Institutional Repositories and the OAI-PMH: beyond Dublin Core

Henry Jerez, Jeroen Bekaert, and Herbert Van de Sompel
Los Alamos National Laboratory, Research Library
Digital Library Research & Prototyping Team
Outline

(1) Motivation

(2) OAI-PMH for content

(3) Example 1 : LANL Repository

(4) Example 2 : mod_oai

(5) Example 3 : DSpace plug-in prototype

(6) Federations of IRs and OAI-PMH

(7) Conclusion
Motivation

• Digital Libraries, Institutional Repositories, Archives
  - Growing interest in exposing/harvesting content, not only metadata
    - cf. DARE, DINI, JISC FAIR, DSpace
  - Growing interest from Web search engines to harvest quality content from these repositories.
  - Well-established adoption of the OAI-PMH. Tools available. It makes sense to use OAI-PMH to expose/harvest content.
    - But can content be exposed/harvested through OAI-PMH? See later.

• The Web
  - Web crawling solutions not utterly efficient.
  - No efficient change control mechanism on the Web.
  - OAI-PMH can provide optimizations.
    - But can general Web content be harvested through OAI-PMH? See later.
Outlines

1) Motivation

2) OAI-PMH for content

3) Example 1: LANL Repository

4) Example 2: mod_oai

5) Example 3: DSpace plug-in prototype

6) Federations of IRs and OAI-PMH

7) Conclusion
OAI-PMH

selective harvesting requests:
• datestamp
• set

provides services using harvested metadata
exposes metadata pertaining to resources
OAI-PMH data model

OAI-PMH identifier
= entry point to all records pertaining to the resource

- metadata pertaining to the resource
- XML data pertaining to the resource
- modeled representation of the resource

- Dublin Core metadata
- MPEG-21 DIDL
- METS
- MARCXML metadata

- simple model
- complex model
- complex model
- simple model
OAI-PMH and complex models

- OAI-PMH record == modeled representation of the resource
- Can be selectively harvested via OAI-PMH ~ datestamp, set
- Resource can be:
  - simple object (1 file)
  - compound object (multiple files)
- OAI-PMH records can contain:
  - Typical metadata
  - A variety of secondary information: rights, relationships, format information, …
  - Actual resource(s)
    - By-Value – base64 encoded
    - By-Reference – http address of resource
    - both
  - Identifiers of metadata and resource(s), unambiguously mapped to the identified data
OAI-PMH and complex models: data/id mapping

- Example: a compound object consisting of:
  - metadata
    (id = info:lanl-repo/opac/LANLb10012271)
  - technical report
    - 1 file: pdf
      (id = info:lanl-repo/tr/LA-9870)
    - 1 file: tiff
      (id = info:lanl-repo/tr/LA-9871)
OAI-PMH and complex models: data/id mapping

<table>
<thead>
<tr>
<th>complex model</th>
<th>simple model : DC</th>
</tr>
</thead>
<tbody>
<tr>
<td>ds2 - id: info:lanl-repo/tr/LA-9871</td>
<td>ref: <a href="http://library.lanl.gov/tr/foo.tiff">http://library.lanl.gov/tr/foo.tiff</a></td>
</tr>
</tbody>
</table>

- No distinction between identifiers & locators
- Unclear relation between identifiers & locators
- Where does the identifier of the metadata go?
OAI-PMH & complex models: related papers

- Using the OAI-PMH ... Differently.
  http://www.dlib.org/dlib/july03/young/07young.html

- Using MPEG-21 DIDL to Represent Complex Digital Objects in LANL
  http://www.dlib.org/dlib/november03/bekaert/11bekaert.html

- Using MPEG-21 DIP and NISO OpenURL for the Dynamic Dissemination of Complex Digital Objects in LANL
  http://www.dlib.org/dlib/february04/bekaert/02bekaert.html

- The multi-faceted use of the OAI-PMH in the LANL Repository
Outline

(1) Motivation

(2) OAI-PMH for content

(3) Example 1: LANL Repository

(4) Example 2: mod_oai

(5) Example 3: DSpace plug-in prototype

(6) Federations of IRs and OAI-PMH

(7) Conclusion
Example 1 : LANL Repository

- Local storage of Terrabytes of scholarly assets
- Upon ingestion, assets are turned into MPEG-21 DIDL documents that contain:
  - Metadata pertaining to assets
  - Assets and/or pointers to assets
  - Identifiers of metadata, assets, DIDL documents
  - A variety of secondary information
- Stored MPEG-21 DIDL documents made accessible to – multiple – downstream applications via the OAI-PMH
- OAI-PMH as a Repository Access Protocol to access metadata and content.
Outline

(1) Motivation

(2) OAI-PMH for content

(3) Example 1 : LANL Repository

(4) Example 2 : mod_oai

(5) Example 3 : DSpace plug-in prototype

(6) Federations of IRs and OAI-PMH

(7) Conclusion
Example 2: Old Dominion University & LANL mod_oai project

- Funded by Andrew W. Mellon Foundation
- Implement OAI-PMH plug-in for – Apache - Web servers
- Will allow selective & incremental OAI-PMH harvesting of content hosted by Web servers
  - OAI-PMH identifiers == URLs
  - datestamp
  - sets ~ MIME type
  - initially static Web content
- Two operating modes for crawlers:
  - General crawler: ListIdentifiers => URLs of Web content
  - Advanced crawler: ListRecords ~ Dublin Core and one or more complex object formats
- OAI-PMH as a tool to make harvesting of Web content more efficient
Outline

