PCDM & Data Modeling

Let's start modeling our digital repository objects.

# RDF (Resource Description Framework)

Briefly...

* Standard model for data exchange on Web
* RDF is made up of triples, i.e.
* Triple: resource\_uri predicate\_uri object\_uri
  + Subject predicate Object
* Can be serialized/stored in number of ways

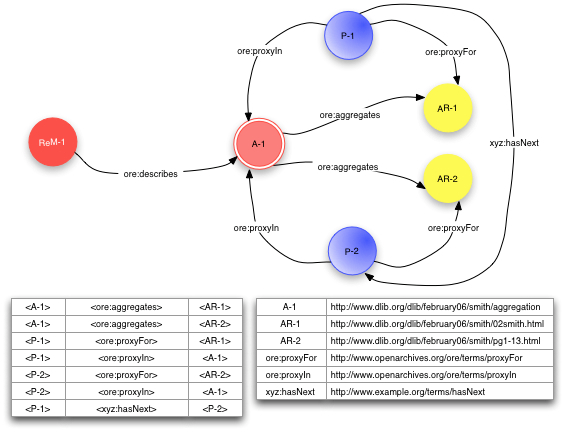
# PCDM: Portland Common Data Model

* Started Officially in 2015
* Community Effort to Make Digital Repository Objects More Interoperable
* Ongoing work & Discussion at:
  + [pcdm.org](http://pcdm.org/)
  + [github.com/duraspace/pcdm](https://github.com/duraspace/pcdm)
  + [groups.google.com/forum/#!forum/pcdm](https://groups.google.com/forum/#!forum/pcdm)

# PCDM Continued

* Models Namespace: <http://pcdm.org/models#>
* Models RDF File: [github.com/duraspace/pcdm/blob/master/models.rdf](http://github.com/duraspace/pcdm/blob/master/models.rdf)
* Builds off Object Reuse & Exchange (ORE) Data Model
* Interacts with other specifications (LDP), data store platforms (Fedora 4), but is meant to be neutral
  + Linked Data Platform (LDP) specifies CRUD operations performed on linked data resources available through HTTP (or for resources “on the web”).
  + Fedora 4 is a graph store for handling repository objects - storing structure, binaries, and metadata.

# ORE Abstract Model

<http://www.openarchives.org/ore/1.0/>

# PCDM Overview



<https://github.com/duraspace/pcdm/wiki>

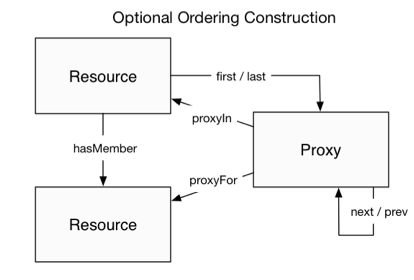
## PCDM Classes

* [pcdm:Object](http://pcdm.org/models#Object): An Object is an intellectual entity, sometimes called a "work", "digital object", etc...
* [pcdm:Collection](http://pcdm.org/models#Collection): A Collection is a group of resources...
* [pcdm:File](http://pcdm.org/models#File): A File is a sequence of binary data and is described by some accompanying metadata...
* [pcdm:AlternateOrder](http://pcdm.org/models#AlternateOrder): An AlternateOrder is an alternate ordering of its parent's members. It should only order the parent's members...
* [pcdm:AdministrativeSet](http://pcdm.org/models#AdministrativeSet)

