The University of Chicago Library Digital Repository (LDR)

Charles Blair

2015-01-27:14-00-00

Contents

1 Past

1.1 Foundational Documents

1.1.1 Preserving Digital Information: Report of the Task Force on Archiving of Digital Information (Garret and Waters, 1996) [PDF]

Emulation; migration.

1.1.2 Reference Model for an Open Archival Information System (OAIS) (June 2012) [PDF]

SIP, AIP, DIP.

1.2 Planning Documents (Historical)

1.2.1 Report of the Digital Archiving Task Force (August 2004)

- electronic mail (LDR has some; some is being lost; what about box.com, etc.?)
- web pages (LDR has some)
- administrative records (LDR has some)
- instructional materials (IR?)
- research datasets (IR)

1.2.2 Recommendation for a Library Program for Digital Archiving (February 2005)

Key functions (following Reference Model for an Open Archival Information System [OAIS]):

- Deposit/ingest
- Discovery

- Dissemination/access
- Deletion/withdrawal

Discovery and access are orthogonal: you can see that we have it, but if it is embargoed, you may not have it until the embargo expires.

1.2.3 Joint Library & NSIT Digital Preservation Project Proposal (September 2005)

This was meant to address the electronic mail piece with an "Archive-It" function in the user's mail agent. However, resources were pulled from the project.

The decision was made to proceed independently. A key hire was made on the basis of this decision: Tyler Danstrom (DLDC).

1.2.4 Planning for a University of Chicago Digital Repository (Winter 2012)

Pages 5 and 6 defined the need for Laura Alagna's position (see below).

2 Present

Ingest requires significant prior work. Tyler can state the problem succinctly. Because we are dealing for the most part with materials from Special Collections (the University Archives), it makes the most sense to follow an archival workflow: transferring (SCRC); accessioning (SCRC, with support from the DLDC); processing (DLDC, with support from SCRC). Non-archival materials (e.g., maps from Chris Winters) can be made to follow this model seamlessly. So the model is:

1. accessioning (deposit)
2. processing (ingest)
3. discovery and access (dissemination)

Another key hire was made on the basis of the need for this workflow: Laura Alagna (SCRC)

2.1 Accessioning

Digital Repository Workflow

2.2 Processing

The OAIS reference model defines a submission information package (SIP) for ingest, an archival information package (AIP) for storage, and a dissemination information package (DIP) for access.

SIPs and DIPs require processing according to a standard. AIPs are system-specific. Discovery also depends on a standard (except in systems which rely solely on keyword searching).

2.2.1 SIPs

Standard packaging formats include METS (Metadata Encoding and Transmission Standard) for digital library objects, MPEG-21 DIDL (Digital Item Declaration Language), promoted by the Los Alamos Digital Library for complex digital objects as an alternative to METS, SCORM (Sharable Content Object Reference Model) for learning objects, the Matroska and QuickTime container formats for multimedia, and others. Problems with using some of them include:

- the learning curve can be high
- implementations often vary, leading to the need for "application profiles"
- a format might make assumptions about what kind of object is to be packaged, making implementation unwieldy or even impossible
- a format might not be suited for every kind of object that needs to be packaged
- the cost of implementation can be high

A very flexible format for object exchange is BagIt, which we use to transfer objects form the LDR to APTrust. BagIt resulted from the need to transfer several terabytes of data from the California Digital Library to the Library of Congress.[1] However, while BagIt is a good packaging format, it is not "semantically" useful; that is, it does not say anything about how objects in a package relate to one another or what they mean.

What is needed is a standard that is flexible enough to model all of one's data at a relatively low cost of implementation, but precise enough to provide needed commonality among all of one's data for discovery, which is the end goal of processing. While a descriptive metadata standard such as Dublin Core can do the latter (though with some loss of precision), it cannot do the former. A data model can, in particular, the Europeana Data Model (EDM), which was designed to model all manner of cultural heritage objects, such as those found in the LDR. EDM is based on OAI-ORE (Open Archives Initiative Object Reuse and Exchange), developed by the same group which developed OAI-PMH (The Open Archives Initiative Protocol for Metadata Harvesting). However, EDM adds some useful things on top of OAI-ORE, extending it in a way that makes it useful for modelling aggregations of objects beyond aggregations of web resources, for which it was initially developed. EDM has been adopted and adapted by the Digital Public Library of America (DPLA).

