Appendix A: Container format for CHO object

Introduction

Why a container format?

Cultural heritage data objects (CHDO) come in many different types. In that respect, they are no different from objects of any other domains like multimedia, e-Learning, libraries a.s.o. This means that a CHDO can contain a wide range of aspects or data, ranging from 3D polygon models over MPEG video to CIDOC-CRM data.

It is obviously clear that a single appearance and structure for a CHO would be welcome, irrespective of what data is associated with the object. That makes exchanging, storing and retrieving a lot more easy.

One approach would be to select one aspect of a CHDO which is always available, and attach or link all other data to it.

Disadvantages:

- There is however no aspect that is aways available. In other words, we cannot select e.g. a 3D polygon file format as the default appearance of a CHDO, because there are CHDO with no associated 3D polygons. And creating 'empty' data sets for this very purpose looks rather unpractical and clumsy, as it would mean that, in this case we would need a 3D viewer to access objects with no 3D data associated. This is not good for the accessibility of the data.

- Also, it means that we would have to extend an adapt any existing standard in order to provide hooks or links to other data. This is at its best mis-using the standard and at its worst creating a proprietary format.

- It also means that this select aspect becomes so important, that we cannot modify it without major impact.

Advantages:

- No new file formats are introduced.

The alternative approach is to select a separate top-level object, which acts like a container to all other aspects of a CHDO. Obviously, an XML solution looks the way to go.

Disadvantages:

- A totally new top-level structure has to be introduced. All viewers of a CHDO have to be able to understand it.

Advantages:

- There is a great flexibility to add additional aspects to a CHDO. All existing objects are not affected by this change.
- The impact of a change or update in the standard used for one of the aspects of a CHDO is limited to just that aspect.
- Linking between aspects of CHDOs can now be done easily, e.g. using a specialized link aspect
- Provisions can be made for inclusion of IPR issues to each of the aspects.

Requirements of a container format?

Obviously, the design and choice of a container format is something that needs careful planning and studying. The development of a formal requirements document might be needed, but could take some time.

Some initial requirements can already be written down:

General needs

- An existing standard should be preferred whenever possible. If that is not possible, extensions to existing standards should be preferred over inventing something completely new.
- Care should be taken to choose the simplest solution that serves the requirements.
- Flexibility for accommodating future changes, because of unforeseen future requirements should be possible.
- The standard should be believed to be stable and in use for a sufficiently long time.
- Intellectual property rights (IPR) issues should be incorporated. Not only for the whole CHO, but also for each of its data items separately.

Technical needs

- Preferable XML based. As it is W3C recommended and in general use, XML looks like the preferred way.
  As for storage efficiency reasons, binary encoding of XML might be an option too. See

http://www.w3.org/XML/Binary for details.

- Links to other objects and links within objects

- Registration of unique identifiers to items or links should be possible. Think of e.g. DOI (see http://www.doi.org). There are alternatives though.

- Write-Only objects

  CHDOs should probably be write-only. As they might be referenced from external sources, any change in them might invalidate these links. The only way to prevent this (apart from unpractical back-links), is not to allow changes to objects at all.

Open issues:

- What is the granularity needed?

  Two extremes cases can describe the problem:

  One extreme would be, to have one single object for e.g. a whole museum. This CHO would possibly be registered. All items in the museum would be sub-items (or sub-sub-items etc.) of this single object.

  Advantage: Only one thing to register; all other links are internal.
  Disadvantage: Any change to any object invalidates the whole collection (write-only objects!)

  The other extreme would be to register an object for each physical item in the museum.

  Advantage: Changes have a minimal impact.
  External searching and linking becomes easier
  Disadvantage: Everything has to be registered, which is cumbersome (and expensive).

  Obviously, the optimal solution will be somewhere in the middle.

- Will we allow multiple standards for the same data item?

  A container item allows to have e.g. both JPEG and JPEG2000 as acceptable still image formats. This can be handy to allow a transition from an old standard to a new one. However it can also needlessly complicate the object reader.

