

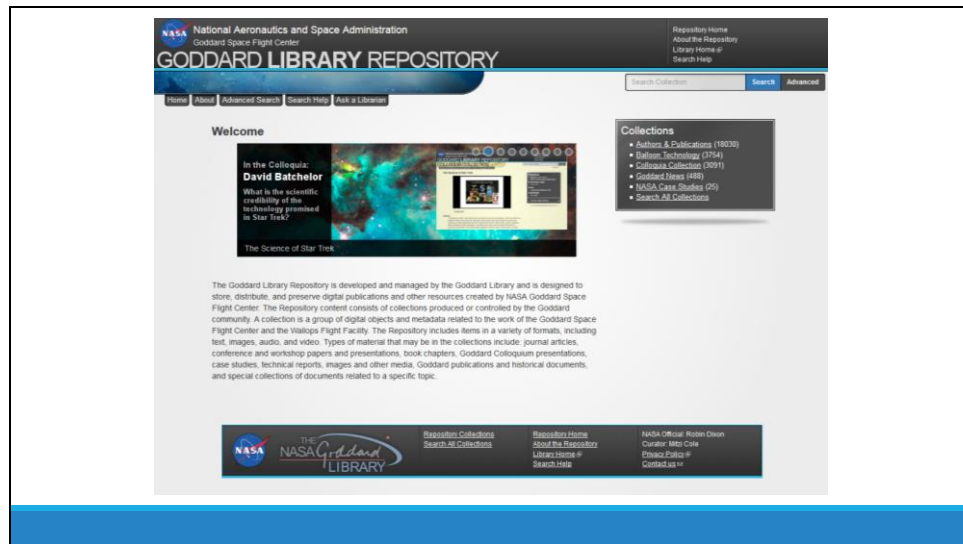


Pull it Together!

EXPORTING, RECONCILING, AND BATCH UPDATING AUTHOR DATA IN
FEDORA 3 & 4

Good morning. Name is Ruth Kitchin Tillman. I'm the Metadata Librarian at NASA Goddard, where I work with our Fedora Institutional Repository. Today I'm going to be talking about a project I did to export, augment, and batch update data from the repository.

Slide 2



Just a quick overview of the repository for those of you who are new. The Goddard Library Repository was launched to the public in February of 2011. It provides a central location for collection work produced by the Goddard Space Flight Center. It has 5 collections: Authors & Publications, Colloquia, Case Studies, Balloon Technology, and a newly-added collection for scanned issues of the Goddard News. The repository runs on a custom Fedora 3.3/Drupal 7 interface.

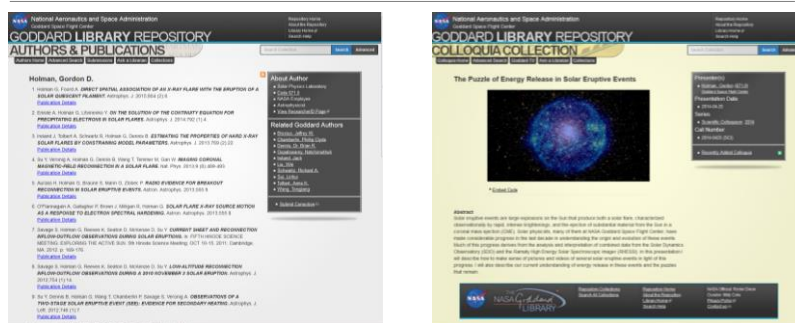
Challenge



The focus of this talk will be on reusing our own data to augment other data. However, steps 2-4 of the process could be done using data from an external source as well. Let's start with my particular challenge. The five collections in our repository, all have different purposes and come from different sources. Each is its own silo of information and relationships. However, two of the collections could augment each other, if only we could bring together the data.

Those two collections are our Authors & Publications collection and our Colloquia collection.

Tale of Two Silos: Authors and Colloquia



On the left is the repository's author page for Gordon D. Holman, including his organizational code/position/title, his Goddard co-authors, and a list of his publications. On the right is a colloquia Holman delivered in 2014. In its sidebar, we get a link to a textual search of colloquia for any other presentations under his and a similar search for code 671. These two do not meet. One cannot move from an author's colloquia to their publications nor from publications to colloquia. Searching for other colloquia also relies on a textual search, which may be a problem depending on the type of data we're given. So what are the collections and why were they built this way?

In brief, Authors and Publications records the publication output of all Goddard-affiliated authors. We harvest publication records from 4 indexing services, crosswalk them into PUB_MD format, attach Open Access PDFs when possible, and create local authority files for each author using extended MADS. The collection currently dates back to 2004 and will not go back beyond 2000.

The Colloquia collection, on the other hand, provides information about and access to colloquia delivered *at* Goddard by a variety of speakers and scientists. It goes back much farther than Authors & Publications, including information about presentations delivered as far back as the 1970s, with newer videos online and physical copies of older ones available in the library. The metadata is recorded in GEMS, based on whatever information was recorded by the colloquia's organizer. When the speaker is from Goddard, we work to get their organizational code, but haven't done any further authority control.