(1) Motivation

(2) OAI-PMH for content

(3) Example 1 : LANL Repository

(4) Example 2 : mod_oai

(5) Example 3 : DSpace plug-in prototype

(6) Federations of IRs and OAI-PMH

(7) Conclusion
Example 3: LANL DSpace plug-in prototype

- Introduced at recent DSpace Federation meeting
- Maps DSpace data model
  
  [item – bundle – component]

  to MPEG-21 DIDL data model

  [Container – Item – Resource]

- Exposes MPEG-21 DIDL documents through built-in DSpace OAI-PMH infrastructure

- Metadata (Dublin Core) and Content (MPEG-21 DIDL) harvestable via the OAI-PMH
MPEG-21 DIDL : Data Model

- Abstract Definitions + W3C XML Schema
- Entities
  - a Container didl:Container
  - an Item didl:Item
  - a Component didl:Component
  - a Resource didl:Resource
  - a Descriptor didl:Descriptor
  - ...
- Remark
  - a DIDL compliant document == a DID
MPEG-21 DIDL: Data Model

Institutional Repositories and the OAI-PMH: beyond Dublin Core
Research Library, Los Alamos National Laboratory
April 19, 2004 – DLF Developers Meeting, New Orleans, LA
MPEG-21 DIDL : Descriptors

• Secondary information pertaining to Entities
  o MPEG-21 defined uses
    - identification information – MPEG-21 Part 3 : DII
    - rights information – MPEG-21 Part 5 : REL / Part 4 : IPMP
    - processing information – MPEG-21 Part 10 : DIP
  o community/application specific uses
    - e.g.: LANL use, DSpace use, …
DSpace DID: general structure

- **Container**
  - <didl:Container>
  - <didl:Descriptor>

- **Item**
  - <didl:Item>
  - <didl:Descriptor>

- **Component**
  - <didl:Component>
  - <didl:Descriptor>

- **Resource**
  - <didl:Resource>

**Bundle**

- **Item**
  - <didl:Item>
  - <didl:Descriptor>

- **Component**
  - <didl:Component>
  - <didl:Descriptor>

- **Resource**
  - <didl:Resource>

**Bitstream**

- **Component**
  - <didl:Component>
  - <didl:Descriptor>

- **Resource**
  - <didl:Resource>
DSpace DID: mapping descriptive metadata & content

Institutional Repositories and the OAI-PMH: beyond Dublin Core
Research Library, Los Alamos National Laboratory
April 19, 2004 – DLF Developers Meeting, New Orleans, LA
DSpace DID Descriptors: identifier

Institutional Repositories and the OAI-PMH: beyond Dublin Core
Research Library, Los Alamos National Laboratory
April 19, 2004 – DLF Developers Meeting, New Orleans, LA
DSpace DID Descriptors: RDF relationships

Institutional Repositories and the OAI-PMH: beyond Dublin Core
Research Library, Los Alamos National Laboratory
April 19, 2004 – DLF Developers Meeting, New Orleans, LA
DSpace DID Descriptors: RDF relationships

- urn:hdl:1751.repo/15
  - dcterms:hasPart #d2e82b56-6091-4f20-9cac-e4b7c54d40da
  - dcterms:hasPart #62ec8366-9a1d-45cd-a167-dabf102988a0
  - dcterms:hasPart #d2e82b56-6091-4f20-9cac-e4b7c54d40da
    - rdf:type http://library.lanl.gov/2003-10/STB-RL/DIR/VOC/content
DSpace to DID: mapping overview

<table>
<thead>
<tr>
<th>Container</th>
<th>MPEG-21 DIDL</th>
<th>DSpace</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>dii (MPEG-21)</td>
<td>Identifier</td>
</tr>
<tr>
<td></td>
<td>diadm</td>
<td>di/terms:created</td>
</tr>
<tr>
<td></td>
<td>dir</td>
<td>rdf</td>
</tr>
<tr>
<td></td>
<td>dipr</td>
<td>created</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Item</th>
<th>MPEG-21 DIDL</th>
<th>DSpace</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>dii (MPEG-21)</td>
<td>Identifier</td>
</tr>
<tr>
<td></td>
<td>diadm</td>
<td>dcterms:created</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Component</th>
<th>MPEG-21 DIDL</th>
<th>DSpace</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>dii (MPEG-21)</td>
<td>Identifier</td>
</tr>
<tr>
<td></td>
<td>diadm</td>
<td>dcterms:created</td>
</tr>
<tr>
<td>didl (MPEG-21)</td>
<td>@mimetype</td>
<td>mimetype</td>
</tr>
<tr>
<td></td>
<td>diadm</td>
<td>digestValue</td>
</tr>
<tr>
<td></td>
<td>diadm</td>
<td>digestMethod</td>
</tr>
<tr>
<td></td>
<td></td>
<td>checksum</td>
</tr>
<tr>
<td></td>
<td></td>
<td>checksum_algorithm</td>
</tr>
</tbody>
</table>
DSpace to DID – mapping considerations

