MODS and RDF Call 2015-10-05

Time: 9am PDT / Noon EDT

Call-In Info: 712-775-7035 (Access Code: 960009)

Homework Reminder:

- Review name element usage and acceptance criteria: MODS Name Individual Institution Usage And RDF Conversion
- Think on problem issues (like order) to discuss.

Moderator: Steven Anderson (Boston Public Library)

Primary Notetaker: TBA (etherpad for roll call: https://etherpad.wikimedia.org/p/RDF-MODS-20151005)

Attendees:

- Steven Anderson (BPL)
- Eben English (BPL)
- Danny Pucci (BPL)
- Jennifer Young (Northwestern)
- Kate Gerrity (Amherst College)
- Sara Rubinow (NYPL)
- Shawn Averkamp (NYPL)
- Julie Hardesty (Indiana University)
- Rebecca Fraimow (WGBH)

Agenda:

1. Fedora 4 Test Code:
    a. Code base to try MODS title element in Fedora 4 -- hopefully will happen this week so people can take a look at how that's being represented within the next week or so.

2. Overview of MODS name mapping attempts
    a. Anybody want to go over their mapping attempt who wasn't on the call last time?
        i. Julie: missed last week's call, has an IU name mapping attempt to go over
            1. Basically just used Dublin Core terms for her example.
        ii. another option is to use MARC relator as URI instead of DC terms as Creator -- these are sub-properties of DC terms already, and are more precise than just Creator/Contributor
        iii. DC:Creator and DC: Contributor with a literal text string; this actually has to be a URI value, literal values are not allowed

            1. Can literals be used with MARC relators? No, as sub-properties of DC:Creator and Contributor, they inherit the same restrictions
            2. how do you solve this? make your own URI, create a blank node?
    b. Document that talks about DC terms elements -- will be added to the notes afterwards, good reference for what can and can't be used with DC properties
        i. https://code.google.com/p/tdwg-rdf/wiki/DublinCore#1._Use_of_Dublin_Core_terms_in_RDF

3. Favorite mapping that anyone wants to talk about that wasn't their own mapping?
    a. Steven: really liked the NYPL's idea of using a local copy of the prefLabel, can add additional properties and then if the linked data source ever goes down, can still display it in the UI relatively easily and always know that it has at least the prefLabel property.
        i. Version 2 of BPL's Document tries to implement this approach: https://docs.google.com/spreadsheets/d/1nNnGI-u9RazlFJ_cdDJPT6dpASjnqHwomGbpScDIATQ/edit?usp=sharing
        ii. Eben: doesn't love the idea of creating a local name for something that already has its own URI, BPL example where the name is something that has an LoC URI but the example uses a local value, which seems counter-intuitive -- why create local names for things that exist in other places? Get the concept of having the text string locally available, but it seems redundant. Can someone from New York talk more about their rationale?
            1. Shawn: Not everyone is going to have the same use case as at NYPL where there is a requirement to store local data about names, whether local or already have a URI; are in a position where they may need to be publishing a dataset for a curator. NYPL's use may be overkill for many implementations. If you still wanted to have a similar use case you could do blank nodes, but not sure how well that would work.
            2. Steven: lots of discussions on this over at BPL, if people need to add additional properties, concern is that when you start using linked data sources for locally minted names from other smaller organizations, higher probability that an end point may go down or that the institution may decide to no longer support that linked data set. By having a prefLabel, you have that for long-term preservation so you know what that URI was at one point and at least know what the name was.
            3. Eben: not that different from what we currently do in MODS where the string is stored as a text node and have the URI as it exists in the URI value attribute on that name, just wondering if other people felt the same way that it seemed redundant. Philosophical question: where do you draw the line in terms of what URIs you're willing to rely on vs. what URIs you make local copies of the prefLabel for? How far does one want to take that going forward?
            4. Julie: are you getting rid of your MODS once you've done this, or does the MODS XML hang around somewhere you have access to?
            5. Steven: this will replace MODS at BPL, maintaining both linked data and updated XML would be too much to handle.
            6. Julie: If you're dealing with both string and URI in XML, makes sense that you'd want to carry that forward. At Indiana, we identify if information is from a URI and where, but we don't store that information -- just keep the string value -- so coming from the literal route regardless.