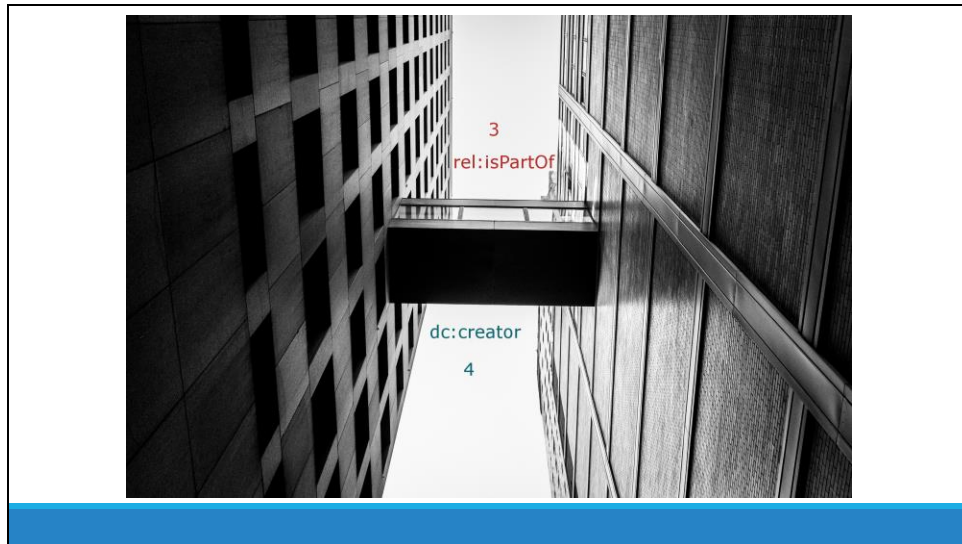
As you see, the collections have differ greatly in purposes and sources. Add to this that less than 20% of colloquia are delivered by Goddard authors and you'll understand why the two weren't created to interrelate.

Slide 5



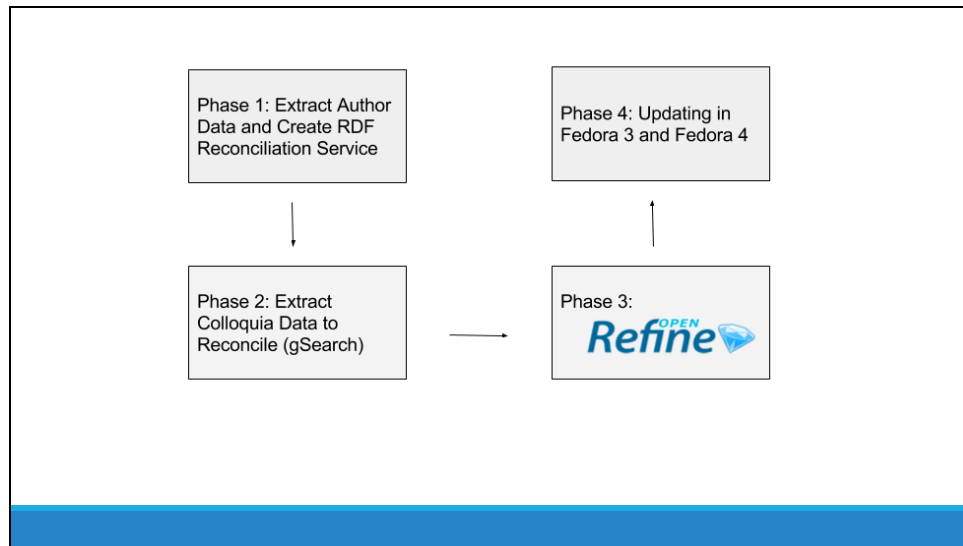
Still, we asked, what would it take to create relationships between the two collections? As we re-envision the site in terms of Fedora 4, as we think more about RDF and relationships between objects, can we bring these two together?

Slide 6



Based on our current models in Fedora 3 and our tentative remodeling for Fedora 4, we'd need to be update the record with either a RELS-EXT statement that Colloquia rel:isPartOf AuthorPID for Fedora 3 or a simple dc:creator relationship to the speaker's object record in Fedora 4.

Slide 7



The process took four stages: Using the author data to create a reconciliation service, extracting the colloquia data to be reconciled, performing the reconciliation in OpenRefine, and finally updating the colloquia in both Fedora 3 and Fedora 4 test instances.

Phase 1: Create a Reconciliation Service

Tools:

- <#ri>
- Sublime Text & Regex
- OpenRefine
- GRefine RDF Extension (DERI)

First, I had to come up with a way to match names from the Author collection with names in the Colloquia collection. To do this, I created a very basic reconciliation service with one or two versions of the author's name (depending on whether they used middle initials in their publications) and the URL of the author's page. This took a few steps and used the tools above, but was much less painful than it initially sounds.