Possible solution with existing standards

There is quite some literature available on this or related problems. Particularly in the world of

digital libraries, people have been confronted with the 'container problem' long ago.

Note: what we need here is sometimes also called 'metadata serialization'.

Solution 1: DIDL from MPEG21

Introduction

MPEG-21 is a standard from the Moving Pictures Experts Group, which is part of ISO/IEC. This group brought us the very successful and widely used standards for video MPEG-1, MPEG-2 and multimedia MPEG-4, and the less successful standards for multimedia content description MPEG-7. And of course MPEG-21, which standardizes a 'multimedia framework'.

Some relevant parts of MPEG-21 are.

Part 2: Digital Item Declaration Language (DIDL)
Part 3: Digital Item Identification Language (DIIL)
Part 4: Intellectual Property Management and Protection (IPMP)
Part 5: Rights Expression Language (REL)
Part 7: Digital Item Adaptation (DIA)

Of interest to this community, as far as is clear for the moment, is mainly Part 2, DIDL. If IPR issues would become more important, Part 4 and Part 5 might be interesting too.

Technically

In DIDL, the following terms are defined. The definitions in this text are NOT the normative definitions as in the standard, but simplified explanations.

1. **Resource:** An 'identifiable asset', e.g. a picture, X3D file, or even a physical object.

2. **Component:** The binding of a **Resource** to a **Descriptor**. Like e.g. bit rates, starting points, character sets.

3. **Descriptor**: Associates information (e.g. a **Component**) with the enclosing element.

4. **Statement**: A literal textual value; like descriptive or identifying information.

5. **Fragment**: Describes a specific point or range in a **Resource**.

6. **Anchor**: Binds **Descriptors** to a **Fragment**.

7. **Predicate**: A 'binary' declaration ('true','false','undecided').

8. **Selection**: Describes a decision, that will affect a **Condition** in an **Item**. Its associated

Predicate becomes true, when the selection is taken.

9. **Condition**: Describes the enclosing element as optional, and links it to relevant **Selection**s.