a. Steven: You could use this approach for one where you don't have URI support, could still have a resolution that would still have the prefLabel without the SKOS exactMatch property. Could still use the data model, just wouldn't be able to link out to another authority.
b. Julie: Makes sense, since Indiana doesn't have that info.

b. Steven: if minting your own URI, something that's only relevant in your local area or etc., do you want additional properties that would be associated with that entry?
    i. For ex: storing FOAF name, which might be different from the prefLabel in that that could just be name part, name w/date, best representation of base name, whatever. To follow that up, if a local-created name doesn't have an external URI, is the minimum that it would be just a prefLabel or does it have other properties we could expect?
        1. This is especially as prefLabel will have to be updated from some kind of sidecar system every once in a while for external sources. So locally minted names could use a similar process to update their prefLabel where FOAF name could be canonical version of name and prefLabel could be generated from that and updated as properties change; if death date is updated, then index circuit could update prefLabel from FOAF name to have the correct death date at the time.
    ii. Question: are there other pieces of data besides birth and death date that get displayed with name? Terms of address, etc.?
        1. Answer: titles, sometimes occupations, there can be quite a bit more; sometimes it's dates when person was most prolific, could be a role for disambiguation purposes
    iii. Question again: If people are only interested in the way that the name is normally displayed, are people fine with that being the only representation of the name that they're storing, or if they want another representation that's the canonical name without its accompanying dates or terms of address, do people have a use case where those things would need to be separated?
        1. Danny: Name is supposed to be a thing on which information can be colocated, so breaking it up into component parts makes colocation harder. John Adams and John Quincy Adams w/out dates makes it hard to facet on, not really in favor of breaking it up. Birth and death dates broken out might be good for searching on those dates, but to this point, the way these headings have been used, making them smaller than the full heading is not useful.
            a. Steven: can just try what seems to be best when they start the collaboration document
    iv. Shawn: Do we need to make a recommendation on how we store the names behind the URI, or do we really need to just make a recommendation on having a URI/string/option for both?
        1. Steven: if we write a shared code base to translate from MODS XML into whatever MODS RDF ends up being, it's necessary to make some of these decisions for what it spits out. So mostly just in terms of collaborating coding efforts more than whether or not it's syntactically valid
        2. Remark: so much variation in how people are structuring name or using it, how general is this going to be?
        3. There's a lot of particular use cases, people are always still welcome to do the things that they want to keep doing, nothing is binding, but this is a long process and there is something to be gained from having some kind of consensus; still, "recommendation" has to be taken with as many grains of salt as possible based on use case.
        4. Julie: we're turning what we're doing into an actual transformation, can we look at this more as creating a base transformation for people to go from? Can this help us streamline what we're doing so we're not having to imagine all the cases and decide what cases are relevant, start with a baseline transformation, give options if time allows for additional features that could be transformed? Part of the recommendation, place where people can start and an aid.
            a. Steven: definitely a place where people can start, but also want to make sure it's something that works and makes sense in and of itself
            b. Julie: if name has multiple parts involved, just bringing them all together as a baseline and dealing with it as a single thing seems like a useful starting point
            c. Steven: Definitely possible, but do we want that to be the maximum amount of specificity that we want to do or should the transformation be intelligent enough to break that out as well? Maybe it's a difference between the simple and complex as to how much it breaks out and how much it retains? If original source records have data split out, do we retain that specificity or agree as a group that it will be lost?
            d. There's no upper limit on specificity that people can implement in their own institutions, but what's the minimum level of specificity that we're recommending or an out-of-the-box transformation would provide? If we're looking at the out-of-the-box use case then having the entire name as a SKOS prefLabel seems like a reasonable baseline.
                i. Simple version could just be the simple label and complex could have a few pieces of information broken out but otherwise would be the same
                ii. Julie: Complex could serve as an example if there's more information that people want to break out.