The key concept in the EDM data model is the provided cultural heritage object, or edm:providedCHO. The equivalent concept in DPLA is the dpla:sourceResource. Neither of these names wins points for elegance. Borrowed from OAI-ORE is the notion of a proxy for the provided cultural heritage object. A proxy consists of descriptive metadata.

[1] BagIt is specified in an Internet draft co-authored by John Kunze of the California Digital Library, last revised on January 28th, 2014.

There can be more than one proxy for the object, which allows for varying ways of describing it. For example, one might provide both TEI (Text Encoding Initiative), if one has it, and Dublin Core. Both can proxy the object, and be put to different uses. Rich metadata can be used to support community-specific impementations, such as the Goodspeed Manuscript Collection, while Dublin Core allows for simple discovery and metadata sharing using OAI-PMH.

EDM allows one to model a repository as a collection of collections, a collection as a collection of, say, titles, titles as a collection of volumes, volumes as a collection of issues, issues as a collection of pages, and pages as a collection of files, representing, for example, the page image and OCR data if available. This fully recursive model means one does not have to invent special vocabularies at each level of the hierarchy; one can re-use elements of the model. It should be clear that the archival notion of collection fits nicely with this model, as does the notion of digital collection.

I have not even scratched the surface of EDM. Suffice it so say that the LDR implements all of the required EDM elements. This means that anyone with a knowledge of EDM, which is independently documented, knowing that the LDR implements all of the required EDM elements, can explore the LDR without knowing anything more about it than that.

Because EDM is represented as directed, labelled graphs, it is linked data. It can be expressed as XML, the same as METS, MPEG-21 DIDL, or SCORM, but because it is linked data, it does not have to be. I like to express linked data as Turtle, or Terse RDF Triple Language. RDF, or Resource Description Framework, is a way of expressing arbitrary metadata as directed, labelled graphs, which is what linked data is. There are several ways of writing Turtle. I like the form that looks the least like XML, for example, this:

```
@prefix edm: <http://www.europeana.eu/schemas/edm/>.
@prefix dc: <http://purl.org/dc/elements/1.1/>.
@prefix dcterms: <http://purl.org/dc/terms/>.
@prefix premis: <info:lc/xmlns/premis-v2>.

<ac/s9gx23kjrzvh8/2014-006/mvol/0002/0017/0001/mvol-0002-0017-0001.pdf>
dc:format "application/pdf";
dcterms:isFormatOf <http://ldr.lib.uchicago.edu/ac/s9gx23kjrzvh8/
2014-006/mvol/0002/0017/0001/mvol-0002-0017-0001>;
premis:objectIdentifierType "ARK";
premis:objectIdentifierValue <http://ark.lib.uchicago.edu/ark:/61001/ac/s9gx23kjrzvh8/
2014-006/mvol/0002/0017/0001/mvol-0002-0017-0001.pdf>;
premis:objectCategory "file";
premis:compositionLevel 0;
premis:messageDigestAlgorithm "SHA-256";
premis:messageDigest "4f6237c25a51382c3f6c489e550f3b2a241574abbfc57adbf9e0f9b6c674b1a5
premis:messageDigestOriginator "/sbin/sha256";
premis:size 31011220;
premis:formatName "application/pdf";
premis:originalName "mvol-0002-0017-0001.pdf";
premis:eventIdentifierType "ARK";
premis:eventIdentifierValue "s9gx23kjrzvh8";
premis:eventType "creation";
premis:eventDateTime "2014-01-21T11:24:06"^^xsd:dateTime;
a edm:WebResource.
```

We are looking at 17 independent assertions, each an RDF triple, but presented in a way familiar to those who are used to dealing with records. All assertions are predicated of the subject at the top, in this case a PDF file which is also a web resource, or something I can find on the web. Each assertion consists of a predicate and object separated by a semicolon. The type statement at the end is followed by a full stop, ending the assertions for this subject. All of the assertions in the LDR are typed. This allows one to search assertions by type.

