TDWG Technical Architecture Group (TAG)
Online meeting
2023-05-08 14:00 UTC

Agenda/Meeting notes:

Participating:

- Steve Baskauf
- Raïssa Meyer
- James Macklin
- David Fichtmueller
- John Wieczorek
- Ben Norton
- Jonathan Rees
- Kit Lewers
- Wouter Addink
- Ian Engelbrecht
- Robert Sanderson

Regrets:

- David Shorthouse

Comments and notes taken during the meeting are in red.
NOTE: please make comments as text directly in the document rather than in marginal comments.

As with previous meetings, Steve will record the meeting for note taking and later viewing.

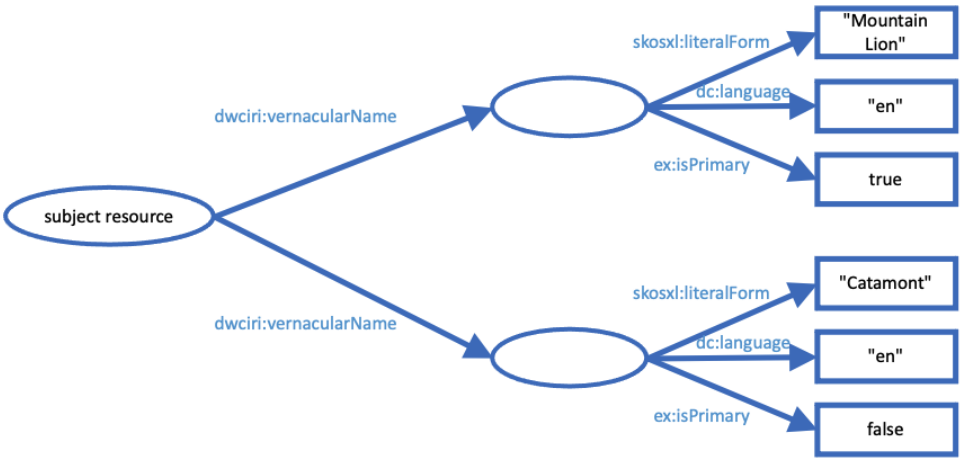
I. Set time and date for next meeting: suggest 2023-07-10 13:00 UTC.
II. Follow-up on items from the previous meeting
    A. Policy on boolean values (Steve)
        1. There was discussion about where this will go on the website, but it hasn't happened yet.
    B. Recommendations for expressing complex values
        1. Background information:
            a) Review of tabular and JSON serializations, ways to express complex values, and related problems. https://github.com/tdwg/tag/blob/master/complex_values/serialization_strategies.md
            b) Systematic categorization of complex value types, JSON serializations, and how to add JSON-LD contexts to make the JSON express an appropriate RDF graph. https://github.com/tdwg/tag/blob/master/complex_values/complex_values_categories.md
            c) Some collected complex value use cases. https://github.com/tdwg/tag/blob/master/complex_values/use_cases.md

    d) Review of existing precedents for handling complex values within TDWG vocabularies. https://github.com/tdwg/tag/blob/master/complex_values/existing_precedents.md
2. Steve introduction to the pressing issues. Presentation slides at https://github.com/tdwg/tag/blob/master/complex_values/complex_values_presentation_2023-05-08.pdf Video of presentation at https://youtu.be/9baIvyJNR4c
3. From the chat:
   From Robert Sanderson : Some hopefully useful links from the IIIF and related spheres:
   * IIIF - https://iiif.io/api/annex/notes/design_principles/ (especially section 3)
   * Linked Art - https://linked.art/api/1.0/principles/ and https://linked.art/api/1.0/json-ld/
   * W3C JSON-LD - https://w3c.github.io/json-ld-bp/
   From Robert Sanderson : The above principles would be to always do on the left hand side. Always an array, and always the same structure within it.
   From Robert Sanderson : In Latimer Core we renamed all our relationship properties to has<NameOfTargetClass> like hasIdentifier. To make it clear between (hypothetically) identifier: “foo” and hasIdentifier: {“value”: “foo”}
   From Ben Norton : If its a measurement class, then you could add a type field, populated with 'samplingEffort'. That way you can avoid using a 'type' as a class, then avoid confusion
   From Robert Sanderson : Can the specifications define the JSON structure, and then have a separate implementation note about how to deal with it in various data management environments?
4. Discussion.
    a) John Wieczorek pointed out that the existing Darwin Core Archive paradigm already has a solution to the problem of multiple values for values that need to be described by multiple properties. Create extensions for Event core by creating XML schemas for them. The "classes" for the multiple values would be defined by the extension. Example: a scope extension. This solution can be implemented now and doesn't require any changes to the Humboldt Extension or Darwin Core. It would just involve adding more tables to the archive that is built. A broader solution is needed if there are multiple values for terms in an extension (vs. the core file).
    b) Question: is there deep nesting in the Latimer Core proposal? Ben: they try to avoid it. In many cases, they just have a simple array. In the case of something like vernacular name, they would have "hasVernacularName" as a property linking to the class, which is described by multiple properties. If serialized as tables, multiple tables would be required with a GUID linking

the rows. Currently, they don't have terms for linking between the tables or a particular structuring specification like DwC-A. Question: how often do they expect people to "flatten" datasets in Latimer Core, or do they assume that everything is relational? In the LC standard, the model is relational – they don't (yet) provide any guidance for flattening.

c) Steve: Audiovisual Core has a "Structure Document" that basically says, "if you want to flatten things, just make up your own terms". That's not very satisfactory if we want to facilitate interoperability. It would be better to have some kind of rule-based system where tabular data could be transformed into JSON in a lossless way.

