Ongoing maintenance and customization of archival standards using ODD (EAC-CPF revision proposal)

Laurent Romary, Charles Riondet

Ongoing maintenance and customization of archival standards using ODD

Laurent Romary[1][2], Charles Riondet[2]
Dariah (1), Inria (2)

The EAC-CPF tag library is natively expressed using the TEI (Text Encoding Initiative) guidelines and maintained collaboratively on GitHub. This solution has already proven to offer some flexibility. Starting from this, we propose to go one step further and **create a complete maintenance framework of EAC-CPF based on the technical means provided by the TEI guidelines**.

A well-documented framework

The Text Encoding Initiative (TEI) is broadly recognized as the *de facto* standard for the representation of a variety of textual content expressed in digital form, but the TEI can be used to represent a wider range of digital resources. For instance, the TEI XML schema and the associated guidelines are maintained with the TEI format, more precisely, with a subset called "One Document Does it all" (ODD) which, as the name indicates, is a description language that "includes the schema fragments, prose documentation, and reference documentation [...] in a single document"[1], based on the principles of literate programming.
Literate programming is a programming and documentation methodology whose "central tenet is that documentation is more important than source code and should be the focus of a programmer's activity"[2]. With ODD, semantic and structural consistency is ensured as we encode and document best practices in both machine and human-readable format. ODD was created at first to give TEI users a straightforward way to customize the TEI schema according to their own practices and document this customization.

It is possible to describe a schema and the associated documentation of any XML format. In the context of the EHRI project (ehri-project.eu), ODD was used to encode completely the EAD standard, as well as the guidelines provided by the Library of Congress. The maintenance on a GitHub repository also offers great possibilities to collectively discuss potential issues, enhancements, etc.

[1] TEI consortium, 2013, TEI: Getting Started with ODDs. http://www.tei-c.org/Guidelines/Customization/odds.xml, accessed May 24, 2017.
[2] Walsh, Norman, 2002, Literate Programming in XML. http://nwalsh.com/docs/articles/xml2002/lp/paper.html, accessed November 29, 2017.

Flexibility

We propose that the EAC-CPF specification is maintained via an ODD document on a GitHub repository. Such way have demonstrated both its robustness and flexibility, for the reasons explained above. Furthermore, it can offer real benefits to the users community.

Applied to EAC-CPF, this framework would also meet the need for more specified subsets of EAC-CPF by implementing the refinement of element content, currently covered by <localTypeDeclaration> and @localType.
As ODD allows anyone to derive a specific schema from the standard one, the compliance towards local conventions or controlled vocabularies can be integrated into the standard validation process.

Let's consider the current example for <localTypeDeclaration> in the Tag Library[3]:

```
<localTypeDeclaration>
       <abbreviation>Categorycodes</abbreviation>
       <citation xlink:href="http://nad.ra.se/static/termlistor/Kategorikoder.htm"
       xlink:type="simple">
       The categorycodes used in Swedish NAD (http://nad.ra.se).To be used in element
function.
       </citation>
       <descriptiveNote>
       <p>Codes for categorizing different types of authority records through
organizational form, operation, function, archival organization et cetera.</p>
       </descriptiveNote>
</localTypeDeclaration>
```

With ODD, it would be possible for the National Archives of Sweden to create an institutional flavour of EAC-CPF, adding a constraint to the element <function> that limits the possible values to its specific category codes. The flavour would take the form of a new ODD specification that inherits the characteristics of the "Master" EAC-CPF, but would declare (and document) specific constraints (directly using the ODD declaration mechanisms or via the Schematron standard).

The benefit is that the technical validation of the schema (via a saxon processor for example), would automatically check this semantic constraint.

Another added value is the generation of the associated documentation for this constraint, that can allow a CHI to share its practices more easily. Then, we could also imagine that such mechanism would facilitate the identification of widely shared "local" practices.

An additional usage of a ODD specification of EAC-CPF could be a simplified management of successive versions, where each new version could be derived from the previous specification document.

[3] http://eac.staatsbibliothek-berlin.de/schema/taglibrary/cpfTagLibrary_EN.html#localTypeDeclaration

Summary

- ODD can be processed to generate an actual schema (a DTD, a RelaxNG XML or compact schema or an XML schema), as well as the corresponding documentation in various formats (XHTML, PDF, EPUB, docx, odt).
- ODD also allows for chaining and derivation, i.e. it would be possible to create a new version of the EAC-CPF schema by inheriting the core of the previous one and just update, add or remove new elements, or by over specifying the behaviour of them (see above).
- An EAC-CPF specification in ODD already exists, hosted on GitHub by the Parthenos Project[4]: http://github.com/ParthenosWP4/standardsLibrary/blob/master/archivalDescription/EAC/Spec/EACSpec.xml

This version is at the disposal of the EAC-CPF maintenance committee, and we are willing to provide technical help to implement this solution.

[4] http://parthenos-project.eu