- DSpace:
  - Lack of identifiers at Bundle and Bitstream level
  - Unknown mimeType
  - Unequal treatment of descriptive metadata and content. cf. MD5 digest.
  - Unclear use of rights and licenses

- DIDL:
  - Digest ~ W3C XML Signature
  - Community defined Namespaces for Descriptors required. For example: RDF.
LANL DSpace plug-in : DIDs via OAI-PMH

• DSpace DIDs contain:
  o identifiers
  o descriptive metadata
  o content
  o secondary information
• Harvestable through OAI-PMH
  o OCLC OAICat
    - Crosswalks
    - OAIDCCrosswalk.java
  o Components of LANL DSpace Plugin:
    - crosswalk: DIDLCrosswalk.java
    - Additional procedures:
      – XML ID creation UUID
      – RDF creation
      – metadata digest creation
      – full content base64 encoding
DIDLCrosswalk

- DSpace API procedures for complex objects
  - Item.java:DSpace:Item = DID.Item {DC}
  - Bundle.java:DSpace:Bundle = DID.Item
  - Bitstream.java:DSpace.Bitstreams = DID.Component
  - BitstreamFormat.java to obtain secondary information
  - BitstreamStorageManager.java DSP:bitstream = DID.Resource
- Additional procedures
  - XML ID creation UUID
  - RDF creation
  - metadata digest creation
  - full content base64 encoding
LANL DSpace plug-in: further considerations

• DSpace DIDL plugin tested at LANL and Ghent University
• Issues encountered:
  o Lastmodified and OAI-PMH datestamp issues
  o Memory issues and the MAX_RECORDS
  o DSpace plugin implementation framework
Outline

(1) Motivation

(2) OAI-PMH for content

(3) Example 1 : LANL Repository

(4) Example 2 : mod_oai

(5) Example 3 : DSpace plug-in prototype

(6) Federations of IRs and OAI-PMH

(7) Conclusion
Harvesting COs from OAI-PMH repositories

OAI-PMH identifier = CO-identifier
OAI-PMH datestamp = datetime of ingestion/update
OAI-PMH response = COs

DSpace repository
baseURL(1)

FEDORA repository
baseURL(2)
Repository Index: listing OAI-PMH repositories of a federation

STEP 1: ListIdentifiers (OAI-PMH)
- baseURL(1)
- baseURL(2)
- baseURL(3)

STEP 2: ListRecords (OAI-PMH)
- List of COs

Repository Index
- baseURL(1): 2002-11-12
- baseURL(2): 2003-01-15
- baseURL(3): 2003-02-20

DSpace repository
- baseURL(1)

FEDORA repository
- baseURL(2)

Repo Index
- baseURL(index)
Identifier Resolver: locating COs in the OAI-PMH federation

<table>
<thead>
<tr>
<th>identifier resolver</th>
</tr>
</thead>
<tbody>
<tr>
<td>identifier</td>
</tr>
<tr>
<td>CO-id 1</td>
</tr>
<tr>
<td>CO-id 2</td>
</tr>
<tr>
<td>CO-id 3</td>
</tr>
</tbody>
</table>

- **CO-id**: CO-identifier
- **baseURL**: Base URL of the repository
- **datestamp**: Date of the repository change
- **repository**: Type of the repository

Diagram:
- **CO-id** is linked to **Identifier Resolver**
- **baseURL & CO-id** is monitored by **Repo Index**
- **Repo Index** exposes **baseURL(index)** for **DSpace repository** and **FEDORA repository**
Single point of OAI-PMH access to COs in the federation
OAI-PMH Federator in a distributed architecture

OAI-PMH Federator 1

OAI-PMH Federator 2

OAI-PMH Federator 3

Identifier Resolver

Repo Index

OAI 1

OAI 2

OAI 3

OAI 4

OAI 5
OpenURL gateway in a distributed architecture
Outline

(1) Motivation

(2) OAI-PMH for content

(3) Example 1: LANL Repository

(4) Example 2: mod_oai

(5) Example 3: DSpace plug-in prototype

(6) Federations of IRs and OAI-PMH

(7) Conclusion
Conclusion: OAI-PMH can be used to harvest content!

- OAI-PMH Advantages:
  - Simple yet powerful protocol.
  - Efficiency through selective & incremental harvesting.
  - Active community. Tools available.
  - Well-established adoption in Digital Libraries, Institutional Repositories, Archives
  - OAI can help (and is very willing to do so):
    - oai-rights – ongoing - how to convey rights in the OAI-PMH framework
    - Could help define - profile(s) of - complex object models that meet the needs

- Complex model advantages:
  - Unambiguous mapping between identifiers and metadata/resources
  - By-reference pointers to resources can be ‘real’ URLs, not hdl, doi, purl
  - Complex models can have simple profiles