Archive Ingest and Handling Test: ODU’s Perspective

Michael L. Nelson
Department of Computer Science
Old Dominion University
http://www.cs.odu.edu/~mln/

DLF Fall Forum 2005, Charlottesville VA, November 7 2005
Fortress Model

Five Easy Steps for Preservation:

1. Get a lot of $
2. Buy a lot of disks, machines, tapes, etc.
3. Hire an army of staff
4. Load a small amount of data
5. “Look upon my archive ye Mighty, and despair!”
ODU’s Research Goals

• We’re in the CS department, not the library
  – Less infrastructure (bad)
  – More freedom (good)

• Interested in repository/object interaction
  – Long-range vision: repositories fade away; objects are responsible for their own preservation
  – Could we accomplish this with our “bucket” technology?
    • Significant questions about archive granularity
    • Transition to MPEG-21 Digital Item Declaration Language (DIDL) based buckets

• New models for digital preservation?
Buckets

- Buckets: self-contained, web-accessible objects
  - Grew out of research for serving NASA documents, esp. NACA Reports
    - [http://naca.larc.nasa.gov/](http://naca.larc.nasa.gov/)
    - [http://doi.acm.org/10.1145/374308.374342](http://doi.acm.org/10.1145/374308.374342)
  - implicit assumptions:
    - 1 bucket = 1 logical item (N physical items)
    - Display is for human use
    - Bucket contents are DOM-parsable
Which Interface?

Display based on web use

Display based on archival use
Bucket / MPEG-21 Model

http://beatitude.cs.odu.edu:8080/bucket/

Bucket Infrastructure
- methods
- logs
- support libraries

MPEG-21 DIDL Payload
MPEG-21 DIDL

- A generic, powerful complex object metadata format
  - Based on an abstract data model
  - Semantics separated from syntax
    - i.e. the tags don’t mean anything -- a little disconcerting at first glance
  - Digital library use championed by LANL
    - http://www.dlib.org/dlib/november03/bekaert/11bekaert.html
    - http://www.dlib.org/dlib/february04/bekaert/02bekaert.html
MPEG-21 DIDL Data Model

How to encode Archive?
• 1 file = 1 DID
• 1 archive = 1 container
• 1 archive = 1 component
• 1 file = 1 component

descriptors are used to convey:
• digital item identification (DII)
• digital item processing (DIP)
• rights expression language (REL)
• digital item relations (DIR)
• creation date (DIDT)

all resources within a component are equivalent by definition
1 File = 1 Component

8 file archive for demo purposes…
http://www.cs.odu.edu/~mln/aiht/
Looking Inside the Archive

<?xml version="1.0" encoding="UTF-8" ?>
<archive.xml>
  <Container>
    <Original Archive Identifier />
    <Descriptor>
      <Creation Date of this XML representation />
    </Descriptor>
    <Archive />
  </Container>
  <Item>
    <Original Archive />
    <Item>
      <Item File Name Mapping />
      <Archive Contents />
      <Component />
      <Component />
      <Component />
      <Component />
      <Component />
      <Component />
      <Component />
      <Component />
      <Component />
      <Component />
      <Item />
    </Item>
  </Item>
</archive.xml>
Looking at a Single File…

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<didl:DIDL xmlns:didl="urn:mpeg:mpg21:2002:01-DIDL-NS">
  <didl:Container>
    <didl:OriginalArchiveIdentifier />
    <didl:Descriptor>
      <didl:CreationDate of this XML representation />
    </didl:Descriptor>
    <didl:Archive />
  </didl:Item>
  <didl:Item>
    <didl:OriginalArchive />
  </didl:Item>
  <didl:Item>
    <didl:FileNameMapping />
  </didl:Item>
  <didl:Item>
    <didl:ArchiveContents />
  </didl:Item>
  <didl:Item>
    <didl:Component>
      <didl:FileIdentifier />
    </didl:Component>
    <didl:Descriptor>
      <didl:MD5Checksum of the File Contents />
    </didl:Descriptor>
    <didl:Descriptor>
      <didl:Output of Jhove />
    </didl:Descriptor>
    <didl:Descriptor>
      <didl:Output of file />
    </didl:Descriptor>
    <didl:Descriptor>
      <didl:FileResource ref="repository/9abd37197bc62a728a303e5931984332a.0" />
    </didl:Descriptor>
  </didl:Component>
  <didl:Component>
  </didl:Component>
  <didl:Component>
  </didl:Component>
  <didl:Component>
  </didl:Component>
  <didl:Component>
  </didl:Component>
  <didl:Component>
  </didl:Component>
  <didl:Component>
  </didl:Component>
  <didl:Component>
  </didl:Component>
</didl:DIDL>
```
Design Decisions: File Storage

• Store each file as a `<Component>`
  – Big: each file is base64’d into the DIDL
  – Small: each file is ref’d from the DIDL to a directory
    • Filename = MD5 hash of the original file name (not contents!) + a version number
    • Example:

