Digital Preservation Metadata and Improvements to PREMIS in Version 3.0

Angela Dappert
University of Portsmouth
Agenda

- Digital preservation metadata
  - Why is it needed and what does it look like?
- PREMIS
  - What is it?
  - Data model
  - How to use it
- From V2 to V3
Agenda

- Digital preservation metadata
  - Why is it needed and what does it look like?
- PREMIS
  - What is it?
  - Data model
  - How to use it
- From V2 to V3
What is digital preservation metadata?

- Digital preservation metadata = Metadata to ensure long-term accessibility of digital resources

- Digital objects must be self-descriptive

- Must be able to describe, manage and discover independently from the systems that were used to create them
  
  XML (machine and human readable)
DP metadata supports preservation goals

Preservation Pyramid
(from Priscilla Caplan)
Domain

Born digital

Digitized
Technology dependence

- Not self-descriptive
- Complex formats
- Complex environments
- No direct access
Technology dependence

**Metadata:**
- Format information
- Rendering information
  - Software
  - Hardware
- Other dependencies: schemas, style sheets, encodings, etc.
Complex structures

Metadata
• Physical structural relationships
  • Embedded files
  • File sequence
• Logical structural relationships
Supporting features

**Metadata:**
Semantic information for the designated community
Supporting features

Example uses and queries

Metadata:
Semantic information for the designated community
Context descriptions

**Metadata:**
Context descriptions

- Original source
- Related items (e.g. migration source)
Obsolescence

-> object transformations

- Pre-emptive preservation actions
  - Bit migration
  - Content migration
  - Replacing part of the rendering stack
- Forensic transformation actions
Obsolescence / object transformations

Goals

- Avoid rights violations
- Prove authenticity

Metadata

- Rights information for preservation actions during copyright / license period
- Provenance metadata:
  - History of all actions performed on the resource
  - History of custodianship

- Events
- Dates
- Changes and decisions
- Agents (decision maker + tools used)
## Obsolescence / object transformations

### Goals
- Manage potential loss of object characteristics
- Demonstrate degree of authenticity
- Explain decisions
  - Documentation

### Metadata
- Significant characteristics
- Lost characteristics
- Business rules (policy, strategy) guiding preservation actions
Mutability

- Intentional or accidental change
- Decay: rapid and potentially complete

Goals

- Viability: the object is readable
- Fixity: the object is unchanged

Metadata

- Data carrier metadata
  - Type of medium
  - Its preservation characteristics
  - Age of medium
  - Date of recording
  - Usage patterns
- Checksums, message digests, hash function
- Event creating them
  - Algorithms creating them
  - Date/time
  - Originator
Mutability

- Intentional or accidental change
- Decay: rapid and potentially complete

Goals

- Integrity: the object is whole and unimpaired
- Authenticity: the object is what it purports to be

Metadata

- Event information for format identification and validation events (= provenance)
- Structural metadata
- Digital signatures
- Access rights
Agenda

- Digital preservation metadata
  - Why is it needed and what does it look like?
- PREMIS
  - What is it?
  - Data model
  - How to use it
- From V2 to V3
The PREMIS standard

- International *de-facto* standard for metadata to support the preservation of digital objects and ensure their long-term usability.
  - Information you need to know for preserving digital objects
    - *Preservation Metadata: Implementation Strategies*
- Developed by an international team of experts.
- Implemented in digital preservation projects around the world.
- Incorporated into commercial and open-source digital preservation tools and systems.
The PREMIS standard

- Data Dictionary (PREMIS 2.2)
  - Version 3 will be released this summer – major release
- XML schema v2.3
- OWL ontology
- Supporting documentation
Activities

- The PREMIS Editorial Committee
  - Coordinates revisions and implementation of the standard

- PREMIS Implementors' Group forum (pig@loc.gov)
  - Email message to listserv@loc.gov:
    Text: subscribe pig <your name>

