Theory to Practice:

Linked Open Data with OpenRefine

Christina Harlow, @cm_harlow

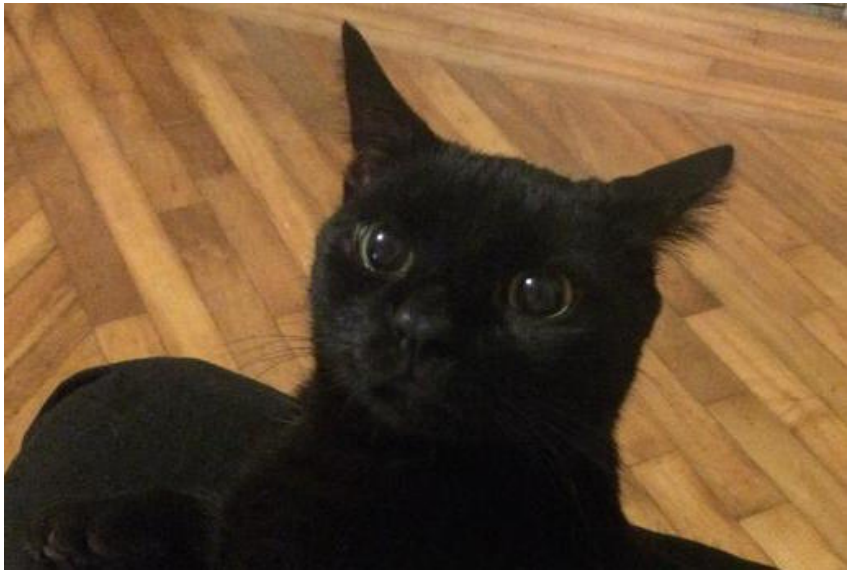
LODLAM Toronto 2016

Slides, Examples, + Install

https://github.com/LODLAM/LODLAMTO16

Installation Backup

Go to Installation instructions & follow RefinePro options



Agenda

1. Introduction
2. Sample Project
3. Importing XML Data
4. Data Munging
5. Reconciliation
6. Mapping & Exporting RDF
7. Wrap-up

Quick Introduction

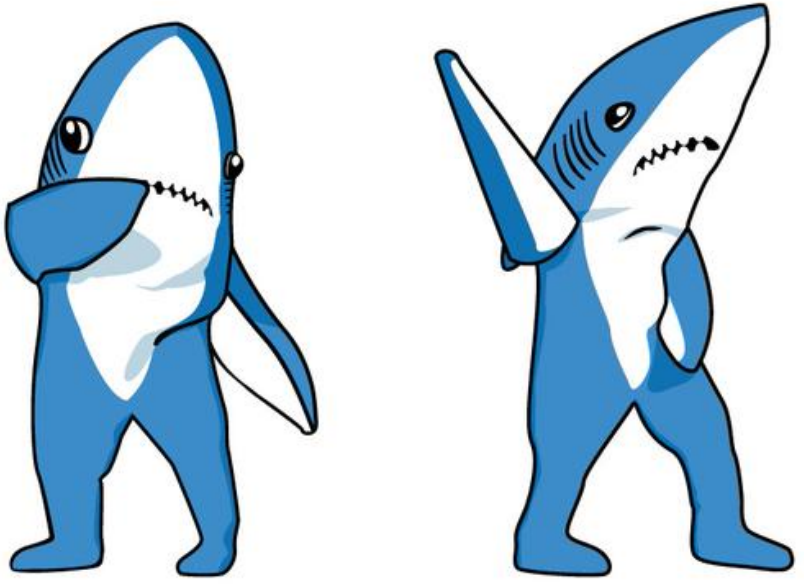
1. **Introduction <==**
2. Sample Project
3. Importing XML Data
4. Data Munging
5. Reconciliation
6. Mapping & Exporting RDF
7. Wrap-up

Learning LOD by Working with LOD

Goal: Learn Linked Open Data by working with it in context of Libraries, Archives, & Museums metadata

Need Help: Raise Hand, Ask Friend, Review Instructions, Check Online

Let's All Left-shark It



"Hacker School Rules"

- No feigning surprise
- No well-actually's
- No back-seat driving
- No subtle -isms

https://www.recurse.com/manual

Quick Intro to OpenRefine

- OpenRefine = power data tool
- Since 2012, community-sourced
- OpenRefine.org
- github.com/OpenRefine/Openrefine
- Java (& Jetty) app running locally
- GUI runs in your chosen browser (**NOT INTERNET EXPLORER**)

☆ NOT ☆

☆ INTERNET ☆

☆ EXPLORER ☆

OpenRefine & RDF

- Native importing of RDF/XML, NTriples
- Freebase Extension
- DERI RDF Extension, LODRefine
  - RDF & SPARQL Reconciliation
  - RDF Skeleton, Mapping
  - RDF Export: RDF/XML, Turtle

Not Just Producing RDF...

