TDWG Technical Architecture Group (TAG)
Online meeting
2023-09-11 13:00 UTC

Agenda/Meeting notes:

Participating:

- David Shorthouse (3-5 min tardy)
- David Fichtmueller
- Kit Lewers
- John Wieczorek
- James Macklin
- Ben Norton
- Anhita Kazem
- John Kunze

Regrets:

- Raïssa Meyer
- Ian Engelbrecht
- Camila Plata

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: Monday, 2023-11-13 13:00 UTC.
II. New items
    A. Null values, particularly with respect to booleans.
        1. The subject of recommendations for null values in booleans has come up twice recently:
            a) In the Boolean Values Best Practices Reference, Section 2 (Recommendations) number 6 https://tag.tdwg.org/reference/boolean/#2-recommendations, which basically says that there should never be missing values for booleans.
            b) In the context of providing values for proposed Humboldt Extension terms that require boolean values. In some circumstances, the guidelines recommend that values should not be provided, for reasons given in Section 3.2.3 of https://eco.tdwg.org/hierarchy/#323-principle-of-non-derivation.
        2. What we have here is basically two contradictory recommendations.
        3. The succinct serialization recipes that we have provided at https://tag.tdwg.org/guides/boolean/ simply do not include missing values in examples. Likewise, the controlled vocabulary that we have created for booleans https://tag.tdwg.org/boolean/ provides values

only for true and false and gives no indication about the circumstance of missing values.

4. Humboldt Extension is currently in public review and if ratified, it will provide normative guidance that is at odds with the guidance we've given in the Best Practices Reference. What do we do about this?
5. For reference, I asked the Biodiversity Data Quality Task Group 2 (Tests and Assertions) about their use of EMPTY as a value. It appears that this value is relevant only in the context of their testing framework and doesn't represent a recommendation for serializing null values. See https://github.com/tdwg/bdq/issues/152 for more as well as this explanation from Paul Morris:

   Within the very specific context of the tests, where we expect that the input to tests will be some text serialization of DarwinCore data, such as csv or tab delimited text files, where a cell contains a non-typed data value where data are likely to have been aggregated from and serialized from multiple sources, including relational databases where boolean nulls and non-string data types may exist, but the data have been exported into a string serialization that supports neither null nor typed data, and where we need a reusable definition for EMPTY that can apply in any test where the concept is relevant, we defined:

   "EMPTY

   An information element that is either not present or does not contain any characters or values other than those in the range U+0000 to U+0020.

   Note: An information element containing invalid characters (e.g. letters in an information element that would be expected to contain integers) or values (including string serializations of the NULL value) are NOT_EMPTY and may be separately detected."

   The phrase "not present" is there to cover cases where a test implementation cannot tell if a particular data set under test includes a particular darwin core term, this allows the test implementations to be independent of and agnostic about frameworks within which the tests are run, and the nature of the data, for csv data, a column is either there or not in a data set, but in an rdf representation, some data objects could have relevant properties and others not - empty and the tests are independent of that.

   We considered, and explicitly rejected, treating common string serializations of null such as \N and NULL as empty values. If "\N" is present in a data set, the tests will explicitly treat that value as NOT EMPTY, and then try to evaluate it against whatever other criteria apply.

This definition is not applicable to a discussion of what value to include in a controlled vocabulary to indicate that no meaningful value is present, so no suggestion is made that "EMPTY" should be used as a data value to represent some form of "Null", "Unknown". "Not Recorded", etc. Choices there would fall into the semantics for some set of controlled vocabularies. The relevance to such a discussion is that this definition would treat an empty string as an empty value, with no semantics attached as to why the value is empty.

Discussion:

Steve: The Humboldt Extension is the first vocabulary to prescribe Boolean values for its terms. Under certain circumstances, it's extremely important to leave those fields blank.