- PREMIS Implementation Fair (PIF)
  - User group meetings (@iPres)
Scope

- What PREMIS DD is:
  - Common data model for organizing/thinking about preservation metadata
  - Standard for exchanging information packages between repositories
  - Implementable
  - Technically neutral
  - Core metadata
Scope

- What PREMIS DD is not:
  - Out-of-the-box solution
  - All needed metadata
  - Lifecycle management of objects outside repository
    - increasing support for integration with outside
  - Rights management standard
    - strong support for rights statements
Agenda

- Digital preservation metadata
  - Why is it needed and what does it look like?
- PREMIS
  - What is it?
    - Data model
  - How to use it
- From V2 to V3
Data Model in PREMIS Version 2

- Entities: “things” relevant to digital preservation that are described by preservation metadata
- Relationships between Entities
- Properties of Entities (semantic units)
Example: Object Entity semantic units

- 1.1 object Identifier
- 1.2 object Category
- 1.3 preservation Level
- 1.4 significant Properties
- 1.5 object Characteristic
  - 1.5.1 compositionLevel
  - 1.5.2 fixity
  - 1.5.3 size
  - 1.5.4 format
  - 1.5.5 creatingApplication
  - 1.5.6 inhibitors
- 1.6 original Name
- 1.7 storage
- 1.8 environment
- 1.9 signature Information
- 1.10 relationship
- 1.11 linkingEventIdentifier
- 1.13 linkingRightsStatementIdentifier
## Sample Data Dictionary Entry

<table>
<thead>
<tr>
<th>Semantic unit</th>
<th>size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Semantic components</td>
<td>None</td>
</tr>
<tr>
<td><strong>Definition</strong></td>
<td>The size in bytes of the file or bitstream stored in the repository.</td>
</tr>
<tr>
<td><strong>Rationale</strong></td>
<td>Size is useful for ensuring the correct number of bytes from storage have been retrieved and that an application has enough room to move or process files. It might also be used when billing for storage.</td>
</tr>
<tr>
<td><strong>Data constraint</strong></td>
<td>Integer</td>
</tr>
<tr>
<td><strong>Object category</strong></td>
<td>Representation</td>
</tr>
<tr>
<td><strong>Applicability</strong></td>
<td>Not applicable</td>
</tr>
<tr>
<td><strong>Examples</strong></td>
<td>2038927</td>
</tr>
<tr>
<td><strong>Repeatability</strong></td>
<td>Not repeatable</td>
</tr>
<tr>
<td><strong>Obligation</strong></td>
<td>Optional</td>
</tr>
<tr>
<td><strong>Creation/Maintenance notes</strong></td>
<td>Automatically obtained by the repository.</td>
</tr>
<tr>
<td><strong>Usage notes</strong></td>
<td>Defining this semantic unit as size in bytes makes it unnecessary to record a unit of measurement. However, for the purpose of data exchange the unit of measurement should be stated or understood by both partners.</td>
</tr>
</tbody>
</table>
Agenda

- Digital preservation metadata
  - Why is it needed and what does it look like?
- PREMIS
  - What is it?
  - Data model
    - How to use it
- From V2 to V3
Evolving metadata
- Increasing experience ensuring the longevity of digital objects
- Changing future technical possibilities
- Changing future legal framework

Tayloring solutions
- Varying needs
  - Content-types
  - Institutional policies
  - Intended use
From here to an implementation ...
Example: Document in 3 representations

Intellectual Entity
2015-05-27-Presentation
<descriptive metadata>

Derivation/
has source

Structural/
includes

File 1
2015-05-27-
Presentation.pptx

Event

Agent

Structural/
includes

File 2
2015-05-27-
Presentation.pdf

Structural/
has sibling

File 3
2015-05-27-Notes.docx
objectIdentifier
    objectIdentifierType: ARK
    objectIdentifierValue: ark:/12148/cb37367035f
    objectCategory: intellectual entity

objectIdentifier
    objectIdentifierType: ARK
    objectIdentifierValue: ark:/9999/h1.version1
    objectCategory: representation

relationship
    relationshipType: derivation
    relationshipSubType: has source
    relatedObjectIdentifier
        relatedObjectIdentifierType: ARK
        relatedObjectIdentifierValue: ark:/9999/h1.version1

objectIdentifier
    objectIdentifierType: ARK
    objectIdentifierValue: ark:/9999/h1.version2
    objectCategory: representation
Tayloring PREMIS to needs