```xml
<didl:Resource mimeType="image/gif" ref="repository/1641ad793a1cc597a18e9dd4dd3c64d5.0" />
```
## Archive Sizes

<table>
<thead>
<tr>
<th>Name</th>
<th>XML File Size (bytes)</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIDL.xml</td>
<td>15382712841 (15 GB)</td>
<td>By-Value. This first upload did not contain the files &quot;outside&quot; of the original tar file.</td>
</tr>
<tr>
<td>DIDL2.xml</td>
<td>35633037513 (35 GB)</td>
<td>By-Value. This upload contained all files (tar file + database files)</td>
</tr>
<tr>
<td>DIDL3.xml</td>
<td>322653621 (322 MB)</td>
<td>By-Reference. All files. File size does not include tar file (26 GB).</td>
</tr>
</tbody>
</table>
Design Decisions: Ingestion

- For every program/process to apply to a file, create a corresponding <Descriptor>
  - Jhove
  - Unix “file”
  - Fred URI
  - MD5 of file contents
- Expandable, scriptable list of metadata extraction / analysis programs
- Ingestion is parallelized over a workstation cluster
Example Output: MD5

<didl:Descriptor>
<didl:Statement mimeType="text/xml; charset=UTF-8">
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://purl.org/dc/elements/1.1/
http://dublincore.org/schemas/xmls/simpledc20021212.xsd">perl/Digest::MD5</dc:creator>
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://purl.org/dc/elements/1.1/
http://dublincore.org/schemas/xmls/simpledc20021212.xsd">52217a1bcd2be7cf05f36066d4cdc9cf</dc:description>
</didl:Statement>
</didl:Descriptor>
Conversion: AVI -> VOB

- Investigated PDF -> SVG, but tools were not mature
- Selected “transcode” for AVI -> VOB conversion
  - http://www.transcoding.org/
- Also implemented ImageMagick based rules for standard graphics conversion

http://beatitude.cs.odu.edu:8080/~gmanepal/Transcode.html
Conversion: Linking Old to New

If the previous version of the Resource was specified as:

```xml
<didl:Resource mimeType="image/jpeg"
ref="repository/9abd37197bc62a72a303e5931984332a.0" />
```
then the new version of the resource is specified as:

```xml
<didl:Resource mimeType="image/png"
ref="repository/9abd37197bc62a72a303e5931984332a.1" />
```
Harvard Ingest

- Harvard’s model was the most similar to our MPEG-21 model
- Ingesting from another archive is (roughly) the same as initial ingest
  - Save any metadata that was delivered in the original METS file as a `<Descriptor>`
    - We don’t trust it, but it might be useful for future forensics
  - Re-ingest in the normal way
- Our export is part of the bucket API:

---

<didl:Descriptor>
<didl:Statement mimeType="text/xml; charset=UTF-8">
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://purl.org/dc/elements/1.1/
http://dublincore.org/schemas/xmls/simpledc20021212.xsd">External Metadata</dc:creator>
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:aes="http://www.aes.org/audioObject"
xmlns:app="http://hul.harvard.edu/ois/xml/ns/drs/app"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:mix="http://www.loc.gov/mix/"
xmlns:tcf="http://www.aes.org/tcf"
xmlns:txt="http://www.loc.gov/METS/text/
xmlns:xlink="http://www.w3.org/TR/xlink"
xmlns:simpledc20021212.xsd">
<file ID="F1" MIMEType="image/jpeg" SEQ="1" SIZE="194914" ADMID="T1"
CHECKSUM="a7969810684c468525313b8282501405" CHECKSUMTYPE="MD5"
OWNERID="aiht/websites/chnm/september11/REPOSITORY/CONTRIBUTORS/1199_photos/wtc_web/wetc5.jpg">
<FLocat LOCTYPE="URL" xlink:type="simple"
xlink:href="file:///aiht/data/2004/12/17/0122.jpg" />
</file>
</didl:Statement>
</didl:Descriptor>
“In Vivo” Preservation

- As part of the ingest process, we looked for copies of the ingested web page in the “living web”
  - Idea: find all replicated / similar pages and maintain pointers to them
  - Problem: We could find related documents, but finding copies was difficult
    - Term Frequency (TF) – easy to compute
    - Inverse Document Frequency (IDF) – difficult to compute
    - Solution: lexical signatures, Phelps & Wilensky:
      - http://www.dlib.org/dlib/july00/wilensky/07wilensky.html
  - Spinoff research:
    - Terry Harrison’s MS thesis
    - Frank McCown’s Ph.D. dissertation
    - Joan Smith’s Ph.D. dissertation
    - NSF proposal on “in vivo” preservation
The DIP is the TMD*

- Using METS or MPEG-21, there is no need for a separate transfer metadata format
- METS & MPEG-21 can be the lumps of XML exchanged between harvesters & repositories
  - http://www.dlib.org/dlib/december04/vandesompel/12vandesompel.html
- Web servers can be made to automatically expose their contents via OAI-PMH
  - http://www.modoai.org/

Figure 1, Bekaert & Van de Sompel
http://www.dlib.org/dlib/june05/bekaert/06bekaert.html

* Eat your heart out, Marshal McLuhan