Phase 1: Create a Reconciliation Service

- Use <#ri> query to export names and PIDs to CSV

- Sample:

"Atlas, David",gsfcirGaca:823462024

"Tilton, James C.",gsfcirGaca:411735411

"Esaias, Wayne E.",gsfcirGaca:868841903

"Lyon, Richard G.",gsfcirGaca:982601626

"Schnase, John L.",gsfcirGaca:580647414

"Esper, Jaime",gsfcirGaca:564921526

We can't query our MADS files through the resource index, but fortunately we store the most authoritative versions of author names as Last, First Middle in a Dublin core file, which I could access. I queried the authors collection and pulled out the authoritative name and the author's PID into a CSV.

Phase 1: Create a Reconciliation Service

Regex

Find: ("(.+?),(\sDr.\s|\s{2}|\s))(.+?)(\s(\w\.)"|"|"),gsfcirGaca:(\d{9})

Replace: <http://gsfcir.gsfc.nasa.gov/authors/id/\7> a foaf:Person;\n foaf:name "\2, \4", "\2, \4 \6".\n\n

Find: foaf:name "(.+?)", "\1 "

Replace: foaf:name "\1".

I could have used OpenRefine's functions to create an RDF file with author information. However, I needed to edit the author information with regular expressions to add a plain Lastname, Firstname in addition to Last, First Middle and decided to make the file as a part of that process. First query looks a bit like gibberish, but it sorts out the possible ways names might appear and captures them individually to be returned in the replacement. However, some people didn't have middle initials at all, which would make their record have two identical names, one with a space on the end—not a problem for the project, but also not good data—so I cleaned it up with the second Find and Replace.

I did all the regular expressions work in this project in Sublime Text, which has excellent Regex support.

Phase 1: Create a Reconciliation Service

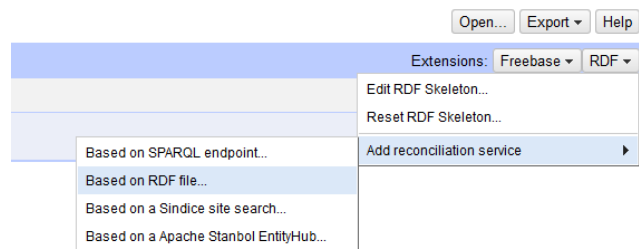
Result:

```
<http://gsfcir.gsfc.nasa.gov/authors/id/823462024> a foaf:Person;  
  foaf:name "Atlas, David".  
<http://gsfcir.gsfc.nasa.gov/authors/id/411735411> a foaf:Person;  
  foaf:name "Tilton, James", "Tilton, James C.".   
<http://gsfcir.gsfc.nasa.gov/authors/id/868841903> a foaf:Person;  
  foaf:name "Esaias, Wayne", "Esaias, Wayne E.".
```

Here's what that apparent-gibberish on the previous screen returns. A URL to the author's page, an RDF type of foaf:Person (which makes OpenRefine happier than no rdf:type), and one or two versions of their name. I chose FOAF because, as you'll see later, it's one of the default vocabularies supported by the tool I would be using. I added the necessary PREFIX information above the names and saved it as a Turtle/RDF file.

I used URLs instead of PIDs here for reasons I'll explain during the reconciliation process.

Phase 1: Create a Reconciliation Service



I had already added the DERI-RDF plugin to my OpenRefine instance. From the RDF drop-down, I selected “Add reconciliation service” and “based on an RDF file.”

Phase 1: Create a Reconciliation Service

The screenshot shows a dialog box titled "Add file-based reconciliation service". It contains the following fields and options:

- Name:** A text field containing "Oxford Authors".
- File details:**
 - ☐ Load file from URL: A text field.
 - ☒ Upload file: A "Browse..." button.
 - File format:** A dropdown menu set to "Auto".
- Label properties:**

Select properties that are used to label resources in the RDF data. These properties will be used to match resources.

<input checked="" type="checkbox"/> rdf:label	<input type="checkbox"/> skos:prefLabel	<input type="checkbox"/> dc:source
<input type="checkbox"/> dc:title	<input checked="" type="checkbox"/> foaf:name	
<input type="checkbox"/> Other...		
- Buttons:** "OK" and "Cancel" at the bottom.

Adding a file-based reconciliation service with the DERI plugin is quite easy. I simply had to give it a name, upload the file, and select label properties that the service should look for. As you see, foaf:name is one of the default label properties.

Now OpenRefine is set up to use the Authors collection names and URIs for reconciliation, but what about the colloquia data?

Phase 2: Extract Data to Reconcile

- Affiliation saved in local GEMS XML, can't be accessed via <#ri>
- Affiliation indexed in gsearch.
- Raw gsearch on affiliation variations: "Goddard", "GSFC"
- XSLT into CSV for OpenRefine

Extracting colloquia data proved to be trickier than using the resource index. Since we have over 5,000 colloquia and fewer than 20% have Goddard authors, I wanted to limit my dataset to colloquia where at least one Author came from Goddard.