John W.: The fact is that these terms are needed to make inferences of presences and absences in some cases. But they don't apply in every circumstance. The Humboldt Extension is based on events, and not all events are the same kind of events. Some have no role at all in making inferences and are just about project level information. So there are cases in which the terms have to be there, but neither true nor false are appropriate. Given the structure where an extension record must connect to a core record, and given that the most popular way that data are shared is via CSV, there's no way around this. Humbolt can't follow this TAG recommendation because it just won't work.

Anahita: I can't see Humboldt Extension working unless there are three possible values: ideally true or false, but a null can exist. In some frameworks, there's a "null Boolean" system where all three values are possible, and it seems like that's the situation we have here.

Steve: My understanding is that it's not that you are providing a null value, but rather you aren't providing any value. I think that's two different things. The Data Quality Task Group doesn't really have a policy on provided values, but rather they have a way of indicating inferred values when they are missing. Aggregators would provide a value indicating that the field was empty. But they didn't have a recommendation on actually providing a null value.

John W.: things are different depending on how the data are serialized. For example, in JSON, if it wasn't appropriate to provide a value, the key just wouldn't be there. So you wouldn't be violating the recommendation by providing third value, you

just wouldn't be providing a value at all. The term wouldn't be there because it's not applicable. But we are stuck with CSVs with rows and columns, where true and false aren't the only possibilities. So we are stuck.

Steve: Maybe what we need to do is to go into the "recipe" document and give this specific example. If the property applies, then you should only use true or false, but if the property doesn't apply, you would leave it blank in a CSV and omit it in serializations like JSON or XML where that is possible.

Ben: It seems practical and pragmatic to have null values in CSVs. There's only two logical values, but there's a third situation where it's not applicable. So in a CSV if it's a null then you would assume it's not applicable.

Steve: So our conclusion is that we don't have a problem with the Humboldt Extension. Rather we need to add more to our recommendations to provide guidance in situations like this.

Anahita: Sometimes something is not applicable and depending on your implementation you might put that as null. But isn't it the case that it doesn't necessarily mean that it's not applicable at this level, but it can also be (given that you are getting data from all kinds of people) that it just hasn't been given. Maybe it is applicable even for that level in the event hierarchy. So a missing value can't necessarily be interpreted as not applicable.

James: The only other way to do this is to do a non-preferred method: separate it into two fields: one is about the applicability and the other is about the Boolean value. It's ugly.

John W.: I don't know if this would help in the CSV scenario. You could say "it's not applicable" but then put a false in there!

John K.: About 20 years ago in Dublin Core I introduced a small vocabulary of missing or unknown values. If you were going to require values, then you would at least have some way to explain why values were missing.

Steve: Right now we have established a controlled vocabulary that only includes the two values: true and false. It's not part of any standard and we could extend it with these additional values that are neither true nor false. If as Ani said, you want to be able to say that we don't have a value for something then maybe we need additional values. It would differentiate

between an empty cell, which means we didn't provide and answer, and explicitly providing a value and saying it's one of these other circumstances. It's good to know about this prior art (the DCMI vocabulary) in the event that we wanted to provide additional values. Conclusions: 1. Humboldt is OK, we don't need to comment (the proximate issue). 2. We probably need to improve our documentation. 3. The longer term question is whether to extend the vocabulary or not.

Ben: If a Boolean value is null, is there a scenario where it should be included in the data set. Meaning that that null has meaning to it. If the value isn't there, then the key shouldn't be included since it's not there. but is there a case where the value has meaning to it and it should be included in the dataset.

David: Specifying a null means "I can tell you that there's no value." Leaving it out can mean that no value was provided or transmitted, but there is one. Having an explicit null means that there is no value here. There is a difference between providing an explicit null and just omitting it.

Anahita: I'm in favor of explicitly stating null. It's a problem if we aggregate data from some format like JSON where we drop a key and mix it with data from a CSV. It would be better to force people to say it's null. Why it's not here is another question, but at least you know it's not here.