- **Evolving metadata**
  - Increasing experience ensuring the longevity of digital objects
  - Changing future technical possibilities
  - Changing future legal framework

- **Tayloring solutions**
  - Varying needs
    - Content-types
    - Institutional policies
    - Intended use
  - Off-the-shelf (OS / commercial) or custom-built

- **Off-the-shelf systems**
  - Predefined metadata profiles
  - Out-of-the-box tools

- **Configured, extended, adapted**
  - Metadata profiles and tools

- **Custom-built systems**
  - Metadata profiles and tools
Agenda

- Digital preservation metadata
  - Why is it needed and what does it look like?
- PREMIS
  - What is it?
  - Data model
  - How to use it
- From V2 to V3
PREMIS: From V2 to V3

- Next major version of the PREMIS Data Dictionary
- Released by July 2014 (hopefully 😊)
- Proof-reading phase
PREMIS: From V2 to V3

- Improving PREMIS based on user needs
- Add preservationLevelType semantic unit
- Add agentVersion semantic unit
- Add “unknown” values
- Add eventDetailInformation semantic unit
- Add authority for controlled vocabulary
- Make Intellectual Entity an Object category
- Make Environments independent Objects
- Add physical Objects
- Update conformance statement
Approved Changes:
Add eventDetailInformation semantic unit.

- 2.1 eventIdentifier
- 2.2 eventType
- 2.3 eventDateTime
- 2.4 eventDetail
- 2.5 eventOutcomeInformation
- 2.6 linkingAgentIdentifier
- 2.7 linkingObjectIdentifier
Approved Changes:
Add eventDetailInformation semantic unit.

- 2.1 eventIdentifier
- 2.2 eventType
- 2.3 eventDateTime
- 2.4 eventDetailInformation
  - 2.4.1 eventDetail
  - 2.4.2 eventDetailExtension
- 2.5 eventOutcomeInformation
- 2.6 linkingAgentIdentifier
- 2.7 linkingObjectIdentifier
PREMIS: From V2 to V3

- Improving PREMIS based on user needs
- Add preservationLevelType semantic unit
- Add agentVersion semantic unit
- Add “unknown” values
- Add eventDetailInformation semantic unit
- Add authority for controlled vocabulary
- Make Intellectual Entity an Object category
- Make Environments independent Objects
- Add physical Objects
- Update conformance statement
Implementation specific change:
Add authority for controlled vocabulary

- **eventIdentifier**: 
  - eventIdentifierType: UUID
  - eventIdentifierValue: 908985d3-9600-4da4-a7e7-c6e9508bf24c
- **eventType**: validation
- **eventDateTime**: 2014-07-03T23:18:19
- **eventDetailInformation**: 
  - eventDetail: program="Jhove"; version="1.5"
- **eventOutcomeInformation**: 
  - eventOutcome: fail
  - eventOutcomeDetail: 
    - eventOutcomeDetailNote: format="JPEG"; version="1.02"; result="Not well-formed"
PREMIS: From V2 to V3

- Improving PREMIS based on user needs
- Add preservationLevelType semantic unit
- Add agentVersion semantic unit
- Add “unknown” values
- Add eventDetailInformation semantic unit
- Add authority for controlled vocabulary
- Make Intellectual Entity an Object category
- Make Environments independent Objects
- Add physical Objects
- Update conformance statement
Approved Changes: Make Intellectual Entity an Object category

- A set of content that is considered a single intellectual unit for purposes of management and description
- For example, a particular book, map, photograph, or database.

