Using OAI-PMH for *Resource*Exchange

OAI Metadata Harvesting Workshop, JCDL 03

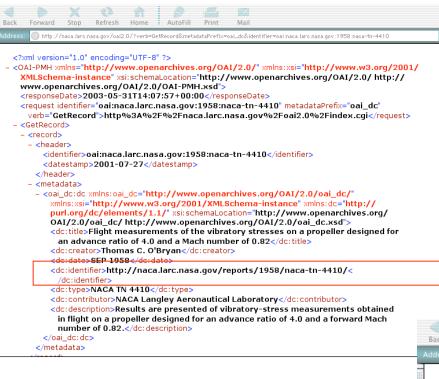
Michael L. Nelson, Terry L. Harrison
Old Dominion University
Norfolk VA
{mln,tharriso}@cs.odu.edu

OAI-PRH?

- using OAI-PMH for resource extraction / exchange
 - yes, OAI-PMH is for metadata not resources, but its going to happen anyway...
 - mirroring
 - preservation (archive "zipping")
 - convergence with OAIS
 - assumptions
 - a digital resource
 - rsync et al. neither appropriate nor possible
 - defer metadata vs. data discussion

Possible Approaches

- 1. Exploit knowledge outside the scope of the OAI-PMH to extract the resource
- 2. Base64 encode the resource and transmit via OAI-PMH as a separate "metadata" prefix?
- 3. Separate metadata prefix with instructions on how to extract / scrape the resource
- 4. Separate metadata format with XML encoded metadata, along with XSLT to decode it



direct pdf

Out of Band Knowledge

- 1. take url in dc:identifier
- 2. parse report number
- 3. append "reportnumber.pdf" to

```
Stop
                   Refresh Home
                                   AutoFill
                                             Print
                                                     Mail
     http://ntrs.nasa.gov/?verb=GetRecord&metadataPrefix=oai_dc&identifier=oai:magicnrc:arcrm3731
<?xml version="1.0" encoding="UTF-8" ?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/" xmlns:xsi="http://www.w3.org/2001/</p>
  XMLSchema-instance" xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/ http://
  www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
  <responseDate>2003-05-31T14:11:50+00:00</responseDate>
  <request metadataPrefix="oai dc" verb="GetRecord" identifier="oai:magicnrc:arcrm3731">http://
   ntrs.nasa.gov/index.cgi</request>
- <GetRecord>
  - <record>
   - <header>
       <identifier>oai:magicnrc:arcrm3731</identifier>
       <datestamp>2002-10-31</datestamp>
     </header>
   - <metadata</p>
     - <oai_dc:dc xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"</pre>
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:dc="http://
         purl.org/dc/elements/1.1/" xsi:schemaLocation="http://www.openarchives.org/
         OAI/2.0/oai_dc/ http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
         <dc:title>On the flow in an isentropic light piston tunnel</dc:title>
         <dc:type>ARC/R&M-3731</dc:type>
         <dc:creator>T. V. Jones, D. L. Schultz and A. D. Hendley</dc:creator>
         <dc:contributor>Aeronautical Research Council, Great Britain</dc:contributor>
         <dc:date>1973</dc:date>
         <dc:identifier>http://naca.central.cranfield.ac.uk/reports/arc/rm/
          3731.pdf</dc:identifier>
          dc:description>No Abstract Available</dc:description>
       </oai_dc:dc>
     </metadata>
   <ahout>
     - - - - orovenance xmlns="http://www.openarchives.org/OAI/2.0/provenance"
```

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation=

Out of Band Knowledge

- pros: tailored, no "accidental" harvesting
- cons: not scalable wrt # of repositories & harvesters, false negatives

	no metadata change	metadata change	
no data change	ok	unnecessary download	
data change	missed update!	ok •	

assumption: change in metadata means a change in data -- not always true!

Base64 Encoding

- define separate metadata formats
 - base64:application/pdf
 - base64:application/powerpoint
- pros: describable with OAI-PMH semantics, accomplished with standard OAI-PMH tools
- cons: heavyweight (could use compression), suitable for simple objects only, accidental harvesting would produce high loads for repositories and harvesters

Metadata as Instructions

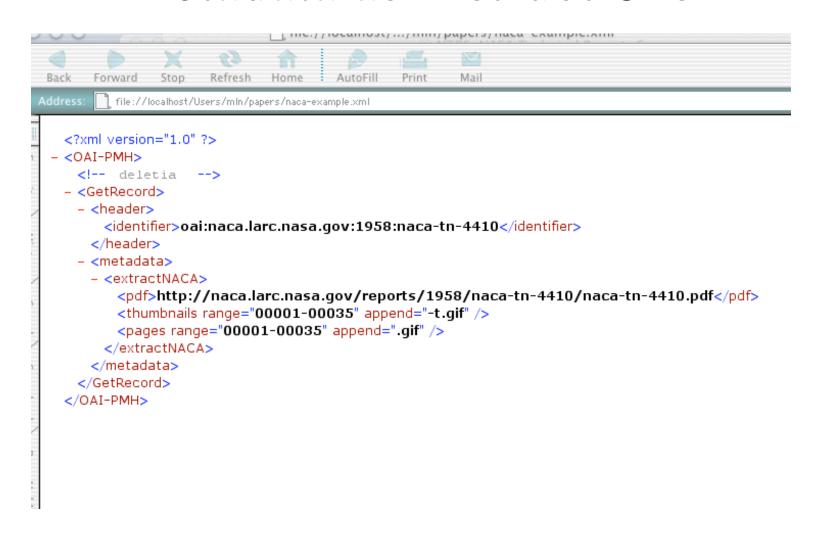


cf. http://genomebiology.com/2003/4/6/R40

Metadata as Instructions

- the resource described in <dc:identifier> could be a complex object
 - may not be appropriate to:
 - "tar" the object into a single file
 - expose all constituent objects through OAI-PMH
 - define a metadata prefix that provides machine readable instructions on how extract the complex object
 - METS?

Metadata as Instructions



XSLT

- if the resource is already XML encoded, include an XSLT to transform into the desired format
 - use separate metadata formats or even sets for the harvester to express their transformation preferences?
- pros: elegant, limited work for repository
- cons: assumes client-side transformation capability, applicable only for XML-encodable resources