Using RDF data & tools like
OpenRefine = better entity matching

Possible Influences/Related Tools:

- VIVO Recon Service
- Nomenklatura
- Ecco!
- Karma

DERI RDF Extension & LODRefine

!!! No longer actively supported !!!

Each complaint re:slowness, bugs = 1

If we reach 30, we all will learn Java + maintain our own tools

Our Sample Project

1. Introduction
2. **Sample Project <==**
3. Importing XML Data
4. Data Munging
5. Reconciliation
6. Mapping & Exporting RDF
7. Wrap-up

Fedora 3 => Fedora 4

1. Importing sample DC/XML metadata to make into PCDM RDF
2. Import your own metadata & DIY it

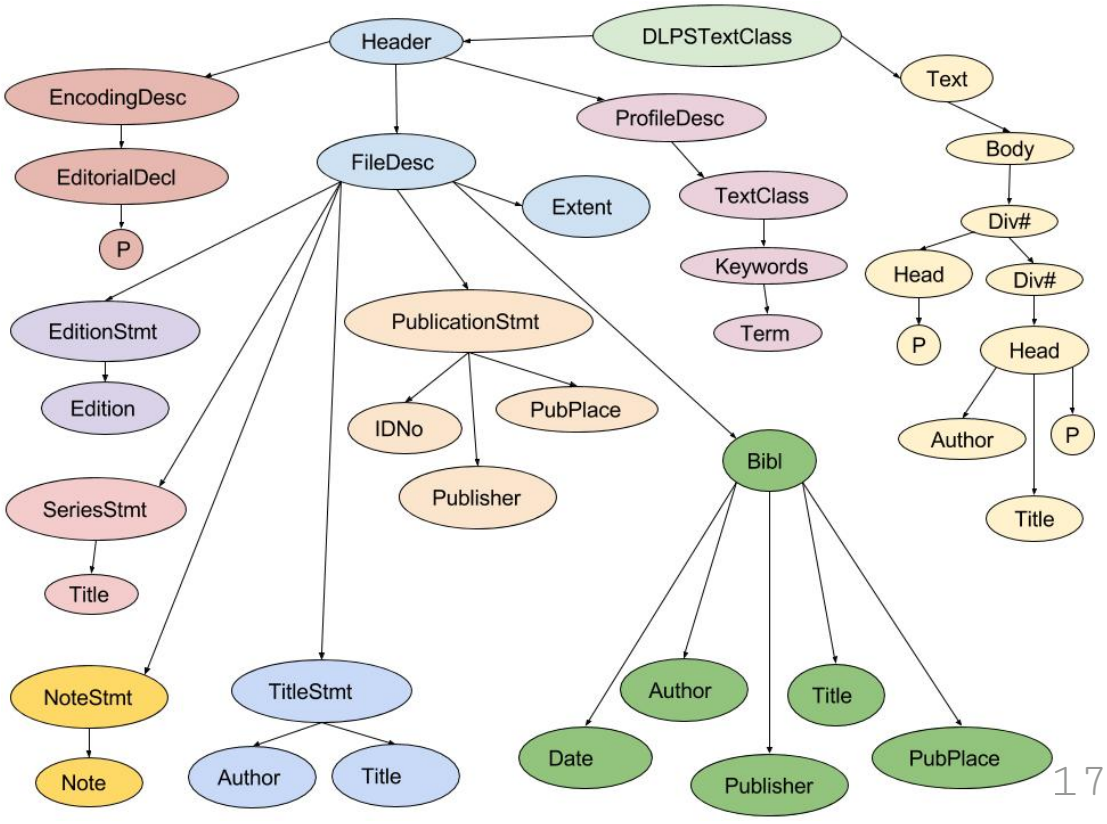
DLXS XML to PCDM RDF

DLXS = Digital Library Extension Service

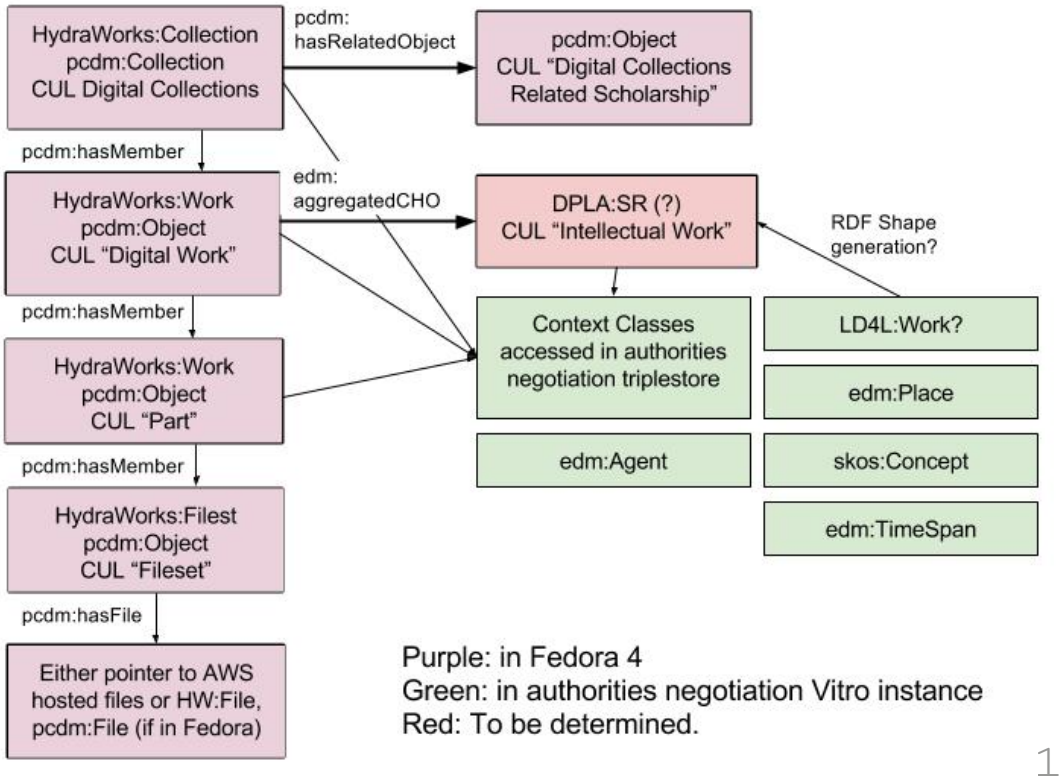
PCDM = Portland Common Data Modeling

From a Live Fedora 4 Migration

The DLXS Past



The PCDM Future *In Flux*



Purple: in Fedora 4
Green: in authorities negotiation Vitro instance
Red: To be determined.

Metadata Mapping

```
      /record/ENCODINGDESC/EDITORIALDECL/P: 124/124 | 100%
                  /record/FILEDESC/EXTENT: 124/124 | 100%
     /record/FILEDESC/PUBLICATIONSTMT/IDNO: 124/124 | 100%
 /record/FILEDESC/PUBLICATIONSTMT/PUBLISHER: 124/124 | 100%
  /record/FILEDESC/PUBLICATIONSTMT/PUBPLACE: 124/124 | 100%
    /record/FILEDESC/SOURCEDESC/BIBL/AUTHOR: 124/124 | 100%