V2:
- Assumed to be held in a container metadata schema
- No Intellectual Entity semantic units
- Exception: identifier to enable linking to a description
- PREMIS Objects link to it.
Approved Changes: Make Intellectual Entity an Object category

V3:
- Possibility to describe preservation aspects of intellectual entities
- Same semantic units as Representations
Approved Changes:
Make Intellectual Entity an Object category

- Relate to PREMIS Events and RightsStatements.
- Support structural and derivative relationships with Objects.
- Represent an aggregate, such as a collection, FRBR work, FRBR expression, fonds or series.
- Capture versioning information and metadata update events at the Intellectual Entity level.
- Associate business requirements with them.
  - Significant characteristics, risk definitions, guidelines for preservation actions, etc.
Approved Changes: Make Environments independent Objects

- What is needed to render or use an object
  - Operating system
  - Application software
  - Hardware
  - Computing resources

- A high-level data model
- **No** detailed characteristics specific to an environment type
Example: Environment stack and dependency relationships

- Modularised environment aggregates as a network
- Re-usable and distributed environment descriptions
  - across different Objects
  - across repositories and registries

**Diagram:**
- **Registry 1**
  - **Repository**
  - **Software application**
    - **File 1**
  - **Operating system**
  - **Software library**
    - **File 2**
  - **Hardware architecture**
    - **Hardware peripheral**
      - **Software driver**

**Relationships:**
- relationshipType: dependency
- relationshipSubType: requires
Data Model in PREMIS V2

Object (including Environment semantic unit container)

Intellectual Entity

Rights

Agent

Event
Data Model in PREMIS V3
Example: An object and its rendering environment

represents = relationshipType: structural relationshipSubType: represents

requires = relationshipType: dependency relationshipSubType: requires
1. **Object to environment** - specify computational context
2. **environment to Object** - documentation, specifications, surrogates
3. **environment to environment** - inclusion, dependency, derivation, other
4. **environment is an Object** - preserved software source code
5. **Agent to Environment** - role of an Agent
6. **environment to Event** - environment specific Events (provenance)
7. **environment to RightsStatement** - software license, policy

“Object”: here a traditional content Object
Expanded relationship types for environment Objects

- **Dependency**
  - Requires, is required by
  - Is deployed on

- **Derivation**
  - Is source of, has source

- **Logical**
  - generalises, is generalised by

- **Reference**
  - Documents, is documented in

- **Replacements**
  - Supercedes, is superceded by

- **Structural**
  - Includes, is included in
  - Represents, is represented as
Semantic units only applicable to environment Intellectual Entities

1.9 environmentFunction

- environmentFunctionType
- environmentFunctionLevel

objectIdentifier
  - objectIdentifierType: ARK
  - objectIdentifierValue: ark:/9999/b1
- objectCategory: intellectual entity

environmentFunction
  - environmentFunctionType: software
  - environmentFunctionLevel: 1

environmentFunction
  - environmentFunctionType: operating system
  - environmentFunctionLevel: 2

XP Professional, Service Pack 3
Semantic units only applicable to environment Intellectual Entities

1.9 environmentFunction
- environmentFunctionType
- environmentFunctionLevel

1.10 environmentDesignation
- environmentName
- environmentVersion
- environmentOrigin
- environmentDesignationNote
- environmentDesignationExtension

**objectCategory**: intellectual entity
**environmentFunction**
  - environmentFunctionType: software
  - environmentFunctionLevel: 1
**environmentFunction**
  - environmentFunctionType: operating system
  - environmentFunctionLevel: 2

**environmentDesignation**
  - environmentName: Windows XP Professional
  - environmentVersion: Service Pack 3
  - environmentDesignationNote:
    - maintenance deadline: 2014-04
Semantic units only applicable to environment Intellectual Entities

1.9 environmentFunction
   - environmentFunctionType
   - environmentFunctionLevel

1.10 environmentDesignation
   - environmentName
   - environmentVersion
   - environmentOrigin
   - environmentDesignationNote
   - environmentDesignationExtension