d) Discussion of this situation:



multiple values, object resource blank nodes

Ben: the subject resource would have a GUID and there would be a table of the values on the right that would link to the main resource using the GUID.

e) Question: how is linking going to happen in the GBIF Grand Unified Model? John: ID terms would be used, it is likely that tooling would provide the ID values where providers typically don't already provide them. In some cases, DwC would define new terms if they were needed, such as "vernacularNameID". So in the diagram above, there wouldn't be any blank nodes.

f) Steve: in the case of the Humboldt Extension, we facilitate both the simple and complex situations: mint terms for simple single values and use the star schema (by defining extensions) for the complex ones. What's happening in Latimer Core? Are they going to create something analogous to the DwC Text Guide that would tell people how to package things up? Ben: that's out of scope. They are providing a bag of terms, the structure is defined and they are leaving the rest to implementers.

g) Steve: How do people share data? Ben: there is JSON. They are basically saying that data should be shared as JSON and

not tables. Steve: there's nothing wrong with that – that's basically what IIIF does (JSON-LD). But that's different from other current standards. TAG people, how do you feel about TDWG having a standard that doesn't tell people how to express their data in tables? David: ABCD isn't table based. People have tables in their databases, but convert it to ABCD XML as an exchange format. Steve: TCS 2.0 is basically being described as RDF. So how is that going to be serialized? What if someone wants to combine DwC, LtC, and TCS data and it's in 3 different formats? We have this grand vision in TDWG that we are going to have Linked Data and link all these things together. How are we going to do that? Turn them into RDF and put them into triple stores? There is no appetite for that. As a TAG chair, I feel uncomfortable sending people DwC as tables, LtC as JSON, and TCS as RDF/Turtle and telling them to just figure it out.

h) James: In some ways TDWG has had a history of being too "nice". Maybe this is a time when we're not.
i) Steve: he feels comfortable with the JSON part because with some tricks using "@context" definitions, it can be converted to Linked Data. All of the semantic meaning that we would want to have would be preserved. How does that interface with tabular data? Is there going to be some systematic way in the GBIF model to translate it into RDF in a consistent way? If that's true, then basically the Linked Data/RDF graph is the way to interconvert everything.
j) Steve: do we talk about this more at the next meeting or just say that it doesn't matter and people do what they want? John: when does it become critical to have an answer? Before a public review of Latimer Core? Steve: If we are OK with Latimer Core saying "this is a standard whose structure is defined by JSON, we have a bag of terms and we have the structure" then maybe it isn't a problem with LtC, particularly if that JSON can be JSON-LD. Maybe we need to go into Darwin Core and see if there is a systematic way of transforming DwC tabular data to JSON-LD. If that's true, then anybody who wants to cross-link data converts everything to RDF (via the JSON-LD). John: I think there is a way. We can make the context documents. But it won't solve the problem of creating more structure where there isn't any. Unless we do things like parsing content of particular fields. And he's uncomfortable about proposing that. Steve: If there is a way to do it and we specify that, it's OK. Even if everyone doesn't use it, it's OK. David: Regarding the use case of what happens if people want to take data from different sources, then we are already at a level going beyond the scope of individual formats. At that point, whoever is doing that work is going to have to put some effort anyway just getting the data from the different scopes to

work together. The format is another step, but it's not fundamentally changing the work that is required to merge different sources. Having different formats isn't that big of an issue, particularly if you can get to JSON as a common format. Theoretically even the XML formats could be converted without much loss.

k) Steve: I'd propose that the ultimate goal would be to figure out how to turn everything into JSON-LD. It seems like from the examples Ben shared that it would just involve making the right context files. Audubon Core isn't that complicated and we could probably figure out a way to do it with Darwin Core. Ben: we have some JSON schema (not JSON-LD) files that are up on the LtC repository. They also have a script and E-R diagram for creating a relational database.

l) From Jonathan Rees (chat): Steve, csvw is a way to convert tabular files to/from json-ld

m) Raïssa: it is an important opportunity to bring up what we are doing in TDWG to a more sophisticated level. She likes that movement towards JSON-LD so that we can more easily use RDF as well. However, seeing it from the data producer side, her institute is still mostly using tabular data. So if we can somehow make use of one of the W3C conversion methods, that would be a good way of doing it. Although we might want to be more stringent, we don't want to exclude our producers who aren't as familiar with the different data types.

n) Steve: I've used the W3C tabular data conversion to RDF, but more people are into Frictionless Data Packages as a way to provide more meaning to tabular data. The uptake for the CSV on the Web Recommendation is very low. Maybe there is some way to work with the Frictionless Data Packages to include information that would allow for conversion to RDF.

C. Best practices for borrowing terms from non-TDWG vocabularies. Did not have time to get into discussing this.
   1. At the previous meeting we said we could continue discussion on this at today's meeting
   2. See notes from last meeting, new items B.
   3. Steve's final comment: The solution that most people seemed comfortable with was: if "same as" relationship: borrow, if not: mint. Audubon Core didn't really do that and therefore has "usage guidelines" for a number of borrowed terms. But that was a decision made at that time.
   4. Discussion if we have time.

III. New items

A. (add if any) None raised

IV. Any additional announcements.
   A. A symposium and a workshop session for “Building bridges: Mapping from, to and between TDWG standards” organized by Mareike Petersen, Holly Little and David Fichtmueller was accepted for TDWG 2023
   B. Kit: I am planning on submitting an abstract regarding the survey for use of standards and standard retirement; planning on reaching out to the organizers after finals

V. Action items for next meeting (or before):
   A. Steve: publish booleans documents somewhere on the TDWG website.
   B. Steve: check for repositories and issue trackers to see if there is an obvious one for each standard.
   C. Camilla: volunteered to do the Spanish translations for the Boolean vocabulary.