Describing Digital Object Environments in PREMIS

Angela Dappert  
Digital Preservation Coalition

Sébastien Peyrard  
National Library of France

Janet Delve  
The University of Portsmouth

Carol C.H. Chou  
Florida Digital Archive
Environment

- Software
- Hardware
- A format
- A document
  - A policy document
  - A manual
  - Documentation
- A cheat sheet
- A user behaviour study
- A process
- “Other representation information”
Goal

- High-level data model for Environments
- Capture the required relationships to other DP entities
- Capture desirable characteristics
- Standardized way of treating Environments
- Information sharing / exchange
- Repositories and registries

- Not:
  - modelling the internals of a given Environment category – as e.g. TOTEM
Guidelines

- Backward compatibility
- Compliance with OAIS
- Straightforward semantics
  - easy to implement
- Clear mapping of historic to new Environment features
  PREMIS 2 -> PREMIS 3
Requirements

- Environments may be digital or non-digital
- Environments may be generic or instances (abstract description or concrete digital object)
- Environments may be tools or services
- Environments have no simple software / hardware distinction (Virtual machines blur the distinction)
The PREMIS Data Model

Slide by S. Peyrard
Example: Object Entity

Main types of information
- identifier
- technical object characteristics
- creation information
- software and hardware environment
- digital signatures
- relationships to other objects
- links to other types of entity
PREMIS – Environment Metadata

1.5.5 creatingApplication

1.5.5.1 creatingApplicationName
1.5.5.2 creatingApplicationVersion
1.5.5.3 dateCreatedByApplication
1.5.5.4 creatingApplicationExtension
Gap Analysis

- OAIS focus on Object:
  - Creating Applications are Environments
Semantic Unit: Environment

- What is needed to render or use an object
  - Operating system
  - Application software
  - Computing resources
PREMIS – Environment Metadata

1.8 **environment**
   1.8.1 environmentCharacteristic
   1.8.2 environmentPurpose
   1.8.3 environmentNote

1.8.4 **dependency**
   1.8.4.1 dependencyName
   1.8.4.2 dependencyIdentifier
   1.8.4.2.1 dependencyIdentifierType
   1.8.4.2.2 dependencyIdentifierValue

1.8.5 **software**
   1.8.5.1 swName
   1.8.5.2 swVersion
   1.8.5.3 swType
   1.8.5.4 swOtherInformation
   1.8.5.5 swDependency

1.8.6 **hardware**
   1.8.6.1 hwName
   1.8.6.2 hwType
   1.8.6.3 hwOtherInformation

1.8.7 environmentExtension
**Environment Example: PDF File**

**Environment Characteristic** = known to work
**Environment Purpose** = render

**Hardware**
- **Hardware Name** = Intel Pentium II
- **Hardware Type** = processor

**Dependency**
- **Dependency Name** = Mathematica 5.2
- **True Type Math Fonts**

**Software**
- **Software Name** = Adobe Acrobat Reader
- **Software Version** = 6.1
- **Software Type** = renderer

- **Software Name** = Windows NT
- **Software Version** = 5.0
- **Software Type** = operating system
Gap Analysis – Environment Subordinate to Object

- **Solution too specific**
  Too complex to handle in an Object repository.

- **Solution too redundant**
  Rarely specific to a single Object.
  Redundancy results in
  - Unnecessary verbosity;
  - Cumbersome management of Environment descriptions as they evolve.

- **Unable to describe stand-alone Environments**
  Independent of Objects -> Registry

- Solution: Environment as first class entities
Gap Analysis – Scope

- Primarily applicable to computing environments (technical level).
- No representation information in the broader sense.
- No explicit possibility to document the nature of dependencies (e.g. operating systems to hardware).
- No links to registry descriptions other than file formats.
- Specification of versions for software, but not for hardware.
Gap Analysis - Relationships

- **Not generic enough: Environments**
  - Can be related to Objects
  - Can be Objects that need to be preserved
  - Can be software Agents in an Events object
The PREMIS Data Model