## PCDM Properties

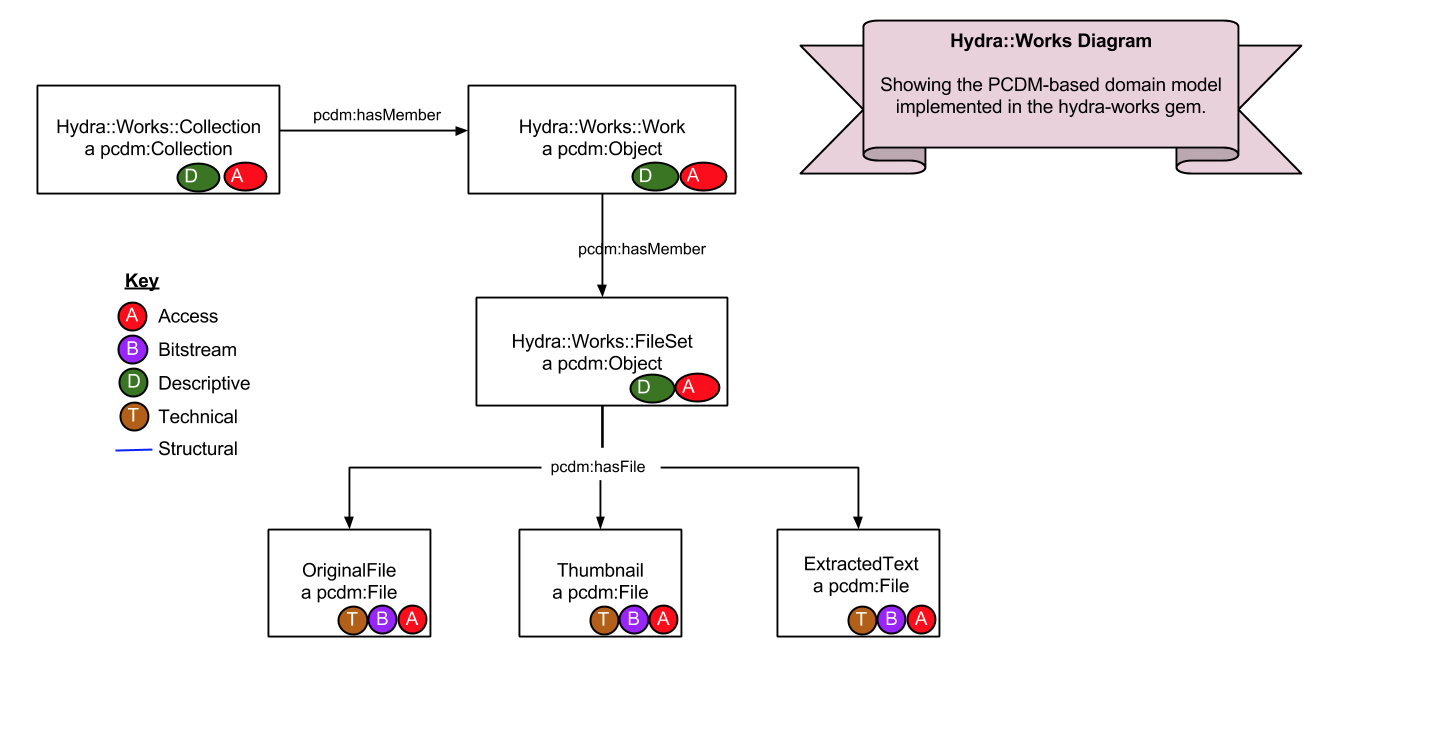
* [pcdm:memberOf](http://pcdm.org/models#memberOf)
  + Domain: ore:Aggregation | Range: ore:Aggregation
* [pcdm:hasMember](http://pcdm.org/models#hasMember)
  + Domain: ore:Aggregation | Range: ore:Aggregation
* [pcdm:fileOf](http://pcdm.org/models#fileOf)
  + Domain: pcdm:File | Range: pcdm:Object
* [pcdm:hasFile](http://pcdm.org/models#hasFile)
  + Domain: pcdm:Object | Range: pcdm:File
* [pcdm:relatedObjectOf](http://pcdm.org/models#relatedObjectOf)
  + Domain: pcdm:Object | Range: ore:Aggregation
* [pcdm:hasRelatedObject](http://pcdm.org/models#hasRelatedObject)
  + Domain: ore:Aggregation | Range: pcdm:Object