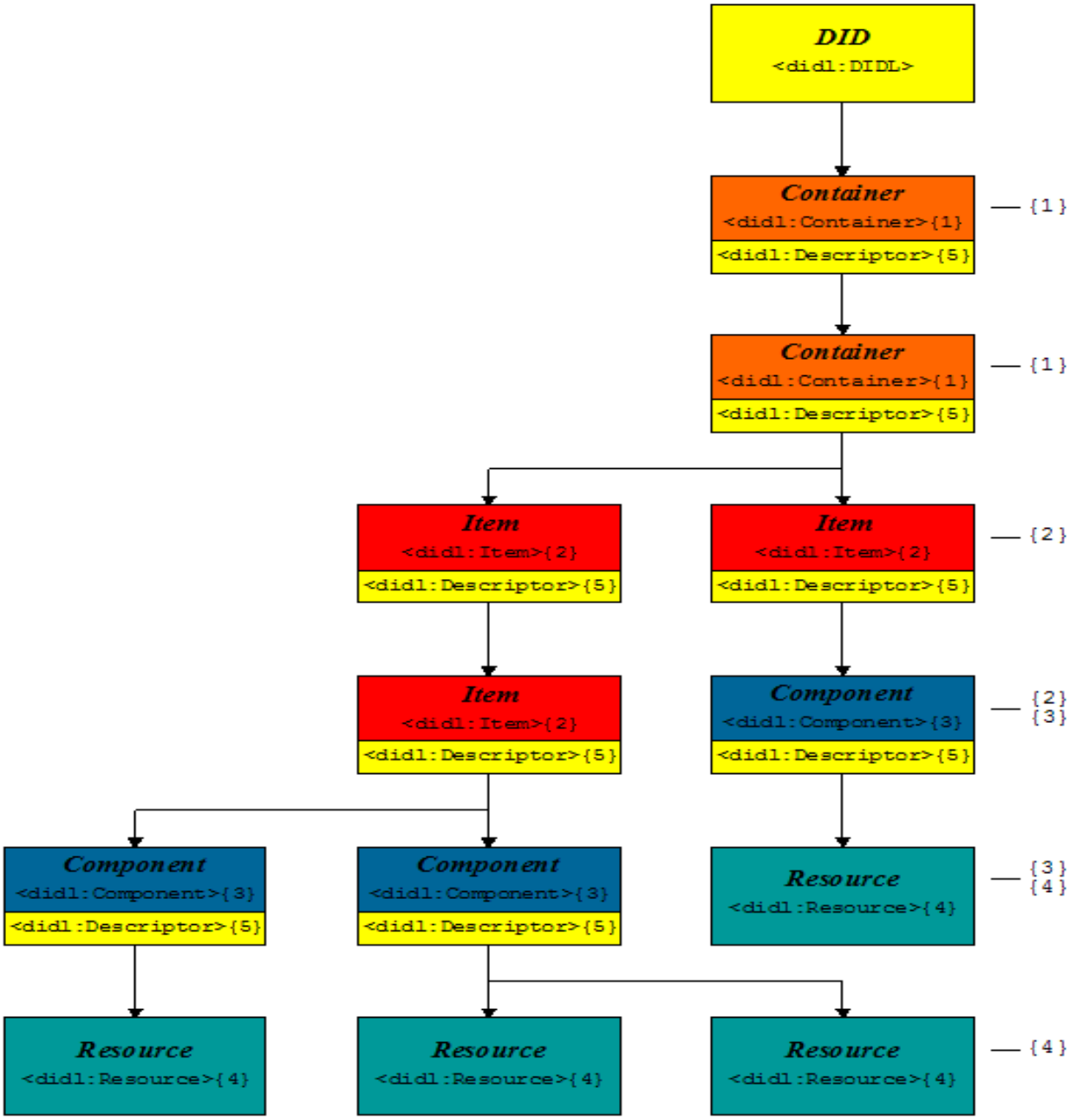
10. **Choice**: Set of related **Selection**s.

11. **Item**: Grouping of **Item**s or **Component**s, bound to **Descriptor**s.

12. **Container**: Structure that can group **Item**s.

13. **Assertion**: A state of a **Choice**.

An example of a hierarchy might be more explanatory. As you can see, a **Container** {1} can contain other **Container**s, or **Item**s {2}. **Item**s contain **Components** {3}, which describe **Resources** {4}. This picture is a simple structure without any choices or conditions etc.



DIDL is of course implemented in XML.

Users:

As is obvious, the terminology is 'tuned' for multimedia objects. Nevertheless, the possibilities of the framework are much wider.

In digital libraries, a famous user is the LANL (Los Alamos National Laboratory) Digital Library (http://library/lanl/gov). A paper describes exactly this implementation: 'Using MPEG-21 DIDL to Represent Complex Digital Objects in the Los Alamos National Laboratory Digital Library', J. Bekaert, P. Hochstenbach and H. Van de Sompel in D-Lib Magazine, November 2003, Volume 9 Number 11.

There is a DIDL plug-in for the D-Space library software environment.

Solution 2: XPackage and RDF

Introduction

RDF stands for Resource Description Framework; Initially, it was a language for representing information about resources on the web; generalized any digital object will do. Details see http://www.w3c.org/RDF.

XPackage is obviously XML package format. It can describe various resources and their associations. It is based on XML, RDF and XLink. Strictly spoken, XPackage is an RDF ontology. See http://www.xpackage.org.

RDF and Xpackage grew inside W3C (http://www.w3c.org).

Web Ontologies like DAML+OIL and OWL are based on RDF and RDF Schema.
These are part of the work going on on the Semantic Web.

Technically

It's an XML Schema, based on the use of URI (Uniform Resource Identifiers). It implements a grammar of 'subject, predicate, object' sentences. Several types of containers are available ('bags', 'sequences' and 'alternatives'). A specific vocabulary can be set up, using the 'Vocabulary Description Language', called RDF Schema (RDFS).