Slide adapted from S. Peyrard
Gap Analysis - Relationships

- **Not generic enough: Environments**
  - Can be related to Objects
  - Can be Objects that need to be preserved
  - Can be software Agents in an Events object

- **Environments may link to other Environments**
  - E.g. software application linking to its hardware platform
The PREMIS Data Model

Slide adapted from S. Peyrard
Gap Analysis - Relationships

- **Not generic enough: Environments**
  - Can be related to Objects
  - Can be Objects that need to be preserved
  - Can be software Agents in an Events object

- **Environments may link to other Environments**
  - E.g. software application linking to its hardware platform

- **Environments may link to Events**
  - Creation, adding memory, ...
  - Environments may need to be versioned
The PREMIS Data Model

Slide adapted from S. Peyrard
Requirements

- Environment Type and SubType
- Environment Identifier
- Environment Designation
  (Name and Version)
- Environment Registry
- Environment Storage
- Relationship to other Environment:
  structural, replacement, dependency,
  generalization, reference, ...
- Link to Object, Agent and Identifier
PREMIS Environments in action

3 use cases
Use case 1: Web archives rendering environment

- The national Library of France uses an environment to render its Web archives
- This environment will need periodical updates
- E.g. the browser used (Firefox 2) does not manage ePubs
  - need to update the environment so that it can render the ePub directly with an EPUBReader plugin
Inclusion links
Dependency links

All the red arrows mean:
relationship
type: dependency
subtype: requires
Why PREMIS 3

- The environment description can be **modularized** and **shared** across different objects.
- Express relationships between environments:
  - **whole/part**: environments can bundle together different pieces of environments.
  - **replacement**: environments can be superseded by more recent ones.
  - **dependency**: environments can be related to other environment that support their use.
- Possibility to associate a **registryKey** with an environment.
Use Case 2: Documenting Environment Used by a Format Transformation Service

- Upon ingesting, Quicktime files with various video and audio encodings are normalized into Quicktime files with MJPEG video with lpcm audio.
- The software used for the Quicktime format normalization, i.e., libquicktime, has dependency on other software libraries and codec plugin.
- Need to record the environment for the format normalization tool so that it can be reproduced on a different server or institution.
Software application
libquicktime 1.1.5
depends_lib: libiconv, libpng, zlib, ffmpeg, faac, faad2, libvorbis, libogg, lame

Event
Normalization 2005-08-01

Agent
software: Daitss Transformation Service
role: performer

Software application
libquicktime 1.1.5
depends_lib: libiconv, libpng, zlib, ffmpeg, faac, faad2, libvorbis, libogg, lame

Object bitstream
video/dv50

Object bitstream
audio/mp3

Object bitstream
video/mjepg

Object bitstream
audio/lpcm

Software library
ffmpeg

Software plugin
faac

Software library
libpng

Software plugin
lame

Software library
libvorbis

Software plugin
libogg
Why PREMIS 3

- In PREMIS 2, software used by Agent can only be described in AgentNote with broad granularity.
- Decouple PREMIS Agent from Environment related information.
- Environment can be versioned and maintained separately from Agent.
- Allow different granularity for recording environment information.
Use Case 3: Environment Used for emulation preservation action

- Example from DNB for EC KEEP project. Digital Object is radar simulation for racing boat training package (1999). Vague systems requirements in catalogue metadata: **PC (hardware) and MSDOS (operating system)**
- For emulation there are two issues: we need **versions**, and they need to be **compatible**. These can be found in **TOTEM** [http://www.keep-totem.co.uk/](http://www.keep-totem.co.uk/) . Several iterations?
- **Hardware emulators** and if necessary, **software emulators** can then be specified.
- We need an **emulation platform**, e.g. KEEP Emulation framework (EF) [http://emuframework.sourceforge.net/](http://emuframework.sourceforge.net/) to run these emulators.
- Finally need **reading room environment** to run this EF.
**Trustworthy Online Technical Environment Metadata (TOTEM) compatibility search**

