




The Portal to Texas History:

a new look, new functionality, and new
technical infrastructure




Digital Object Model
Flexible Metadata Model
Scalable Infrastructure



Infrastructure Discussion



Started off with a simple digital object model




1 metadata record + 1 file = digital object



But we have books...

We broke our model




1 metadata record
+
1 or more files
=
digital object



Can we OCR our book pages and have those
indexed for search?

We broke our model



1 metadata record
+
1 or more sets of files
=
Digital object



Set of files = fileSet

Think of a scanned book page:

- 1 archival tif
- 1 thumbnail jpg
- 1 medium size jpg
- 1 large web jpg
- 1 OCR text file
- 1 file of bounding box coordinates

All are part of one fileSet



Think of a scanned page of a letter
(in spanish)


- 1 archival tif
- 1 thumbnail jpg
- 1 medium size jpg
- 1 large size jpg
- 1 translation
- 1 transcription

All are part of one fileSet



Think of a newspaper page:

- 1 archival tif
- 1 thumbnail jpg
- 1 med size jpg
- 1 large size jpg
- 1 pdf of page
- 1 jpeg200 image
- 1 OCR text file
- 1 bounding box file



1 metadata record
+
1 or more sets of files
=
Digital object

good...



Wait!!!

We have:

A scanned report = 100 fileSets

A pdf version of report = 1 fileSet

A txt version of report = 1 fileSet

An html version of the report = 40 files

It has 1 metadata record

And should only be one object in the system



Digital Object

Metadata Record

Manifestation 1

fileSet 1

fileSet 2

fileSet 3

Manifestation 2

fileSet 1

Manifestation 3

fileSet 1

fileSet 2



Now we need to store our digital object model

METS = Metadata Encoding and Transmission Standard

Big scary XML!!!



METS is a library “standard”

Allowed us to model our digital object...

Extensible
(for when we decide to add stuff)

Like page numbers
Like sections of a book



Descriptive Metadata Scheme



Started with Dublin Core



We took simple Dublin Core

dc:title = Moonrise, Hernandez, New Mexico, 1941
dc:creator = Ansel Adams

And locally qualified it.

dc:title.official = Moonrise, Hernandez, New Mexico, 1941
dc:creators.photographer = Ansel Adams

Using all sorts of library standards



We also added fields to help us keep things organized...

Institution = owning institution

Collection = a way to organize objects in our system

Primary Resource = good to know

Degree Information = Major, Level, School, College

Note = everyone needs a note field

Meta = Information about the record



Oh yeah we qualified some of those too:

untl:institution = Hardin-Simmons University
untl:collection = Hardin-Simmons University Yearbook
untl:primaryResource = True
untl:note.public = missing page 5
untl:note.private = we should find a better can of page 6
untl:meta.ark = ark:/67531/metaph45302
untl:meta.dateRecordCreated = 2008-03-03T03:02:01
untl:meta.recordCreator = mphillips



We decided on an identifier scheme:

We selected Archival Resource Keys (ARK)



ark:/67531/metaph18826

Has 4 parts:

ARK Label = ARK:

Name Assigning Authority Number = 67531 (or UNT Libraries)

Name = metaph18826 (our identifier internally for object)

Qualifier = stuff after the name (more in a second)

On the web it looks like this

<http://texashistory.unt.edu/ark:/67531/metaph18826/>



We mapped our digital object model to ARK

ark:/67531/metaph18826/ = object

ark:/67531/metaph18826/m1/ = manifestation 1

ark:/67531/metaph18826/m1/1/ = first fileSet of manifestation1

ark:/67531/metaph18826/m1/1/ocr/ OCR file of fileSet 1 manifestation 1

We have everything mapped like this


ark:/67531/metaph18826/
ark:/67531/metaph18826/?
ark:/67531/metaph18826/??
ark:/67531/metaph18826/thumbnail
ark:/67531/metaph18826/metadata/
ark:/67531/metaph18826/metadata.dc.xml
ark:/67531/metaph18826/metadata.untl.xml
ark:/67531/metaph18826/metadata.mets.xml
ark:/67531/metaph18826/citation/
ark:/67531/metaph18826/m1/
ark:/67531/metaph18826/m1/1/
ark:/67531/metaph18826/m1/1/thumbnail/
ark:/67531/metaph18826/m1/1/ocr/
ark:/67531/metaph18826/m1/1/zoom/



Why?

It allows everyone who works with our digital objects access to all parts of them.


Programmers
Designers
Developers
Metadata Librarians
Researchers, Students, Public



We used these three concepts to build the
system



We call the system Aubrey



Aubrey takes a request for an object
Retrieves the METS Record
Retrieves the metadata Record
Creates the requests “view” of the object



Other pieces that are required



Number Server for minting identifier names



Controlled vocabulary management system



Hierarchical subject management system



Collection/Partner management database



Metadata Creation Tools (web based)




Full-Text Search Service



Metadata versioning system




User authentication system



Built with open-source tools
Built on open-source platforms

Tools with large user communities

Tools Like...



Ubuntu
Django
Apache
Memcached
PerIBal
OpenLayers
Solr
Lucene
Python

And many others



Built with scaling in mind.



What if we get a ton of traffic?



What if we get a ton of content?



What if we outgrow one component?



What if one of our servers goes offline?



Or if the latest and greatest tool comes along




We thought about these and other questions



And arrived at a scalable system

That grows in the way that we grow

And is inexpensive to scale



Digital Object Model
Flexible Metadata Model
Scalable Infrastructure



Any Questions?