4. Other topics to discuss: what do you do about ordering? If you have multiple authors and you're trying to keep a consistent ordering because the first author would get upset if he wasn't listed first, anybody think of approaches for that?
    a. Steven: BPL's attempt at a solution (last example in our mapping document), if you have two authors, would have them still linked like normal, only difference is that you would have an additional predicate that is author # order with an RDF list that would represent the URIs that make up that order, so if you have an object that cares about order, you would have the property that would create a listing of what the order is, and if you don't care about order you don't need to worry about supporting that predicate.
    b. Question: Do you want to add that # order suffix onto a relator term or use something more simple like DC terms Creator? Lots of different relator terms, do you always use relator author to represent order, or would it be an easier/more general use case to use DC terms Creator to represent order?
        i. Steven: Adds complications, because some are subproperties of contributor instead of creator, also then you have to order all the objects rather than just author if you only care about author.
    c. Will you want to order different names with different rules in some cases? Wouldn't you want generic name predicates rather than a specific relator that corresponds to one but not all of the names?
        i. Danny: we're not an IR, we don't have a lot of books and papers with multiple authors where order becomes important, but lots of items with lots of different kinds of creators, and would want some generic way to retain order of creators rather than specifically for authors
    d. Julie: Anything in MODS allow this same sort of ordering?
        i. Steven: XML by definition retains hierarchy and structure within document, so a name that comes into an XML document first is by default the first name. RDF statements are inherently unordered and can be reordered within the data store, may come back 1st in the results first time and 15th next time.
        ii. Someone else: MODS also has ability to add usage attribute, usage=primary or something, but that is only used for primary author -- doesn't preserve anything beyond "this is number one"
    e. Is the worry with multiple names that there will not be enough role ability in MARC relator to differentiate people by roles, which can then be ordered? Can you identify roles by order, and then identify individuals within roles by # order?

i. Steven: you can definitely support both, depends on what use cases exist and what people want to support. Will be custom property however we define it, question of whether you want to order by sub-property or order every single name in the record.
    1. Consensus seems to be to order all marc relator terms if order is needed. So rather than "authors" and "photographers" having different order predicates, they would all be ordered in one order predicate...

5. Another topic: problem with affiliation
    a. BPL is fine with dropping it, but at HydraConnect sounds like Emory and University of Alberta would have uses cases where they would need that
        i. U of Alberta: not 100% sure (Note: Later clarified via email that they don't have a need for it)
        ii. (Emory University reiterated their need via email since they couldn't make this call).
    b. Is this an instance where the base recommendation would not include affiliation but the more complex recommendation would? Something that could be managed locally?
    c. Steven: it's something that we maybe don't implement if there are only one or two outliers that need it, but still something worth talking through as a group to tell those people how you might support it.
    d. Issue with affiliation is that you can mint an affiliation URI onto the object, but you can't accurately portray relationship both with time (when person was affiliated) and source record (under which affiliation record was created). If the person has two affiliations, how do you represent that the record is associated with the specific affiliation you want to associate it with?
        i. Attempt in BPL's most recent mapping to solve this but solution isn't that feasible.
        ii. No one else has any ideas how to solve this issue.
        iii. Punted on for now until the next call.

6. Simple vs. complex - simple version will be still using MARC relators to indicate the role, would resolve to a local URI that is simply a prefLabel. In complex case, would support prefLabel and then a few other broken-out properties.
    a. What's the difference again?
        i. In simple case, MARC relator that goes to a local URI and then just goes to a prefLabel; in complex case, same thing but support a few additional properties like birth/death date or affiliation if that needs to be supported in an example.
        ii. And if a precise MARC relator role is not identifiable, it's just creator?
            1. Yes, in that case it would just be MARC relator version of Creator
        iii. Agreement that separating out parts of the name represents an advanced use case

7. Action Items
    a. BPL will attempt initial implementation, see what this looks like in Fedora, throw in MODS XML records and see what comes out of them
    b. Homework: start looking at the next element in MODS, individual use cases & attempts at mapping -- Type of Resource, which is theoretically less complicated.
    c. Take a look at the collaboration document for Name once it is up.