The following piece of XML gives a taste of how it works. It's an example of a Magazine, described using Dublin Core in RDF. The code should be clear enough to explain itself.

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:dcterms="http://purl.org/dc/terms/">
    <rdf:Description
rdf:about="http://www.dlib.org/dlib/may98/05contents.html">
      <dc:title>DLIB Magazine - The Magazine for Digital Library
Research
        - May 1998</dc:title>
      <dc:description>D-LIB magazine is a monthly compilation of
       contributed stories, commentary, and
briefings.</dc:description>
      <dc:contributor>Amy Friedlander</dc:contributor>
      <dc:publisher>Corporation for National Research
Initiatives</dc:publisher>
      <dc:date>1998-01-05</dc:date>
      <dc:type>electronic journal</dc:type>
      <dc:subject>
        <rdf:Bag>
          <rdf:li>library use studies</rdf:li>
          <rdf:li>magazines and newspapers</rdf:li>
        </rdf:Bag>
```

```
      </dc:subject>
      <dc:format>text/html</dc:format>
      <dc:identifier rdf:resource="urn:issn:1082-9873"/>
      <dcterms:isPartOf rdf:resource="http://www.dlib.org"/>
    </rdf:Description>
 </rdf:RDF>
```

XPackage contains items like the following, which make the packaging more standardized, than when one would define his own RDF description for this.

`<xpackage:resource>`
(resource) A generic resource.
`<xpackage:package>`
(resource) A generic package (such as an archive) that might contain other resources.

`<xpackage:manifest>`
(property) A list of other resources one resources contains.
`<xpackage:organization>`
(property) The order of other resources as they appear inside one resource.
`<xpackage:contentType>`
(property) The MIME media type of a resource.
`<xpackage:location>`
(property) The physical location of a resource, in a file system for example.

XOP, (XML Optimized Packaging) is a means of more efficiently serializing XML Infosets that have certain types of content. It is comparable, but more recent, than XMLTape.
XOP is compatible with Xpackage and RDF.

Users

RDF is very widely used on the web. RSS is based on RDF. Xpackage is supposedly the competitor of DIDL, part of MPEG21.

The major difference is that RDF is less specific, i.e. less tunes to a particular application, but therefore also more flexible for non-standard needs.

Solution 3: METS

Introduction

Metadata Encoding and Transmission Standard (METS) is a framework to structure relevant metadata (descriptive, administrative and structural) in the digital library world. It's an XML Schema. METS is more or less the multimedia version of the old an well known classical library standard MARC. The 'Digital Library Federation' started this effort. METS is in actively being developed further.

Technically

In essence, a METS document, contains the following parts:

1. Header: With information as creator, editor etc.

2. Descriptive metadata: Can be either internal (in this file) or external (in a file pointed to). Contains things like a URN or URL or DOI, MARC data, Dublin Core data etc.

3. Administrative metadata: Four major types of information: 1) technical metadata on the file in the container 2) IPR info 3) source metadata (e.g. an analog original of some image) 4) digital provenance metadata (interrelations between original and derived data). This can again be internal or external.

4. File section: The data itself, either as a reference (e.g. a URL) or in-line or embedded (in that case base64 encoded).

5. Structural map: A hierarchical structure of the data provided.

6. Structural links: Links between items in the structural map. This is based on the Xlink XML syntax.

7. Behavior section: Can be used to describe e.g. how to present the item to a viewer.

There are software tools (some of which are free) available to work with METS items.

Sample documents can be found at http://www.loc.gov/standards/mets/mets-examples.html.

Users

METS seems to widely used in digital libraries and museums. DSpace has a METS profile.

The Library of Congress (http://www.loc.gov/index.html) uses METS. So does the Oxford Digital Library (http://www.odl.ox.ac.uk).

Conclusion

