Round tripping minutes 2017-10-19 pm

Assuming a tool to convert from XML to RDF or vice versa, will the round trip results be the same? This will require that something be written to use the generated classes.

What language should be used for implementing the prototype Java?, C# ?, Scala? (lives in JVM). Should this be implemented in a framework that gives the ability to have a Web application which could be provided to users. Forms to convert among bindings or run tests.

Comparisons - pairwise would require the number of combinations of the N bindings two at a time C(nbindings,2) = N\*(N-1)/2 comparison tools. This gets big with many bindings.

It would be better to have a tool that could consume instances and represent them internally in some common form.

Identification, Content,

A tool would be needed to do the comparison,

W3C canonical XML

 Has fussy requirements e.g. precise (used for digital signatures)

Class definitions with

Tooling spit out by same tool used to generate XMI?

Comparisons of XML to RDF by two different tools ( or before and after round trip)

 Isomorphic graphs (sparsely implemented tools exist to do this. , e.g. Java, and they work (JavaScript uses Eric’s C code)

Equivalence when not isomorphic?

 Comparing sets - need canonicalization

 Red black and green is set of green black and red when not ordered

 Tool would need to sort in some fashion (e.g. lexically) and then do equivalence for unordered sets and bags

XML to abstract model and RDF to abstract model if models equivalent then equivalent

Bindings

 XML

 RDF

 SQL

 Program libraries

 Java

 CSharp

 Python

JavaScript

 JSON(-LD) (one fixed order?)

generate JSON from the PIM not from RDF

Compare with RDF as a check

RDF to JSON-LD might not be very useful

Validation tools

JSG - originally JSON form of ShEx

JSON schema (widely deployed)

Java objects via JAXB-2 from the XML Schema might be useful

Javascript is a well trod path

Issues

 Hash maps (are they used in DDI?) and Sets don’t allow strict equivalence

A program language binding as the comparison basis

Annotating program libraries to ingest the bindings might be difficult. JAXB type filters need to be created. A hand crafted code generator would need to be developed.

XSLT or equivalent is one approach to spitting out binding

Is an alternative to go to program library directly annotated?

**Summary**

*The Prototype should both consume and output XML and RDF via an abstract representation. The test cases would make sure that the round trip is possible and provide guidelines for implementers to avoid different dialects for DDI4.*

Two approaches to using vocabularies in RDF

 At model level

 In binding process

Serialization and parsing require knowledge of equivalence between serialized form and the abstract representation. When using other vocabularies in the RDF bindings we need explicit mappings between model classes and properties and external RDF names

 (e.g version ⇔ pav:version). This is a format binding transformation process which exists probably for every format binding.

RDF naming conventions (object uppercase, properties lowercase)

Need clear rules for case in XML bindings

Identification

 DDI has used ISO 11179 for IDs. How do we do that in other bindings.In XML version of an item includes properties and relationships. With RDF a SPARQL query might not return all of the properties and relationships for a given object. Could this be checked for being the same version as another RDF representation?

Have a media type for DDI4 RDF triples?