<table>
<thead>
<tr>
<th>OS Name:</th>
<th>Windows NT</th>
</tr>
</thead>
<tbody>
<tr>
<td>OS Version Name:</td>
<td>Windows 2000</td>
</tr>
<tr>
<td>Release Date:</td>
<td>2000-02-17</td>
</tr>
<tr>
<td>Description:</td>
<td>Four editions of Windows 2000 were released, listed here in increasing ranking: Professional, Sever, Advanced Server, and Datacenter Seener.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>OS Name:</th>
<th>Windows NT</th>
</tr>
</thead>
<tbody>
<tr>
<td>OS Version Name:</td>
<td>Windows 2000</td>
</tr>
<tr>
<td>Release Date:</td>
<td>2000-02-17</td>
</tr>
<tr>
<td>Description:</td>
<td>Four editions of Windows 2000 were released, listed here in increasing ranking: Professional, Sever, Advanced Server, and Datacenter Seener.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>OS Name:</th>
<th>Windows NT</th>
</tr>
</thead>
<tbody>
<tr>
<td>OS Version Name:</td>
<td>Windows XP 32-bit Edition</td>
</tr>
<tr>
<td>Release Date:</td>
<td>2001-10-25</td>
</tr>
<tr>
<td>Description:</td>
<td>The first Microsoft operating system for 32-bit processors designed for working with large amounts of memory and projects such as movie special effects, 3D animations, engineering, and scientific programs.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>OS Name:</th>
<th>Windows NT</th>
</tr>
</thead>
<tbody>
<tr>
<td>OS Version Name:</td>
<td>Windows XP 64-bit Edition</td>
</tr>
<tr>
<td>Release Date:</td>
<td>2001-10-25</td>
</tr>
<tr>
<td>Description:</td>
<td>The first Microsoft operating system for 64-bit processors designed for working with large amounts of memory and projects such as movie special effects, 3D animations, engineering, and scientific programs.</td>
</tr>
</tbody>
</table>
relationship type: conceptual
subType: is
specialized in

object file
.object file .img

urn:x-nbn:de:o1

environment DNB catalogue

relationship type: structural
subType: includes

MS-DOS operating system

role: render

MS-DOS 7.1 operating system

relationship type: conceptual
subType: is specialized in

PC hardware

urn:x-nbn:de:y3

DOSBox software emulator

urn:x-nbn:de:y6

DOSBox software emulator

relationship type: emulation
subType: is emulated by

QEMU 1.2 hardware emulator

urn:x-nbn:de:y7

QEMU 1.2 hardware emulator

relationship type: dependency
subType: requires

Emulation Framework 2.0.0

urn:x-nbn:de:y8

Emulation Framework 2.0.0

relationship type: dependency
subType: requires

DNB reading room environment

urn:x-nbn:de:y9

DNB reading room environment
Link to Technical Registry TOTEM

{version of original hardware platform as located in environment registry that is compatible with the software version chosen above}

environmentIdentifier (M, R)
  environmentIdentifierType (M, NR): URN
  environmentIdentifierValue (M, NR): urn:x-nbn:de:y3 (not real identifier)

environmentDescription (O, R)
  environmentName (O, NR): PC
  environmentVersion (O, NR): IBM x86 {chosen as it was current in 1996 and is compatible with MSDOS 7.1}

environmentRegistry (O, R)
  environmentRegistryName (M, NR): TOTEM {hardware is not in PRONOM}
  environmentRegistryKey (M, NR): TUID-xxxx
  environmentRegistryRole (O, NR): external

Emulation Complexity

- Version details are vital for SW, OS, HW etc.
- There are complex interdependencies between SW/OS/HW
- We need iterative technical registry calls to determine these.
- We have stacked environments.
- Finding emulation information is not straightforward, so it is imperative we keep emulator details.
- Running Emulation Frameworks is new for memory institutions, and is complicated, so important to record reading room environment details where this happens.
Conclusion

- Design Environment stand-alone entity
- Validate on community-provided use cases: TOTEM technical registry, IIPC, DAITSS, TIMBUS project, New York University
- Propose to PREMIS Editorial Committee
Thank you!