Fortunately, our affiliation data was indexed in gSearch. So I performed a raw gSearch to return XML results for two affiliation variations—any affiliation containing Goddard or GSFC.

I used XSLT to crosswalk only the important data into a CSV, which could then be imported into OpenRefine.

Phase 2: Extract Data to Reconcile

```
- <object score="15.432882" no="98">
  <field name="PID">gsfcirCollq:477</field>
  <field name="accessRights">gsfc</field>
  <field name="affiliation">|Goddard!| Space Flight Center</field>
  <field name="browse">T</field>
  <field name="callNumber">1988-1021 (SCI)</field>
  <field name="creator">Suarez, Max</field>
  <field name="date">1988</field>
  <field name="dc.creator">Suarez, Max</field>
  <field name="dc.date">1988-10-21</field>
  ...
```

This is a sample of the gSearch result data. As you see, the affiliation search highlights Goddard even when it's only a part of the field. That way I could be sure I was getting results for any variations that existed such as NASA Goddard, NASA Goddard Space Flight Center, and plain old Goddard.

Phase 2: Extract Data to Reconcile

Extracted, for each AUTHOR (not each Colloquia):

- PID
- Name
- Date
- Title

"gsfcirCollq:5088","McElwain, Michael","10/8/2014","See the Ball, Be the Ball: How to Find and Characterize Planets Beyond our Solar System"

While the results I returned were for colloquia records, I was able to write an XSL Transformation that focused on extracting data for each creator mentioned in the record.

For each *speaker* in a colloquia, my resulting CSV had a line with the object PID, the speaker's name (note the same Lastname, Firstname format), and the date and title of the presentation in case they proved helpful in determining whether or not something was a match.

I then imported the CSV into OpenRefine.

Phase 3: Reconcile Data in OpenRefine

- Reconcile CSV against Phase 1 RDF file in 2 steps:
 - automatic matching
 - manual matching
- Derive PID for reconciled names

The RDF reconciliation service is remarkably straightforward. I set it running on the data, took a walk around the building, and came back to 180 matches. I then went through the rest of the results and selected from possible matches where the names didn't quite line up. Once that was done, I used Google Refine Expression Language to extract the author URI into a new column and derived the PID.

Phase 3: Reconcile Data in OpenRefine

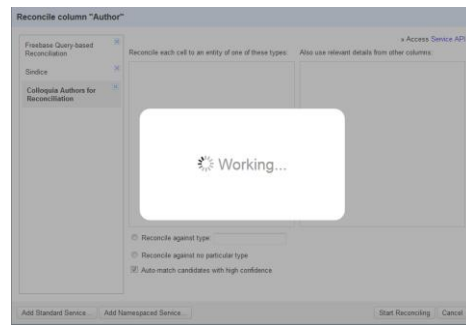
710 rows

Show as: rows records Show: 5 10 25 50 rows

	All	PID	Author	Date	Title
1.	gatorColq51066	Facet		4/15/2015	Eta Carinae: Astrophysical Laboratory for the Study of the Most M
2.	gatorColq51062	Text filter		4/1/2015	Coronal Mass Ejections: from a Novelty to a Natural Hazard
3.	gatorColq51091	Edit cells		3/6/2015	The Interplanetary Transport Network
4.	gatorColq51054	Edit column		3/4/2015	Let it Rain and Snow: A Year of Global Precipitation Measurement
5.	gatorColq51019	Transpose		12/15/2014	Risk Classification and Risk-based Safety and Mission Assurance
6.	gatorColq51001		anagement	12/1/2014	Supervisor All Hands
7.	gatorColq51013	Sort...		12/1/2014	Liquid Natural Gas in the United States: A History
8.	gatorColq51012	View		11/17/2014	Nuclear Rockets for the Future
9.	gatorColq5008	Reconcile			Start reconciling... je Ball, Be the Ball: How to Find and Characterize Planets Be
10.	gatorColq5008	Reconcile			Reconcile test in this column with topics on Freebase
11.	gatorColq51017	Facets			Reconcile test in this column with topics on Freebase
12.	gatorColq51009	QA facets			ated Organics in a Gale Crater Mudstone and the Imprint of
13.	gatorColq50076	Actions			Cube High-Performance Hybrid Data Space Processing Syst
14.	gatorColq50078				GSFC Containerized Computing
15.	gatorColq50080	Copy reconciliation data...			view of JPSS GRAVITE system
16.	gatorColq50082	Discover related RDF data			Find Planetary Systems through Citizen Scie

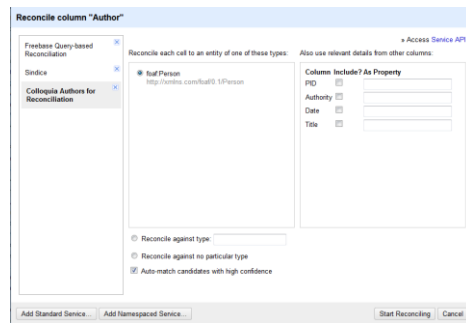
To reconcile on a column of data, simply select Reconcile from the drop-down menu and click Start reconciling.