## PCDM Ordering



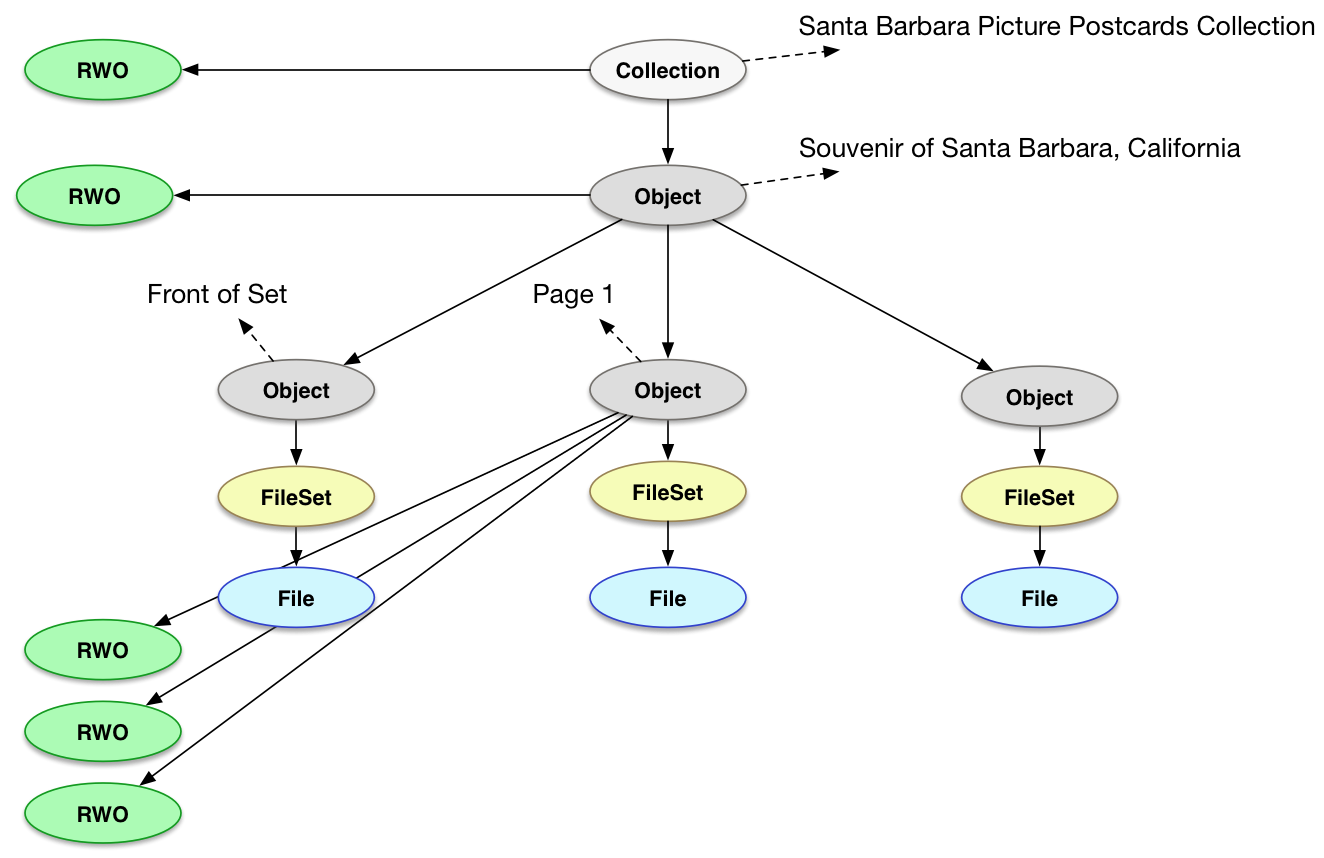
<https://github.com/duraspace/pcdm/wiki>

