IT’S 2019

DO YOU KNOW WHERE YOUR METADATA IS?

Oregon Digital and the Joys of Migrating...Again

Cara Key, Oregon State University
Greg Luis Ramirez, Oregon State University
Linda Sato, University of Oregon
Ryan Wick, Oregon State University
About the repository - https://oregondigital.org
Decision to migrate/rebuild

- Not if but when/how
- Current stack is aging
- Balance work and resources needed with other projects
- How many features are in Hyrax now vs. what we want

Image: PH037_b079_MF00860 Angelus Studio Photographs. Special Collections & University Archives, University of Oregon Libraries. [https://oregondigital.org/catalog/oregondigital:df7f2z15p](https://oregondigital.org/catalog/oregondigital:df7f2z15p)
Project goals

- Rebuild Oregon Digital starting with latest Hyrax
- Less custom code, more manageable local customizations
- Better compound object support (yay PCDM)
- More metadata enhancements, better context (esp. archival items)
- Better and fewer media viewers
- Build a foundation for Spotlight exhibits
- Clearer display of Rights and License information, encourage re-use
- User Collections, ‘My Shelf’
- Looking ahead to Valkyrie implemented in Hyrax
Project management

- Continue OSU and UO partnership
- Four technical teams: Features, Metadata, Infrastructure, Migration Tooling
- Product Owners and Service Managers determine requirements and priorities
- Workcycles are 6 weeks long, broken into 2-week sprints
- GitHub, Slack, Google Drive
Zooming in on migrating assets

- Since we’re not upgrading-in-place: external migration
- Want as short a time of ‘new content freeze’ as possible
- Metadata Team and Migration Team working closely together
- Metadata updates covered first
- Second, introducing Hyrax-migrator, our migration gem
Metadata
Overview of metadata changes

- Creating an updated Metadata Application Profile (MAP) was a very early priority
- This effort was followed by crosswalking and remediation of existing OD metadata to fit new MAP
  - Updates to predicates, vocabularies, allowed values & data types
  - Cleanup of legacy problems
- Emphasis on where metadata impacts user experience, for example:
  - Streamlined facets for better browsing
  - Using metadata values to manage access restrictions
Metadata strategy

Step 1: Update MAP

Step 2: Update existing metadata to align with MAP

Step 3: Migrate perfect metadata to new system

Step 4: Party!
Metadata remediation

- We are doing the bulk of our metadata remediation in the current (old) Oregon Digital - before migration of the assets
  - Not fundamentally changing the nature of the metadata
  - Allows concurrent development and metadata work
  - Less complication within the migration process, vs remediating as part of migration pipeline
  - More “mature” product at delivery, vs remediating after migration
- Most changes happen in batches, either via export + CSV editing + scripted updates or directly with scripts
- Requires updating configuration of old system to accommodate changes
Metadata example: Rights statement update

- Changing all Rights values to RightsStatements.org URIs
  - Formerly a mix of OpaqueNamespace.org URIs and Creative Commons licenses
- Confirm all items have a Rights field value
- Map existing values to RightsStatement.org values
- Update OD1 to fetch & validate RightsStatement.org URIs
- Run batch processes to swap out URIs / move all Creative Commons URIs to License field
- Map RightsStatement.org values to proposed OD2 re-use categories for planned rights-based searching
Hyrax-Migrator Gem
Hyrax-Migrator: Design

Inspired by Hyrax Actor Stack

Main components:

1. **Work** stores filepath of uploaded files, metadata, migration status
2. **Actors** update the work and launch services
3. **Services** do the work and are specific to our organization
4. **Middleware** can be modified by writing new actors/services and adding them to the configuration
Hyrax-Migrator: Ingest

3t945r08v.zip
|--data
|----3t945r08v_content.jpeg
|----3t945r08v_descMetadata.nt
|----manifest-md5.txt
|--manifest-sha1.txt

work, config
create(work)
Hyrax-Migrator: Strategies / Implementation

Crosswalk:
- property: title
  predicate: http://purl.org/dc/terms/title
  multiple: true
  function:
- property: creator_attributes
  predicate: http://purl.org/dc/elements/1.1/creator
  multiple: true
  function: attributes_data
Hyrax-Migrator: Running the stack

```ruby
aasm do
  state :bag_validator_initial, initial: true
  state :bag_validator_succeeded, :bag_validator_failed
  event :bag_validator_initial do
    transitions from: %i[bag_validator_initial bag_validator_failed],
    to: :bag_validator_initial
  end
  event :bag_validator_failed, after: :post_fail do
    transitions from: :bag_validator_initial,
    to: :bag_validator_failed
  end
  event :bag_validator_succeeded, after: :post_success do
    transitions from: :bag_validator_initial,
    to: :bag_validator_succeeded
  end
end
```
Challenges
# Configurations

c = Hyrax::Migrator::Configuration.new

c.upload_storage_service = :file_system

c.ingest_storage_service = :file_system

c.file_system_path = "/data/tmp"

c.ingest_local_path = "/data/tmp"

c.skip_field_mode = true

c.migration_user = 'admin@example.org'

c.crosswalk_overrides_file = "tmp/crosswalk_overrides.yml"

# Ingest service

i = Hyrax::Migrator::Services::BagIngestService.new(['batch_demo'], c)

i.ingest
Hyrax-Migrator: Batch demo
Testing and Next Steps

Seed data - wide range of metadata and content types from both institutions

Testing locally and on staging

Validation

Finalize workflow

GUI

Image: Fruit Testing, Food Technology Department, Oregon State College, 1948. OSU Special Collections & Archives Research Center https://oregondigital.org/catalog/oregondigital:df70d726p
Did we really learn our lesson(s)?

Mostly. What are we doing different from last time?

- We have built up a lot of experience with our metadata/RDF/linked data, and how to do updates, so we’re doing nearly all of it before the migration begins.
- More development time spent on migration tooling.
- Hyrax-migrator is stricter about what predicates to map to.
- Compound objects are still complicated.
Questions?
Thank you!

https://oregondigital.org
https://github.com/oregondigital/hyrax-migrator