Phase 3: Reconcile Data in OpenRefine



The reconciliation popup shows services active, including the Goddard one and you just need to click to get it started querying the service. Even a pre-loaded set of triples like mine has to be re-parsed by the service in the context of the data you're using before it offers options for reconciling.

Phase 3: Reconcile Data in OpenRefine



Since the only `rdf:type` of data was `foaf:Person`, it selects that as the default type against which you can reconcile. You could also specify another type or choose to reconcile against no particular type if you had multiple types. Sadly, the side column which lets you include additional data is not currently working for services with the DERI RDF plugin. In some cases, including another project of mine, it could be very useful.

Phase 3: Reconcile Data in OpenRefine

PID	Author
gsfcrColq:5166	Gull, Theodore Choose new match
gsfcrColq:5162	Gopalswamy, Nat Choose new match ✓ Gopalswamy, Nat (0.825) ✓ Create new topic Search for match
gsfcrColq:5191	Folta, Dave ✓ Folta, David (0.833) ✓ Create new topic Search for match
gsfcrColq:5154	Skofronick-Jackson, Gail ✓ Jackson, Gail Skofronick (0.125) ✓ Jackson, Clifton (0.022) ✓ Create new topic Search for match
gsfcrColq:5119	Leitner, Jesse Choose new match

Here's an example of matches after the basic reconciliation has been run. The green line shows the percentage of matches found. Each of the others is followed by one or two links along with their matching confidence factor. This is where it's helpful to have URLs instead of PIDs as a person's identifier. While Nat Gopalswamy and Gail Skofronick-Jackson are clear matches, it's helpful to click David Folta's link and see that yes, he works on flight dynamics analysis and thus is likely to be talking about the interplanetary transport network (title cut off in this image).

Phase 3: Reconcile Data in OpenRefine

PID	Author
gsfcrCollq:5166	Gull, Theodore Choose new match
gsfcrCollq:5162	Gopalswamy, Natchimuthuk Choose new match
gsfcrCollq:5191	Folta, David Choose new match
gsfcrCollq:5154	Jackson, Gail Skofronick Choose new match
gsfcrCollq:5119	Leitner, Jesse Choose new match

I quickly reconcile the data. Once I finished, I had the data to update over 250 colloquia. But next, I need to get it out of OpenRefine.

Phase 3: Reconcile Data in OpenRefine

Author	Date	Title
Facet		
Text filter	4/15/2015	Eta Carinae: Astrophysical Laborator
Edit cells	4/1/2015	Coronal Mass Ejections: from a Nove
Edit column	Split into several columns...	planetary Transport Networ
Transpose	Add column based on this column...	in and Snow: A Year of Glot
Sort...	Add column by fetching URLs...	assification and Risk-based S
View	Add columns from Freebase ...	for All Hands
Reconcile	Rename this column	atural Gas in the United State
Search for match	Remove this column	
Hrabar, John (0.53)	Move column to beginning	
Aspel, John (0.5)	Move column to end	
Gipson, John (0.5)	Move column left	
Create new topic	Move column right	
Search for match		
Eberstein, Igor (0.4)		Rockets for the Future
Alekhov, Igor (0.4)		
Chernobyl disaster (0.4)		

To retrieve reconciliation data, you must pull it into another column. From the column dropdown, choose to add a column based on this column.

Phase 3: Reconcile Data in OpenRefine

Add column based on column Author

New column name:

On error: ☒ set to blank ☐ store error ☐ copy value from original column

Expression: Language: Google Refine Expression Language (GREL)

No syntax error.

Preview History Starred Help

row	value	cell.recon.match.id
1.	Gull, Theodore	http://gsfcir.gsfc.nasa.gov/authors/id/128120310
2.	Gopalswamy, Nat	http://gsfcir.gsfc.nasa.gov/authors/id/293646633

Then, using GREL, you can extract the cell reconciliation match id into your new column. The result is in the preview, a name in the original column and a second column with the URI.

Phase 3: Reconcile Data in OpenRefine

- Replaced URL prefix with Author PID prefix.
- <http://gsfcir.gsfc.nasa.gov/authors/id/128120310> becomes
gsfcirGaca:128120310
- Exported as spreadsheet.
- Removed all columns except Colloquia PID and Author PID

Since I'll need the Author PID, I choose to do a quick find & replace on the column using the author PID prefix. I then exported it as a spreadsheet and removed the author name, colloquia title, and date, leaving only the Colloquia PID and the Author PID.

