OWL/RDF Tuesday morning

10.17. 2017

The XML can be validated by the schema, but for RDF there is not a similar approach. There are two languages you can use to define tests for validation purposes.

Eric: ShEx2 - Simple Online Validator, writing a grammar for what a graph would look like. Can run across SPARQL.

Can you validate the schema or just instance data? It just parses the schema (a trivial validation). Doesn’t work with RDFS, not sufficient information for validation. From the PSM you would generate the ShEx file from which you would validate the instance.

How to define more complex tests (on data variables, for instance)? How much manual work is necessary?

FHIR schema is generated by its own native structure definitions (basically because he wants it to look nice). A simple JSON representation takes no work at all (not human readable).

DDI Schema shouldn’t be too complicated, though.

Users need: Validator (java, javascript…) See shex.io for the implementations.

ShEx is more grammar oriented (Schema), SHACL can be extended to be represented in SPARQL (more of a rules language, more like Schematron).



Can you write a configuration file that can output a ShEx, SHACL, etc.? The original PIM has a label property and we are then imposing more constraints because everything in the modle is optional (for the most part), so an institution profile, more restrictions will be needed.

Machine-readable profile that restricts the original PIM, but everything in the profile is available in the PIM. The issue of validating against different rules. Validating against DDI (that is looser), but they are actually validating against a more strict sense of rules.

FHIR is a loose schema as well, but for clinical interoperability, machine learning, etc. you need to control the value sets and the units and types of tests. Argonaut (institution) building tighter profiles. But they are doing it at the model level, not the configuration. An identifier for the constrained profile on which you can base rules for acceptance, “valid according to identified profile AND valid according to the full schema)

**Summary:** Validation will not be part of the Prototype, too much work from now. Validation against PSM on basic things, like datatypes, is important. Possible to generate both languages without too much effort. Experiences from FHIR. Need a secondary validation to a subset of the whole DDI for the purposes of functional views, user groups, and tools.

**Item for discussion:** How to identify and define the local implementation of a functional view (subset of the model). As long as you stay within the circle of the model and don’t extend beyond it.

For each view there is an RDF vocabulary.

CESSDA - Not using DDI, the keynote at EDDI will be on how DDI is not enough?

**Item for discussion:** Might make sense to include in the prototype a general validation without subsets, subsets should come after prototype publication.

During FHIR development, schema developed along with examples and the tools. Tons of errors found in the generated RDF. Many may have gotten by without a schema.

**Second Morning Session:**

Discuss the syntax rules

**Afternoon session:**

**Question:** Using classes of other vocabularies on the PIM level? - the advantage is the definitions would be available in the full downstream. The round-trip between bindings would be supported. The disadvantage - it’s work to go through the model to see which DDI item corresponds to a class of another vocabulary. Is it close enough to be reusable. If this class is the same as DDI then we have to look at all the attributes, which will be different.

Counter-point - Guillaume - you can use the skos concept without all of the attributes & properties, but skos and xskos doesn’t care about an exact match. You can extend the class or restrict the class. Jay - problem of semantics of what you can do with skos, xskos, and ddi in the relationships.

Darren: How do you choose? The DDI collection patterns or the skos relationships. dc: copyright. Easy to bolt it on at the end of the chain, because if you put it in the model do you take in the rest of the classes?

Downside is how to realize it in the XML and the round-trip requirement.

If you take the time to use the attributes and relationships, will it be more widely adopted if the users don’t have to?

Data integration across scientific domains - codata conference (2nd meeting in November) Where does the DDI Alliance locate the standard in this new world? If only for traditional DDI users, development can proceed more quickly. Data description, as is currently exists, could be useful to this group. Should it be published as a standalone standard, not the DDI Alliance. Could then relate in some way to DDI.

Darren: Should have had this discussion 2 years ago. Use SKOS concept.

**Question:** Integrating other vocabularies only on the binding level?

(Should not have multiple DDI bindings in the public space)

If it’s just a matter of property transformations, just start doing the mapping, if more complex transformations then we sigh and move forward.

If you can map both to your base level...the abstract model (very generic) the canonical representation. If you look at a more specific model and a binding it shouldn’t be too different. The RDF lists become difficult. Are we pushing the transformations to the tools, not the specification? FHIR - you can go from XML to substantiated object madel to RDF, it can’t require extra information that wasn’t there. Everthing there must come back. FHIR metadata vocabulary (insert link).

Substituting one property with another property…

Does each binding have its own community? Java, C#, python...yes. Tool-chain ghettos. Based on language, not content.

Eric: We should leave that up to the person writing the tool.

Achim: if we push to the user community, then a non-standard or RDF and XML and they don’t talk to each other.

Eric: assumed there would be someone in the DDI community writing this tool?

Jay - Round trip- you can do it, no matter how you decide to do it.

Individual Recommendations:

Darren: retrofit existing vocabularies into the model would be a lot of pain, but it would be ideal if you were starting the development today

Guillame: definitely put it into the PIM, not a huge amount of work.

Ben: also include it into the PIM, especially considering the points from the user community. More work for implementers, too much work therefore I should just do my own model.

Jay: With specific vocabularies, it would be very intrusive to put in PIM. With others it would be less intrusive. I would start on the binding side and then migrate specific vocabularies that complement what we have into the PIM, but other vocabularies.

Darren: if you are only taking the class name from skos, then why mislead the user that they now have access to the skos vocabulary. Why bother?

Eric: There are plenty that won’t be difficult. If not an easy fit (skos) then don’t use it. Decide if it’s worth the effort for each one.

Achim: The requirement of round trip as important. The interoperability also. Opens future worlds of DDI users. The price is high to do the work. For teh prototype, maybe we ignore the vocabularies? We produce a good RDF binding, maybe a core round trip between XML and RDF. Later on, we think more about which vocabularies are easy to incorporate.

Jay: Iterative approach. There may be some cultural issues to subsume some of these vocabularies.

Achim: In the end, we need something with strong relationships.

Jay: We should begin on the island.

Say more about the subtleties of round-tripping, some things that FHIR does...