Expanding the CDL Digital Preservation Repository for New Projects

Fall 2006 Digital Library Federation Forum

Stu Sugarman, Shifra Pride Raffel, David Loy and Mark Reyes, California Digital Library
{stuart.sugarman, shifra.raffel, david.loy, mark.reyes}@ucop.edu
Outline

• Brief look at Web Archiving Service
• Digital Preservation Repository architecture
• Open Content Alliance project: Tracker and Feeder Services
• Web Archiving Service project
• Longer demo of Web Archiving Service
Web Archiving Service

Listed below are the captures you have defined and their current status. Click "Start" to initiate a capture. Use the manage captures screen to edit captures.

<table>
<thead>
<tr>
<th>Name/Description</th>
<th>Status</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yoga Sites</td>
<td>Ready</td>
<td><img src="start.png" alt="Start" /></td>
</tr>
<tr>
<td>WMember2 created this capture</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Test site capture</td>
<td>Ready</td>
<td><img src="start.png" alt="Start" /></td>
</tr>
</tbody>
</table>

Refresh status
Digital Preservation Repository

• A secure service for the 10 UC libraries to store their digital collections

• Went into production July 2005

• About 120,000 objects, about 500 GB

• Growing to about 60 TB quite soon

• Web interface; java toolkit,
  http://www.cdlib.org/inside/projects/preservation/dpr/toolkit/
DPR: Flexible and Extensible

• Service-oriented
• Loosely coupled pieces
• Based on OAIS Model
• J2EE Java
• Configurable – pattern of project structure and deployment
• Jhove, NOID (ARK) -- easy to incorporate
Adding a New Service

- Client layer(s)
  - Base client classes
  - Requester for each task
- Service layer(s)
  - Receiver
  - Handler class for each task
- Management layer(s): controllers, database managers, tool interfaces
- Configuration of properties, including for deployment
Open Content Alliance
Tracker and Feeder Services
Ingest

- Ingest client
- METS
- Ingest
- Repository
Ingest

- User creates METS record
- User submits METS record to Ingest Service using SOAP
- Ingest copies digital data to repository
- Ingest returns SOAP response to user
Feeder

Diagram:
- Web
- Feeder
- Status
- Ingest
- METS
- Repository
Why Feeder

- Need for batch handling of multiple ingests
- Ability to customize “feeding” process to a specific project:
  - OCA
  - WAS
  - OAI
  - Google
- Customized and Automated METS generation based on project
- Need to monitor results for long running loads with multiple ingests
Tracker

• Logging the manual processes involved in the QA and preservation of digital objects
• Event types: publish, validate, qa, ingest, send_scp
• Preserve: eventid, eventtype, eventdate, objectid, userid, altobjectid, collectionlibrary, eventagent, eventstatus, eventnote, batchid
Tracker

- Event types: publish, validate, qa, ingest, send_scp
- Logging the manual processes involved in the QA and preservation of digital objects:
  - Receiving notice that item is available at IA
  - QA of the Digital Object
  - Identification of loading problems
- Logging of automated processes:
  - Feeder validation of data format
  - Feeder submission to ingest
Tracker and Feeder Development

Feeder Submit -> Feeder

Tracker -> Feeder Submit

Tracker
• 3-year NDIIPP project, “The Web at Risk: Preserving our Nation’s Political and Cultural Heritage”, with NYU, UNT and other partners


• Uses feeder, building on OCA work

• Internet Archive’s Heritrix crawler
WAS Indexing and Extraction

ARC

Index

Repository

Search And Display

OpenSearch results & ARC records
WAS Indexing and Extraction

- Index during ingest (NutchWAX)
  - Store Nutch index in AIP
- Extract index on demand (Search)
  - Keyword and URL search
  - Cache index to local disk
- Extract ARCs on demand (Display)
  - Cache ARCs to local disk
Questions?