Phase 4: Updating Colloquia Objects

Challenges:

- Fedora 3 requires complete rewrite of any updated file
- Fedora 4 allows partial updates, but using Postman requires object-by-object updates

Getting data out is easy. Reconciling data isn't even that hard. Getting the data back in? This is where it gets tricky. Doing it manually might be a last resort for 250 objects, but for larger projects it would be prohibitive.

One of the first things I did after starting the project was look into how one could do batch updates in Fedora 3. I'd seen an option in the client, but had never experimented with the language. Fedora 3 uses Fedora Batch Modify files, which allow you to create, delete, or update any datastream within any object. The catch? When updating a datastream, you must replace the file with a new one. So if I wanted to add a new `rel:isPartOf`, I'd have to completely recreate the RELS-EXT.

Since we may not do the project until after migration to Fedora 4 and I wanted to examine the differences between the two, I also looked into what it would take to update the objects in Fedora 4. Fortunately, SPARQL-update is a great language for updating objects, but the Postman interface I'd been using would be *more* time consuming than a Fedora Batch Modify file because I'd have to do it one-by-one.

Phase 4a: Updating Fedora 3

- Language: Fedora Batch Modify XML
- Minimal wrapper but complete datastream contents
- All RELS-EXT data can be extracted through the <#ri>
- Broke down into steps:
 - What is the same across Colloquia RELS-EXT?
 - What is different?
 - What will the new info be?

Let's look first at Fedora 3. What is Fedora Batch Modify language? It's a very minimal wrapper which includes PID, datastream ID, and replacement datastream title enclosing a complete XML file.

So in order to do the update, I was going to have to recreate the RELS-EXT for every single colloquia I wanted to update. Since RELS-EXT data can all be extracted through the resource index, I broke it down into steps to determine what I'd need to recreate a file. What is the same across all Colloquia RELS-EXT files? What can just be part of my template? Next, what is different? What am I going to have to get out of the current system? And finally, what does the new info I'm adding look like?

Phase 4a: Updating Fedora 3

- Same:
 - Collection membership – rel:isMemberOfCollection
 - Models – fedora-model:hasModel
- Different:
 - Subcollection membership – rel:isMemberOf
- New:
 - Author relationship – rel:isPartOf

Fortunately, our RELS-EXT for colloquia are pretty straightforward. All colloquia belong to the colloquia collection. All use the same two Fedora models.

What's different between colloquia is that we subdivide our colloquia by the hosting organization—engineering, science, systems, etc. Of course, some co-host, which means a colloquia may have several subcollection memberships. We use rel:isMemberOf here instead of isMemberOfCollection to avoid system confusion.

And finally, the new data is the author PIDs using the rel:isPartOf.

Phase 4a: Updating Fedora 3

- Extracted Colloquia PIDs and Subcollection PIDs from <#ri>
- Combined in Excel with Colloquia PIDs and Author PID.
- Result file:

Colloquia PID	Subcollection PID	Author PID
---------------	-------------------	------------

- One line per relationship. No order.

So, I extracted all Colloquia PIDs and their respective Subcollection PIDs from the resource index. I added a new column in the Excel file which already had Colloquia and Author PIDs (bear with me, I know it sounds a little odd) and put them in.

In the resulting file, each line had a single relationship, a Colloquia PID and either a Subcollection PID or an Author PID. A colloquia with 2 subcollections and 3 authors would have 5 rows. Most just had one row with a subcollection, since I'd exported subcollection info for every single colloquia as it was much faster than extracting based on several hundred PIDs. Obviously, it needed more work done to get it usable.

First, I ordered the rows on the Colloquia PID, so that the information for each object would be adjacent. This was critical for the next steps, after I imported it into OpenRefine.

Phase 4a: Updating Fedora 3

2719.	gsfcirCollq:5056	subcollection:5	
2720.	gsfcirCollq:5056		gsfcirGaca:196599745
2721.	gsfcirCollq:5056		gsfcirGaca:530890245
2722.	gsfcirCollq:5056		gsfcirGaca:939436726

Here's an example of what the data looked like in OpenRefine for a colloquia with one subcollection and three authors.

Phase 4a: Updating Fedora 3

Object	Collection	Author
Facet	lection:2	
Text filter	lection:2	
lection:2		
Edit cells	Transform...	
Edit column	Common transforms	
Transpose	Fill down	
Sort...	Blank down	
View	Split multi-valued cells...	
Reconcile	Join multi-valued cells...	
gsfcrCollq:5049	subco	Cluster and edit...

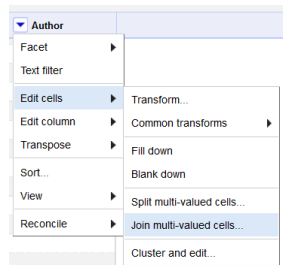
I used the “Blank Down” feature, which removes duplicate data in *subsequent* (and only adjacent) rows...