      /record/FILEDESC/SOURCEDESC/BIBL/DATE: 124/124 | 100%
      /record/FILEDESC/SOURCEDESC/BIBL/NOTE: 124/124 | 100%
 /record/FILEDESC/SOURCEDESC/BIBL/PUBLISHER: 124/124 | 100%
  /record/FILEDESC/SOURCEDESC/BIBL/PUBPLACE: 124/124 | 100%
     /record/FILEDESC/SOURCEDESC/BIBL/TITLE: 124/124 | 100%
          /record/FILEDESC/TITLESTMT/AUTHOR: 124/124 | 100%
           /record/FILEDESC/TITLESTMT/TITLE: 124/124 | 100%
/record/PROFILEDESC/TEXTCLASS/KEYWORDS/TERM: 124/124 | 100%
               /record/TEXT/BODY/DIV1/HEAD: 124/124 | 100%
```

Importing Data

1. Introduction
2. Sample Project
3. **Importing XML Data <==**
4. Data Munging
5. Reconciliation
6. Mapping & Exporting RDF
7. Wrap-up

Import data into OpenRefine

1. Start up OpenRefine or LODRefine
2. Click on Create Project Tab
3. Click on Web Addresses (URLs)
4. Enter the URL for GitHub Raw Object of Starter Dataset you want to use

(Or download/save your metadata to working environment & use 'This Computer')

Import Your Data

Go ahead and import the data for this workshop:
OpenRefine_Tutorial/Data/Toronto_examples/hunt.xml

Bonus: Once your main project is created, export one of the sample RDF documents to see how it looks as an OpenRefine project. This differs from what the DERI extension expects.

