapsulating content, along with metadata about the content,
  - and actions that can be performed on the content.

- This linkage of the content to the applications that are used to search, render, and display it, distinguishes Fedora from other digital repository systems.
- Clients interact with the Fedora repository through an API that provides management and access services.

Conclusions

- Comparatively, core Fedora doesn't have much support to many of the routine activities such as:
  - workflow management,
  - access control,
  - self-archiving,
  - web based submission modules,
  - role management, etc.
- Various third party tools are available for Fedora to perform almost all of the above tasks.
  - e.g.: Fez, Elated, Valet, Muradora, etc

Conclusions

- Comparatively, core Fedora doesn't have much support to many of the routine activities such as:
  - workflow management,
  - access control,
  - self-archiving,
  - web based submission modules,
  - role management, etc.
- Various third party tools are available for Fedora to perform almost all of the above tasks.
  - e.g.: Fez, Elated, Valet, Muradora, etc

Thank YOu

ard@drtc.isibang.ac.in