Phase 4a: Updating Fedora 3

gsfcirCollq:5056	subcollection:5	
		gsfcirGaca:196599745
		gsfcirGaca:530890245
		gsfcirGaca:939436726

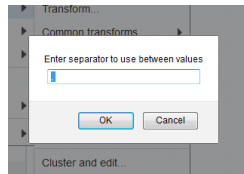
To end up with this. Now OpenRefine allows you to toggle at the top from viewing this as 4 rows to viewing it as one “record.”

Phase 4a: Updating Fedora 3



Since OpenRefine now views it as one record with several rows, I can then manipulate the data further. The join multi-valued cells function combines multiple rows in a record which all fall into the same column.

Phase 4a: Updating Fedora 3



I just choose a separator, in this case the default comma...

Phase 4a: Updating Fedora 3

gsfcirCollq:5056	subcollection:5	gsfcirGaca:196599745, gsfcirGaca:530890245, gsfcirGaca:939436726
------------------	-----------------	--

And here we go. I now have all my data on one line.

Phase 4a: Updating Fedora 3

Result:

- Tab-separated rows of comma-separated PIDs.

Transforming:

- Basic template for finished RELS-EXT
- Regular Expression using capture and backreferences, to populate template

This resulting data is tab-separated with sibling PID-types comma-separated, which will matter for later manipulation with Regular expressions. I exported it back into Excel, where I sorted again and removed all rows which didn't have anything in the authors column.

Now I just needed to get it into RELS-EXT. My thoughts first went to scripting or even XSLT, but just as I found when creating the authors RDF file, I quickly realized that the simplest method would be to use regular expressions. I created a template for the RELS-EXT file with variations to allow for up to three subcollections and up to four authors (both the maximum number my visual scans had observed).

I then wrote the surprisingly-simple regular expression and variations thereof necessary to capture the PIDs and populate the template using backreferences.

Phase 4a: Updating Fedora 3

Sample line:

```
gsfcirCollq:5056 subcollection:5 gsfcirGaca:196599745,  
gsfcirGaca:530890245, gsfcirGaca:939436726
```

Sample Matching Regex:

```
(gsfcirCollq:\d{1,4})\t(subcollection:\d)\t(gsfcirGaca:\d{9}),  
(gsfcirGaca:\d{9}), (gsfcirGaca:\d{9})
```

Here's the example from previous images, one Colloquia PID, one Subcollection PID, and three Author PIDs. This variation on the regular expression looks for this combo of PIDs and wraps each in parentheses to be returned later.

Phase 4a: Updating Fedora 3

```
<fbm:modifyDatastream pid="\1" dsID="RELS-EXT" dsControlGroupType="X" dsLabel="colloquia_RELS-EXT_ds"
logMessage="BatchModify - modifyDatastream">

<fbm:xmlData>

<rdf:RDF xmlns:rel="info:fedora/fedora-system:def/relations-external#" xmlns:fedora-model="info:fedora/fedora-
system:def/model#" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">

  <rdf:Description rdf:about="\1">

    <rel:isMemberOfCollection rdf:resource="info:fedora/collection:3"/>
    <rel:isMemberOf rdf:resource="\2"/>

    <fedora-model:hasModel rdf:resource="info:fedora/cmodel:2"/>
    <fedora-model:hasModel rdf:resource="info:fedora/cmodel:4"/>
    <rel:isPartOf rdf:resource="\3"/>

  </rdf:Description>

</rdf:RDF>
</fbm:xmlData>
</fbm:modifyDatastream>
```

This is the overall file, but it's a bit hard to read on the screen, so let me pare down to the RDF Description section.

Phase 4a: Updating Fedora 3

```
<rdf:Description rdf:about="\1">
  <rel:isMemberOfCollection
rdf:resource="info:fedora/collection:3"/>
  <rel:isMemberOf rdf:resource="\2"/>
  <fedora-model:hasModel rdf:resource="info:fedora/cmodel:2"/>
  <fedora-model:hasModel rdf:resource="info:fedora/cmodel:4"/>
  <rel:isPartOf rdf:resource="\3"/>
  ...
</rdf:Description>
```

As you see, in `rdf:about`, I have a backreference, backslash 1. This refers to the Colloquia Object PID capture. I use that backreference in the larger file as well. Backreference 2 occurs within a `rel:isMemberOf` statement for the Subcollection PID and backreference 3 occurs within a `rel:isPartOf` statement for the first Author PID. Backreferences 4 and 5 would be duplicates of that `isPartOf` statement.

After I'd run through the variations, I added Fedora Batch Modify XML headers and footers and checked the document's XML validity to confirm that I hadn't missed a line of data. But my calculations had been correct and the document was good to go. I ran it against one of our sandboxes and then ran queries to confirm that it functioned as expected. The site didn't yet support display, but I could use the same queries that returned an author's publications to now return their colloquia as well.

On to Fedora 4!

