GIS Data Management

Eliot Jordan (Princeton)
Darren Hardy (Stanford)
We'll be talking about GIS Data Management. According to the schedule we'll be covering data modeling for geospatial assets, web services for GIS data, and practices for data wrangling.
Like Kim and Darren, I would also like to talk about workflows.
What are the steps that we go through with our geospatial datasets and images? Where are they stored, how do we create metadata, where do we store this metadata, how do we make them searchable and discoverable.
What do we do with our data?

What do we do with all this data that we collect?
Stanford has some really nice tools, but we're not all as advanced. I'd like share some of our data management practices at the Princeton University Library. It's a little embarrassing at times, but perhaps some of you can relate our challenges. This is a typical workflow from several years ago.
First thing to do is go down to Babbages at the mall and pick up a shrink wrapped copy of Oregon Trail, and a sets of shapefiles of roads in Kazakhstan. Or we might send away for some interesting data through the mail.
The data arrives on CD-ROMS or DVDs.
The purchase is logged and the discs are shoved onto a shelf in an obscure office waiting for their day in the sun.
Some of the lucky ones are zipped up and indexed into an ArcIMS Metadata Explorer, but the process is so time consuming that an enormous backlog develops.
What’s wrong with this scenario? It’s pretty obvious that the main problem here is that these datasets are extraordinarily difficult to discover and access. How do patrons even know that we own a dataset, let alone understand where to go to get access to it.

What’s wrong with this?

• discoverability
• accessibility
• time consuming
Since then, we've improved our processes and habits quite a bit, but we're not 100% there yet. What happens right now is this. Data is copied from disc or is downloaded onto a hard drive.
FGDC metadata is created or enhanced in ArcCatalog.
The data is zipped up and placed in a pair-tree directory structure on a windows file server.
Then, the metadata is ingested into a home-grown search portal, which is really not much more than a thin layer on top of Apache Solr, and then made available for download.
Our scanned map workflow is similar, except that it usually involves working with MARC records in our catalog system.
I'm sure that looks familiar.
This whole process involves tons of spreadsheets, an access database or two...
and a huge amount of work done by hand. Something like artisanal data management.
How can we better manage our GIS data?

How can we be smarter. How can we be more efficient? How can we better manage the GIS data in our institutions and libraries?
One answer is with a digital repository. A repository like Project Hydra. I pretty most of you have at least heard of hydra, but for those that haven’t, it's an open source, community driven, repository solution for digital content.
It's built in ruby-on-rails on top of Apache Solr, Blacklight, and the Fedora Commons repository system.
Fedora Commons is itself an open source system for data storage, preservation, and access.
This is a screenshot of the current Fedora web front-end. The latest hydra applications use fedora version 4 which stores metadata as linked data in the form of RDF triples.
The big story here is the mandatory RDF metadata and the use of something called the Linked Data Platform protocol to communicate with clients, such as hydra.
An outgrowth of this change in technology was the creation of the Portland Community Data Model.
What is PCDM?

A shared and flexible domain model for use in a wide array of repository and digital asset management applications.

PCDM is a shared and flexible domain model, initially drafted in Portland Oregon, for use in a wide array of repository and digital asset management applications.
Linked data best practices

• using URIs to identify resources
• using common vocabularies when possible
• subclassing existing classes and properties

It encourages linked data best practices such as using URIs to identify resources, using common vocabularies when possible, and subclassing existing classes and properties when creating new terms.
The PCDM base model provides for files, objects, and collections. A collection is a group of resources. An object is an intellectual entity, often called a "work" or a "digital object." A File is a sequence of binary data that is described by some accompanying metadata.
In June of 2015, the Hydra Geospatial Interest Group chartered a GIS Data Modeling Working Group. The members of the group, besides myself, are James Griffin from Lafayette College, Darren, John Huck from Alberta, Eric James from Yale. The prime directive of the group is to create PCDM implementations of GIS data models. We decided that a comprehensive model for “georeferenced maps” would be a good place to start. And this is what we came up with.
I'll walk you through this a bit. We'll start on the left side.
An our model, an image work or a scanned map, has a title and a rough bounding box. It has an image file object which contains the original tiff and a JPEG2000 derivative image. The scanned map can have an external metadata file such as MODS or MARC.
Moving on to the right. The scanned map might then be geo-rectified at which point it has a child geo-referenced raster work.
That work has a raster file object which contains a GeoTIFF and again derivative jp2. The raster work can have external metadata such as ISO 19139 or FGDC.
Lastly, someone could create a feature extraction from that raster work which then has a child vector work.
The vector work has a vector file object that contains the original shapefile, a derivative shapefile that's reprojected into WGS84, and a preview image.
Once we settled on the basics of the PCDM model, we set about to implement the model in a project called GeoConcerns.

https://github.com/projecthydra-labs/geo_concerns
This is a diagram of the various layers in the hydra stack for a Sufia repository.
Without going into too much detail, this the approximate location on the diagram where our project, GeoConcerns, lives. It's built on top of a ruby gem called HydraWorks, which adds models and behaviors around the concept of multi-file "works", and it extends the CurationConcerns gem which adds a basic User Interface and CRUD (Create, Read, Update and Destroy) functionality.
GeoConcerns is more like a rough sketch right now, but I'd like to try a live demo to show what some of this workflow might look like in practice.
In our next GeoConcerns works sprint we like to add functionality to generation shapefile derivatives, automated creation and loading of geobacklight schema documents, and loading of data into geoserver. Bridge the gap between the repository and the discovery layer.