Steve: Well then do we need to quickly create a null value and say to use it in Humboldt Core? Because right now it’s an empty cell.

Ben: If it's for interoperability at ingest it's a simple question. If you need it to merge a CSV with JSON that's a different question than if it has meaning.

John W.: I also worry in the common case of sharing data with CSVs where you need to create a string that means "null", that's troublesome to me.
If there were controversy about a term in Humboldt Extension that used Booleans, that would be bad because it would have to be left out and the extension wouldn't work. But there isn't really anything binding in the comments that say best practice is to use a controlled vocabulary. So people could start sharing data and Humboldt would go along even if action on these issues comes along later.

Steve: If the term metadata suggest using the controlled vocabulary, then future changes would take place at the level of the controlled vocabulary and would not necessitate changing a bunch of term definitions.

John W.: the XML schema points to the GBIF controlled vocabulary that lives on GBIFs server. And that includes null as a possibility. GBIF does generally follow the values from the officially ratified controlled vocabularies.

Conclusion:
There was consensus that the Humbolt terms refer to the TAG Boolean vocabulary and that Latimer Core do so as well. Then the issues could be hashed out at the TAG and not require many term changes by two different groups.

The TAG should have additional discussion about the issues we discussed today.

III. Follow-up on items from previous meetings
   A. "Technical" menu in TDWG website with links to our recommendations.
   B. Status of the Identifiers Task Group
      1. Have not heard back from Dave Bloom.
   C. Best practices for borrowing terms from non-TDWG vocabularies (held over from previous meeting).
      1. Needs more work. Do we want to craft a policy, or kick this to the Mapping Task Group?
         David: the group is getting started. They might put this on the agenda, but he doesn't want to make any promises about whether they can get to it.
         Steve: it seems pretty directly related because we are talking about the circumstances under which you borrow terms vs. circumstances under which you mint your own terms and map. And that's very directly related to mapping.
         David: it seems like it would fit in the scheme of the group.
   D. Recommendations for expressing complex values (held over from previous meeting).
      1. Needs more work.
         Not time to discuss in this meeting.
   E. Categorization of prior standards and retiring standards.
      When the website was revised, all of the standards got thrown into one group. That situation was really unacceptable, but it's the status quo until the TAG does something about it. That's why it's time-sensitive.

      The proposal is NOT primarily about what to do with retired standards, but rather about making the standards web pages more usable.

1. Review proposal at https://docs.google.com/document/d/1qg8JKHmsEzBWcSo2Ip0ga6PcS4vJjTt2-r-XJALjk5E/edit#heading=h.y5jixqti1sjq
2. Discussion
3. Recommendation to Executive Committee.
   There was a significant amount of time spent by Steve explaining the rational of the proposals and their implications, with a few clarifying questions. But there was no real objection to recommending the proposals and the group agreed to send it to the Executive Committee as a TAG recommendation.

IV. Any additional announcements.

A. Ben: Latimer Core's documentation is generated by a Flask application that takes CSV input. He hopes to extend this mechanism so that it can be used by any maintenance group to generate documentation. There's a GitHub repo for it.
   Steve: There's an existing (parallel) pipeline for doing this -- we should get together and make one best system There is a separation of responsibilities: the maintenance groups are responsible for generating the human-readable documentation and the standards maintainers (Steve mostly) feed data into a server that generates the machine-readable metadata on the fly. Both forms of documentation must be generated from the same CSV files in the rs.tdwg.org repo so that the docs provide identical information.

   Ben and Steve will talk about coordinating this effort.

V. Action items for next meeting (or before):

A. (old task) Camilla: volunteered to do the Spanish translations for the Boolean vocabulary.
B. Ben and Steve to coordinate on streamlining the process of generating human and machine readable standards documentation.
C. Steve will forward the proposal for categorizing standards to the Executive Committee.
D. David to put criteria for borrowing vs. minting under consideration as a task for the Mapping Task Group.