If we can point to a digital object using EDM, nothing prevents our making other assertions about it as well. The linked data approach not only allows but encourages this. Thus the technical metadata which Laura generates for each file are recorded using the set of elements defined by the PREMIS Data Dictionary for Preservation Metadata, version 2.2.[2] All required elements are present. The LDR is both EDM-conforming and PREMIS-compliant. The linked data approach, using an underlying data model, allows for rigor on the one hand, flexibility and extensibility on the other.

These Turtle statements constitute our SIPs. Tyler produces them programatically, from the data Laura enters into the database, from the technical metadata she generates automatically, from knowledge of the repository file structure, which is itself a kind of flat-file database, according to the EDM and PREMIS specifications I provided. They are well-structured, well-documented, well-understood, lightweight, flexible assertions about the contents of the LDR at the file level, which is at bottom what we must manage, as well as at the level of the intellectual object, whether page, issue, volume, title or collection. I like to think of them as possessing very high tensile strength despite their light weight and flexibility.

2.2.2 AIPs

SIPs are loaded into MarkLogic, a commercial NoSQL database which we license, and which in its current version supports linked data. I have looked at the roadmap for the next two versions, and linked data support, already very robust and easy to work with, will only get better.

[2] "The PREMIS Data Dictionary for Preservation Metadata is the international standard for metadata to support the preservation of digital objects and ensure their long-term usability." http://www.loc.gov/standards/premis/

2.2.3 DIPs

There is a standard query language for RDF, which forms the basis of linked data, called SPARQL. RDF and SPARQL, like XML, HTML, and CSS, among others, are defined by the World Wide Web Consortium, the main international standards body for the World Wide Web. Thus the technology which makes the processed component of the LDR available for discovery conforms to international standards, which have been implemented not only by libraries and archives but also by government and industry. This means that the resources available for doing this work are large and not confined to the world of libraries.

DIPs are produced as needed by SPARQL queries. A query is crafted against the LDR to produce linked data suited for the purpose. For example, a researcher in England might want the digital masterfiles and only the digital masterfiles for our Chopin collection produced after the last time he made that request. Since we record events in the accessioning database, and since we know, on the basis of our data model, what the masterfiles for this collection are, we can produce the links to those objects, and either use the links to retrieve the objects and put them on disk, or simply send him the links. We can do the same for those objects needed for Campus Publications. We can work with faculty to tailor a request to just those objects he or she needs, perhaps in an iterative fashion. For example, they might ask, Tell me what you have for 1968. If the count is huge, we can hone in on what exactly might be useful.

`http://ldrp.lib.uchicago.edu/`

```
sem:sparql(fn:concat('
PREFIX cts: <http://marklogic.com/cts#>
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX edm: <http://www.europeana.eu/schemas/edm/>
PREFIX ore: <http://www.openarchives.org/ore/terms/>

SELECT DISTINCT ?what ?date ?about ?pi ?pdf ?object
FROM <http://lib.uchicago.edu/campub>
WHERE {
?s dc:title ?what . filter cts:contains(?what, cts:word-query(?title, "case-insensitive
OPTIONAL {?s dcterms:description ?about}
?s edm:year ?when . filter cts:contains(?when, cts:word-query(?year)) .
?s dc:date ?date .
?s ore:proxyFor ?where .
?where dcterms:hasPart ?page . ?ocr ore:proxyFor ?page .
?ocr dcterms:description ?word . filter regex(?word, ?keyword, "i")
?aggregation ore:aggregates ?where
OPTIONAL {?aggregation edm:isShownAt ?pi}
OPTIONAL {?aggregation edm:isShownBy ?pdf}
{?aggregation edm:object ?object}
} ORDER BY DESC(?date) ?what', ' LIMIT ', $limit, ' OFFSET ', $offset), $map)
return sem:query-results-serialize($result)
```

3 Future

- Tyler is working on a front end to allow Laura to specify those components needed to make SIPs, which can then be produced programatically
- I need to model the rest of our digital collections
- Fred needs to productionize MarkLogic 7
- We need to wrap our OAI-PMH provider around the LDR to provide metadata for VuFind
- Because we can now link to data in the LDR, we need to think about when it is useful to do so from our digital collections, for example, linking to PDFs for Campus Publications
- We need to modify our BagIt script to pull from our processed collections, not only directly from accessions
- We need to model EADs for archival and manuscript collections in the LDR, examples of which SCRC will provide