**W3C rax Template for Use case and Tools description**

**Topic title:**

Create XML documents from RDF data

**Contributed by (Name, Affiliation):**

Bernát Kalló, z-Bible.org

**Date of contribution:**

5 Jul 2016

**Please give a short description of problem you want to address (if possible with links to previous work, etc.):**

Given an RDF dataset containing Bible text and related information, we need to render a user-readable output from these. The first step towards a user-readable format is an XHTML-based XML format. For example, if we have

z:Matt a z:BibleInterval ;  
 z:title "Gospel of Matthew" .  
z:Matt.1.1-Matt.1.17 a z:BibleInterval ;  
 z:in z:Matt ;  
 z:title "Genealogy of Jesus" ;  
 z:text "The book of the genealogy of <em>Jesus Christ</em>..." .  
  
we need to render z:Matt as follows:

<z:map id="Matt">  
 <z:label><h1>Gospel of Matthew</h1></z:label>  
 <z:box id="Matt.1.1-Matt.1.17">  
 <z:label><h1>Genealogy of Jesus</h1></z:label>  
 <div class="textBody">  
 The book of the genealogy of <em>Jesus Christ<em>...  
 </div>  
 </z:box>  
</z:map>

We need a document that can describe this transformation. It needs to be capable of:

* Having input parameters (e.g. to use z:Matt as the root is provided in a parameter)
* Outputting RDF data to XML with various serializations (e.g. IRIs with a prefix removed [id attribute], escaped text [titles], unescaped text [text])
* handling namespaces
* executing complex RDF queries and order results based on various criteria (about the same feature set SPARQL Select queries provide)
* generating sequential output (results of a query as consecutive XML elements, ordered by some value) and conditional output (e.g. different XML output based on an RDF value)

As in many web development environments, templates for markup are edited by designers who are not programmers. This imposes some new requirements for the syntax:

* The format should be concise, easy to read.
* Declarative syntax is preferred over imperative.
* English keywords are preferred over punctuation.
* An XML format is preferred (people familiar with HTML will be more likely to understand XML than a traditional programming language)

Though we need to convert RDF to a custom format, the tool I envision should be also capable of generating standard XML formats from RDF data, like XHTML pages, or even simple SVG diagrams.

**What kind of solution/deliverable would you expect from the group (process, format, tool, best practice, etc.):**

I’d expect a format for describing an RDF → XML conversion.

**Are there already solutions available that try to tackle that problem (even partially); please describe, validate and give links, e.g. to tools:**

* SPARQL Web Pages (formerly UISPIN)  
  [uispin.org](http://uispin.org/)  
  Uses SPARQL queries to generate XML. It is closely tied to SPIN, a format for encoding SPARQL in RDF. Developed by Top Quadrant.  
  My personal opinion: I like the conciseness and declarativeness of the syntax. Contains many interesting and useful ideas to make the markup clear and more concise, e.g. a dedicated XML namespace for declaring variables. I think the conciseness could be a bit further improved, and the close integration with SPIN makes me doubt if it can be used without SPIN at all.
* XUL templates  
  <https://developer.mozilla.org/en-US/docs/Mozilla/Tech/XUL/Template_Guide/RDF_Query_Syntax>  
  Generates XUL elements based on RDF data. Contains an XML-based RDF query language. Developed by Mozilla.   
  My personal opinion: Even though the query language is simple, it is much more verbose and less powerful than SPARQL. Many times there is no way to shorten IRIs with prefixes. However, I like the concept of rdf: urls that easily reference data obtained from the RDF dataset.
* XSPARQL  
  <https://www.w3.org/Submission/xsparql-language-specification/>  
  A fusion of XQUERY and SPARQL for the purpose of converting between XML and RDF. A W3C Member Submission.  
  My personal opinion: Even though the language is very powerful and the merging of the two languages seems well-thought-out, for my purposes a declarative language would be better.

**Would you be able to spend resources in that group effort?**

I have not decided yet how much.