1.11 environmentRegistry
   - environmentRegistryName
   - environmentRegistryKey
   - environmentRegistryRole

objectCategory: intellectual entity
environmentFunction
   environmentFunctionType: software
   environmentFunctionLevel: 1
environmentFunction
   environmentFunctionType: operating system
   environmentFunctionLevel: 2
environmentDesignation
   environmentName: Windows XP Professional
   environmentVersion: Service Pack 3
environmentRegistry
   environmentRegistryName: PRONOM
   environmentRegistryKey: x-sfw/8
   environmentRegistryRole: identity
Semantic units only applicable to environment Intellectual Entities

1.9 environmentFunction
   - environmentFunctionType
   - environmentFunctionLevel

1.10 environmentDesignation
   - environmentName
   - environmentVersion
   - environmentOrigin
   - environmentDesignationNote
   - environmentDesignationExtension

1.11 environmentRegistry
   - environmentRegistryName
   - environmentRegistryKey
   - environmentRegistryRole

x-sfw/8
Description of Windows XP Professional in PRONOM

relationshipType: dependency
relationshipSubType: requires
relatedEnvironmentPurpose: render
relatedEnvironmentCharacteristic: recommended
relatedObjectIdentifier
   relatedObjectIdentifierType: PUID
   relatedObjectIdentifierValue: x-sfw/8

Alternative: Link to an external registry
Semantic units only applicable to environment Intellectual Entities

- 1.9 environmentFunction
  - environmentFunctionType
  - environmentFunctionLevel

- 1.10 environmentDesignation
  - environmentName
  - environmentVersion
  - environmentOrigin
  - environmentDesignationNote
  - environmentDesignationExtension

- 1.11 environmentRegistry
  - environmentRegistryName
  - environmentRegistryKey
  - environmentRegistryRole

- 1.12 environmentExtension

- 1.13 relationship
  - relatedEnvironmentPurpose
  - relatedEnvironmentCharacteristic
objectCategory: intellectual entity
environmentFunction
  environmentFunctionType: software application

Firefox 10.0

relationshipType: dependency
relationshipSubType: requires
relatedEnvironmentPurpose: render
relatedEnvironmentCharacteristic: known to work

BlueGriffon 1.6

relationshipType: dependency
relationshipSubType: requires
relatedEnvironmentPurpose: create

Content Object
formatName: text/html

1.13 relationship
  ...
  relatedEnvironmentPurpose
  relatedEnvironmentCharacteristic
Approved Changes:
Add physical Objects

- A physical Object is
  - A content Object, such as a manuscript, or printed document
  - An environment Object, such as a physical hardware device.

- Representation: A digital or physical Object
  - Either one instantiates or embodies an Intellectual Entity

- Digital and non-digital Objects can be captured uniformly.
  - Physical Objects can relate to digital Objects and other physical Objects.

- In V3 storage is applicable to Representations.
  For physical Representations: the physical location, e.g. a shelf location.
Approved Changes:
Add physical Objects

objectIdentifier
  objectIdentifierType: ARK
  objectIdentifierValue::ark:/12148/cb37367035f
  objectCategory: intellectual entity

relationshipType: structural
  relationshipSubType: is represented as

[Physical representation]

relationshipType: derivation
  relationshipSubType: has source
  relatedObjectIdentifier
    relatedObjectIdentifierType: Internal call number
    relatedObjectIdentifierValue: Rés.Ye-3535

objectIdentifier
  objectIdentifierType: ARK
  objectIdentifierValue: ark:/9999/h1.version1
  objectCategory: file
  format
    formatDesignation
      formatName: image/tiff
      formatVersion: 6.0
PREMIS: From V2 to V3

- Improving PREMIS based on user needs
  - Add preservationLevelType semantic unit
  - Add agentVersion semantic unit
  - Add “unknown” values
  - Add eventDetailInformation semantic unit
  - Add authority for controlled vocabulary
  - Make Intellectual Entity an Object category
  - Make Environments independent Objects
  - Add physical Objects
  - Update conformance statement

Thank you!

- Resources: http://www.loc.gov/standards/premis/
- PREMIS Implementors Group Forum: PIG@listserv.loc.gov