## PCDM (Works) ... TBD?



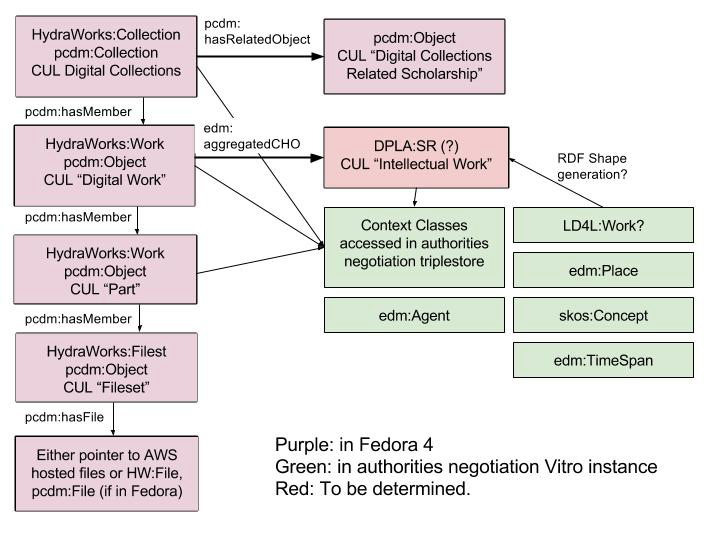
<https://github.com/projecthydra/hydra-works>

## Classic Example: Postcards



<https://github.com/hybox/models/blob/master/notes/usecase2.md>

## Cornell's PCDM

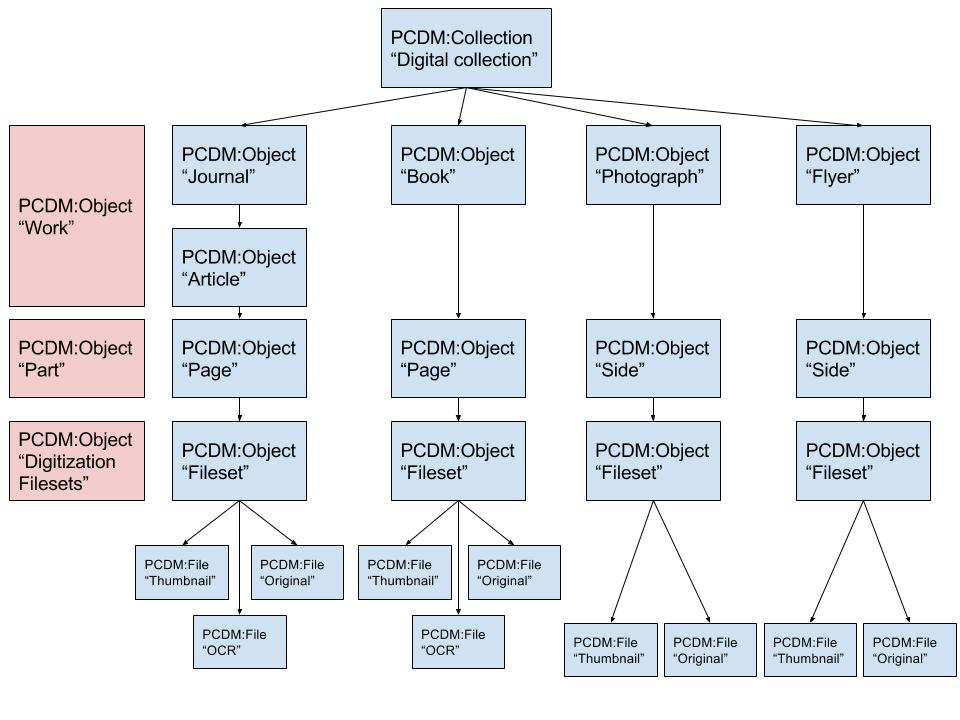


# And Now You...

Building off the metadata review for your objects:

1. Group Your Entities into Possible PCDM Classes;
2. Map fields to relationships between Objects;
3. Try Drawing Model in Google Drawings;
4. What Fits? What Doesn't?
5. Start thinking about properties: [Linked Open Vocabs can help! http://lov.okfn.org/](http://lov.okfn.org/dataset/lov/terms)

# Filling in the Details: Classes



# Filling in the Details: Properties

* Becomes Easier to Determine with Clearer Classes
* Be Aware of Predicate Domains & Ranges
* DCMI Publishing Guidelines Helpful Example
* This supports Machine Inference, Not Really for Validation Purposes

# Right now...

Building off the metadata review for your objects:

1. Create your group's PCDM Model in Google Drawings;
2. Map fields to relationships between Objects;
3. Review mapped properties in merged MAP
4. What Fits? What Doesn't?