2020-11-19 Whitman College Migration Pilot Metadata Meeting

Attendees

- Alan Stanley
- Amy Blau
- Dana Bronson
- David Wilcox
- Paige Morfitt

Agenda

Topic
1. Implications for collection structure, site build (fields, forms) of the decisions we are making now 2. Establishing a timeline for providing Alan with mapping, fields, sample data, any additional documentation that will help to clarify our collection structure and expectations for searching, sorting, and faceting
1. Metadata fields a. At what point do we sit down with the fields we've made and set out what encoding or type (string, taxonomy, linked agent, etc) will be in I8? b. What will we need to do about creating taxonomies (and setting up linked agents) ahead of migration? (In particular for Contributor field) c. Will we be able to modify some of this in I8? For example, we have an entry in our Creator field that is encoded in our MODS as a person when it's really a corporate entity. The MODS won't come over exactly. We could make a taxonomy of corporate entities and one of people. How do we set it up in I8? Can more than one taxonomy be used in a given field? d. How would having multiple media types on the back end impact the work_type (and vice versa)?
1. Mapping a. Should we provide a list of Solr fields? b. Feedback? Timeline on feedback?
1. Metadata display a. Special characters: html or unicode in fields -- issues fixing in 7 or moving to 8? Issues in RDF? b. If we don't have html, is there a way to have several paragraphs in one field? c. How hard to remediate once we're in 8? (Workbench) d. We should be able to filter on collection name that is drawn from RELS-EXT rather than fielded metadata?
1. Collection structure a. We are currently making some changes to collection-level structure. Would it be easier to provide a chart of this structure and to tweak the hierarchy as part of the migration than to rearrange the hierarchy in 7 for the migration? If so, by when would you need the new structure chart? b. Blocks -- will these work as in I7?

Notes

1. Alan: still deciding on form of migration, we may use more than one strategy to document them. We can do clean-up as we go. There is a plug-in to the migrate module. Processes are piped. Can skip data, put in defaults, etc. If there are terms we've identified as identical, we can resolve there. There are bulk editing tools in Drupal. Creating properties, align headers with property names.
2. Alan: Vocabularies 8 uses authority records less, we can define roles can say if a term does not exist can ignore it or build it on the fly. Don't know if we can flag it for later use, for instance if you had a name that was misspelled. Also disambiguation. We can control that by vocabulary. You have the toption when you put a taxonomy term in, to set up an if then to create or include a name if it isn't already in a taxonomy. Could have two voc items that appear to be identical with different TIDs.
3. Alan: site presents as a hierarchy but it is actually really flat. Objects can be in multiple collections at the same time as well. Don't worry about that aspect of it. Structure is all done with taxonomy terms. Node ID is the only thing that isn't changeable.
4. Alan: two different definitions for vocabularies. List in Drupal vs external. In some cases import local copies, in others would have external lookup. Where possible when storing metadata as entity reference (exists in Drupal already, can be node, file, media, taxonomy term). Happier to have something internally controlled where possible. Put subject in save URL, can pull in. Calls can be expensive, lookups can take longer. Chance to have caching. Out of the box Drupal vocabularies have one field (name). A lot of flexibility. Migrate API can do test batches, roll back, redo.
5. Alan: would use an entity relationship for creator, links out to an entity, doesn't matter what. When setting up the interface, want to make sure that new content can be created.When you set up to draw from vocabulary, can set up to draw from a set of vocabularies.

6. Alan: SPARQL, Solr, CSV. Solr fields information. With MODS forms there is a place to enter the role. As long as it's consistent it can be parsed out.
7. Alan: encoding is a problem. Html not a problem in and of itself, can specify that for each field. Can embed html. Multiparagraph fields without html can be handled with line feeds. As long as we know which fields have html in them, we can set it up as a formatted field. In 8 you have a single node with as many fields as you need for information, repeated fields can be stuck in paragraphs. Grouped bits of data.
8. Alan: blocks as static text on a page vs. javascript pop-ups. In Drupal 8 configuration is different, but you can stick javascript on anything. Currently is twig templates which is like java expression language. Library (css + javascript) you attach to a node. Done by themers, not part of migration per se.
9. Alan: every datastream is pulled in, if it's in RELS-EXT, you can pull it out. Can also be from a triple store. Caveat, paged content has thousands of solr fields. Solr config page where you pick fields you want.
10. Alan: lots of plug-ins, skip on empty.
11. Alan: mapping looks complete, from programmatic point of view, as long as data is consistent should be fine, we'll need to look at sample records. Need to know which values will repeat and which ones won't. Can set up everything as a multiple but might be useful to have some things as unique.
12. Alan: in copyright statement would be javascript, with theme.
13. Workflow of testing and we will have feedback on sample data to make sure things look right.
14. Alan: grab object information, can write a filter to check on data.
15. Alan: hoping to throw some of the work back to the community. Typically start with easy stuff. HOCR hard because not editable.