Phase 4b: Updating Fedora 4

- Simply adding dc:creator with Fedora Object URI (not PID)
- Initial test using Postman
- Postman too time-consuming
- Used Postman to generate sample CURL
- Updated using single CURL file with multiple PATCHes

As I'd mentioned previously, what we needed to do in Fedora 4 would be far simpler. I simply needed to send a SPARQL-update with additional dc:creator information and the author's Fedora object URI.

I did several updates using Postman with complete success, but quickly realized it would be far too time-consuming. So I investigated alternatives. Postman helpfully let me export a CURL of one of my queries and I was soon up and running with the same Regular Expression method to batch-generate PATCH updates in one giant CURL file.

Phase 4b: Updating Fedora 4

Sample data:

gsfcirCollq:1430	gsfcirGaca:160862315
gsfcirCollq:1471	gsfcirGaca:160862315
gsfcirCollq:2087	gsfcirGaca:290886104
gsfcirCollq:3608	gsfcirGaca:166053105

No subcollections here and no need to combine all updates for an object into a single statement. I could PATCH a colloquia as many times as I wanted, which made the process very simple. I took my original OpenRefine export...

Phase 4b: Updating Fedora 4

Find: gsfcirCollq:\d{4})\tgsfcirGaca:\d{9})

Replace:

```
curl -X PATCH -H "Content-Type: application/sparql-update" -H "Cache-Control: no-cache" -d 'PREFIX dc: <http://purl.org/dc/elements/1.1/>
```

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```

```
DELETE { }
```

```
INSERT {
```

```
  <> dc:creator <http://localhost:8080/fedora/rest/authors/\2>. }
```

```
WHERE { } 'http://localhost:8080/fedora/rest/colloquia/\1'\n
```

...and ran this find and replace to generate the PATCH. You may notice that instead of using PIDs, I'm using the Fedora object URLs. We're still figuring out how we want to transition PIDs, but my current model has a similar structure to our front-end URL model. Thus, I captured the numeric portion of the PID and concatenated it with the collection's base internal URL.

Phase 4b: Updating Fedora 4

Advantages to Fedora 4 Method:

- Lower risk: update not replacement
- Less information: no need for subcollections or grouping together
- Simple update: CURL runs from command line

Unsurprisingly, I found the Fedora 4 method much simpler to work with, assuming we will be able to derive the URLs of our final repository structure from the current PIDs. It was much lower-risk and for a collection with a more complicated RELS-EXT, we would definitely have to wait until Fedora 4 to do this kind of project.

Because there was less information to update, there were far fewer steps.

And finally, CURLs are simply easier to run.

Next steps

- Timeline uncertain on implementation
- Need to decide whether to add author records for colloquia-only authors
- Data ready for whenever we need it

All in all, I was surprised by how little this project took—3 days total with less than half of each spent on the project. I did a great deal of intellectual groundwork beforehand by working with OpenRefine or regular expressions and thanks to Christina Harlow's excellent Code4Lib DMV workshop this summer which made me decide the project was feasible to begin with.

Our timeline on implementation is uncertain, just as is our Fedora 4 migration timeline. While I could add this data now, our Drupal modules won't search for and thus won't display it. One major decision we'll need to make is whether we're only going to record colloquia author information for authors who also have publications in our repository or whether we'll add new records for Goddard employees who've never published but delivered colloquia.

Since I explored both Fedora 3 and Fedora 4 options, however, the data is ready for whenever we decide to move forward on implementation.

Thanks

Christina Harlow - @cm_harlow for OpenRefine tutorials and write-ups.

Kate Deibel - @metageeky & Ben Armintor - @BArmintor for listening as I worked through roadblocks in the process.

I'd like to close with especial thanks to Christina Harlow, whom many of you may know, for her OpenRefine workshop and her tutorials. Without her work, this would've taken much longer and I wouldn't have had nearly such a good footing to start. I also greatly appreciate Kate Deibel and Ben Armintor, who helped me talk through a few issues I encountered.

Tools Used:

OpenRefine: <https://github.com/OpenRefine> or <http://openrefine.org/>

DERI RDF Refine plugin for OpenRefine: <http://refine.deri.ie/>

Sublime Text (trial version available): <http://www.sublimetext.com/>

cURL: <http://curl.haxx.se/>

Postman (Chrome only): <https://www.getpostman.com/>

Code & Contact

Code:

Presentation repository: <https://github.com/ruthillman/local-reconciliation-project>

Christina Harlow's C4L Presentation: <https://github.com/cmh2166/c4IMDCpres>

Contact Me:

ruthillman@gmail.com / ruth.k.tillman@nasa.gov

@ruthbrarian

I've put these slides and a selection of code snippets used in the project in a repository which you can visit here. I'm also happy to share that link afterward. You can contact me at ruthillman@gmail.com or ruth.k.tillman@nasa.gov

Attributions

Image 1: “Silos” by Anthony Arrigo, Flickr. <https://www.flickr.com/photos/anthonyfarrigo/>

Image 2/3: “Skybridge” by Billie Ward, Flickr.
<https://www.flickr.com/photos/wwward0/21180929605/>