Without further investigation, it's unwise to decide which of these three options to choose for the CHDO. Nevertheless, there might be a slight advantage to the METS approach, for the following reasons:

- METS and MPEG21 are targeted to specific domains. The METS domain (digital libraries) probably lies closer to cultural heritage than MPEG21's multimedia domain. So of these two, METS might be preferable (unless other needs surface).
- RDF is not targeted to any specific domain, and is therefore much more flexible. But that

also means that we would have to develop the needed infrastructure ourselves. This would cost time. We would also miss the leverage of building further on a more concrete existing standard.

But once again, these are preliminary conclusions.

Glossary

XMLTape	A binary container of XML items (in XML); works with byte-offsets to increase speed. Internal data is zipped.
Dublin Core	http://de.wikipedia.org/wiki/Dublin_Core http://dublincore.org Dublin Core (DC), also called Dublin Core Metadata Initiative (DCMI) A set of 15 keywords (core elements) for metadata and some extras (element refinements) (Identifier, Format, Type, Language, Title, Subject, Coverage, Description, Creator, Publisher, Contributor,Rights Holder, Rights, Source, Provenance, Source, Relation, Audience, Instructional Method, Data) Normally implemented in XML; widely in use, e.g. In each Linux machine by the 'scroll keeper' documentation system.
OMF	http://www.ibiblio.org/osrt/omf/ Open Source Metadata Framework (OMF) The OMF aims to collect data about Open Source documentation, or metadata, that will be used to describe the documentation. Uses Dublin Core.
SCORM	http://de.wikipedia.org/wiki/SCORM Sharable Content Object Reference Model A collection of standards of e-learning. Currently SCORM2004 Contains: AICC (Aviation Industry), DCMI (Dublin Core), IEEE, IMS (Instructional Management System) and Ariadne(?)
METS	http://www.loc.gov/standards/mets Metadata Encoding & Transmission Standard XML Schema for encoding descriptive, administrative and structural metadata for objects in a digital library.
OAI-PMH	http://en.wikipedia.org/wiki/OAI-PMH

	Open Archives Initiative Protocol for Metadata Harvesting
	Protocol from OAI; XML over HTTP; currently version 2.0 Adopted by Herbert Van de Sompel (Gent), for LANL (Los Alamos National Laboratory). Can be registered (not obligatory). Used in commercial search engines, and by Google.
OAI	http://en.wikipedia.org/wiki/Open_Archives_Initiative
	Framework for digital libraries offering 'value added services'. Requires mapping to at least Dublin Core, but better MARC
MARC	http://en.wikipedia.org/wiki/Machine_Readable_Cataloging
	MAchine-Readable Cataloging. XML Schema for bibliographic data (originated from 'Library of Congress'.) Very old, but widely used.
AACR2	http://en.wikipedia.org/wiki/AACR2 http://www.aacr2.org/
	Anglo-American Cataloguing Rules, Second Edition
	Set of 'standards' used by the American & Canadian Library Association. (very very old, 1967)
OWL	http://en.wikipedia.org/wiki/Web_Ontology_Language
	Web Ontology Language.
CHIN Data Dictionaries	http://www.chin.gc.ca/English/Collections_Management/index.html
	The CHIN Data Dictionaries (Humanities, Natural Sciences, and Archaeological Sites) contain a description of units of information for museum collection and archaeological site documentation and management. Each data field in the CHIN Data Dictionaries is described by a field label, a mnemonic, a name, a definition, entry rules, related fields, a data type, examples, a discipline, and a source.
	Used in Canada
SPECTRUM XML DTD	http://www.mda.org.uk/stand.htm
	An XML DTD created by mda (UK);The SPECTRUM XML DTD is intended to be a universal interchange format for museum collections management systems that are based on SPECTRUM or that can map to SPECTRUM. The SPECTRUM XML DTD "will allow different collections management systems to exchange complete museum records that are compliant with the SPECTRUM standard. It will also provide a means for testing
XPackage	http://www.xpackage.org/specification/
	An XML package format, for describing resources and their associations. Based on XML, RDF and XLink. Can package various content into a single searchable XML

file.