Import data into OpenRefine

1. Preview your data as project
2. Change settings as needed
    - XML, Json: need to choose 'record' object
    - CSV, Excel: review for header rows
    - RDF: Preview options for loading
3. Once ready, give name, Create Project

Viewing OpenRefine Project

- Saved Automatically
- Undo / Redo Panel
- Rows/Records == VERY IMPORTANT
- Extensions, Export Options in Top Right
- Facet, Filter panel on left
- If something freezes, refresh the browser (gahhhh)

Data Munging

1. Introduction
2. Sample Project
3. Importing XML Data
4. **Data Munging <==**
5. Reconciliation
6. Mapping & Exporting RDF
7. Wrap-up

Metadata Munging in OpenRefine

Ways to Normalize, Remediate Data:

- Join, Split Rows
- Splitting, Renaming Columns
- Faceting, Clustering, Filtering
- Google Refine Expression Language (GREL)

github.com/OpenRefine/OpenRefine/wiki

Prepare Your Data

- Get columns renamed as reviewed, mapped
- Get cells joined
- Facet, review
- Facet, cluster, normalize
- Filter to target, map values to new fields

Reconciliation

1. Introduction
2. Sample Project
3. Importing XML Data
4. Data Munging
5. **Reconciliation <==**
6. Mapping & Exporting RDF
7. Wrap-up

OpenRefine Reconciliation

Reconciliation broadly: Compare values in my dataset with values in an external dataset, if deemed a match, link and pull in external datapoint information

Add column by fetching URL...

- HTTP requests to external data API in UI
- takes far longer to pull data
- requires parsing returned data with GREL

Standard Recon Service API

- RESTful API between OpenRefine and external data
- handles JSON reconciliation objects btwn datasource API + Openrefine

DERI RDF Extension

- no longer actively supported
- Standard Recon Service API to work with RDF, SPARQL endpoints
- RDF docs held in memory
- SPARQL recon dependent on SPARQL server details

Reconciliation Demos

- LCSH via SPARQL
- Languages via RDF Doc
- Geonames via Recon Service
- VIAF hosted service
- LCSH and LCNAF hosted service

OpenRefine Recon

1. Run Recon according to your choosing - see options in Recon instructions, links
2. Pull URIs for a particular field
3. Pull other information helpful for your projects
4. Make sure to pull in URIs, information

Mapping & Exporting RDF

1. Introduction
2. Sample Project
3. Importing XML Data
4. Data Munging
5. Reconciliation
6. **Mapping & Exporting RDF <==**
7. Wrap-up

DERI RDF Creation

RDF Extension button > Edit RDF Skeleton...

- Add Namespaces/Utilize Namespaces
- Can assign types, create blank nodes
- Preview the Output
- Save your skeleton
- Export > RDF...

Classes & Predicates...

- What class of PCDM does this description belong to?
  - PCDM:Collection
  - PCDM:Object Work or Part
  - PCDM:Fileset
  - PCDM:File
  - Context Class?
- Do the domain & range of your predicates work?
- It can be helpful to rename columns in this effort

Classes & Predicates Help

- PCDM Namespace RDF
- PCDM Docs
- DCMI Docs
- EDM Docs
- EDM Namespace RDF
- DPLA Docs
- CUL Mappings so far
- Variety of Mappings

Map & Export

Map your data to RDF using the RDF skeleton, preview the Turtle, then export when you're ready.

Bonus: Export your doc then use for a test RDF Doc reconcile.

Wrap-Up

1. Introduction
2. Sample Project
3. Importing XML Data
4. Data Munging
5. Reconciliation
6. Mapping & Exporting RDF
7. **Wrap-up <==**

Links + Contact

cmh329@cornell.edu

http://openrefine.org/

http://github.com/openrefine/openrefine

@openrefine, @cm_harlow