Some remarks on the EAC-CPF XML schema

Detlev Balzer[1]

05 March 2011

These are some observations from an attempt to use version 2010 of the CPF schema as an exchange format for person authority files from the Deutsches Filminstitut (DIF). Implementation-specific comments relate to the XSD version of the schema.

Identities

EAC-CPF defines five concepts of identity, two of which consider multiple identities, but none which would refer to a consolidated identity, i.e. an identity that has been composed of two or more identities while eliminating all redundancy. This can occur whenever two or more data sets about persons or corporate entities are merged and resulting duplicates are united.

A "consolidated identity" would also be useful whenever authority data from an archive or library context is put to use by other communities for which multiple identities are not acceptable, e.g. by their definition of a person as an indivisible entity ("individual").

Semantics of attributes fromDate and untilDate

Presumed application:

```
<fromDate notBefore="1850" notAfter="1860">1855<fromDate>
<untilDate notBefore="1895" notAfter="1905">1900</untilDate>
```

Presumed semantics:
If a date value is qualified with `notBefore` and/or `notAfter`, then it is to be processed as an uncertain date. Otherwise it can be assumed to be certain (whether in the form of a `standardDate` or not).

Question:
Could the tag library be a bit more precise here? Computing and reasoning over time spans (even fuzzy ones) is not too uncommon any more.

Connecting time and place

```
<existDates>
  <dateRange>
    <fromDate standardDate="1765-09-18">September 18, 1765</fromDate>
    <toDate standardDate="1846-06-01">June 1, 1846</toDate>
  </dateRange>
</existDates>
<place>
  <placeEntry altitude="389" latitude="46.140833"
longitude="12.215556">Belluno</placeEntry>
  <placeRole>Birthplace</placeRole>
  <date standardDate="1765-09-18">September 18, 1765</date>
</place>
```

This example from the tag library illustrates the diversions and redundancies that follow from the lack of an event (i.e. space-time) node for the description of the life span of an individual. It also violates the philosophical definition by which a place (or better: region of space) as such is something without temporal properties.

[1] Detlev Balzer is an information technology consultant working for several major cultural heritage institutions and projects in Germany and neighbouring countries.

Suggestion:
Introduction of an event type that can be specialized into lifecycle events of an individual, such as birth, death, begin and end of an occupation or mandate, marriage / divorce, etc.
This would also avoid the questionable assignment of activities to a person as a whole. In modern societies at least, nobody gets born as a prime minister or a railway labourer. These are properties of periods (i.e. events) within a life span. Unknown time spans do not invalidate the notion of event.

I cannot estimate to what extent this can be brought into accordance with ISAAR(CPF). However, the CPF taglib defines an el

Regional variants of name entries

`nameEntry` has an element for indicating a time span during which a name was used, and also an `xml:lang` attribute. Names, however, can also vary between geographic regions without belonging to a specific language. As an example, an artist's name may have been chosen for (or attributed by) an audience in a specific region where more than one language is spoken.

ISAAR(CPF) mentions "territorial designation" as a possible part of a name, however, this probably applies to the realm in nobility titles.

Local type declarations

`localTypeDeclaration` can be used to link an abbreviation to a URI reference in a similar was as in an XML namespace declaration:

```
<localTypeDeclaration>
    <abbreviation>difzf1</abbreviation>
    <citation xlink:href="http://filmstandards.org/localdefs/de/dif/zf1"/>
</localTypeDeclaration>
```

where this declaration could be referenced as follows:

```
<nameEntry localType="difzf1:VN">
      <part localType="difzf1:Vornamen">Dorothee</part>
      <part localType="difzf1:Nachname">Dhan</part>
</nameEntry>
```

In principle, this could be made to work like an XML namespace reference, provided that it can be transformed in a pre-processing step (using e.g. XSLT).

Alternatively, local type references could always be given as full URIs, obviating the need for using the `localTypeDeclaration` element:

```
<nameEntry localType="http://filmstandards.org/localdefs/de/dif/zf1/VN">
      <part
localType="http://filmstandards.org/localdefs/de/dif/zf1/Vornamen">Dorothee</part>
      <part
localType="http://filmstandards.org/localdefs/de/dif/zf1/Nachname">Dhan</part>
</nameEntry>
```

None of the above methods will produce a CPF-compliant XML document in which elements from foreign namespaces can be validated. Finally, there is the option of changing the EAC schema such that it permits foreign namespaces within selected elements, e.g.

```
<xs:element name="part">
      <xs:complexType mixed="true">
            <xs:sequence minOccurs="0" maxOccurs="1">
                  <xs:any processContents="strict"/>
            </xs:sequence>
```

```
             (...)
```

which requires the additional namespace being declared in the XML instance, e.g.

```
<eac-cpf  xmlns="urn:isbn:1-931666-33-4"
    xmlns:difzf1="http://filmstandards.org/localdefs/de/dif/zf1"
    (...)
```

allowing us to wrap up foreign elements in a CPF element, such as

```
<part>
      <difzf1:Vornamen>Dorothee</difzf1:Vornamen>
</part>
```

At a first glance, this would simply weaken the control over the type of CPF element contents. Then again, what can an XML schema processor make of a `localType` attribute? Since any content labelled as `localType` must be taken at face value, a method for validating foreign types will actually allow for increased control over the content of CPF elements, particularly when parsing of foreign types is enforced with the `processContents="strict"` directive.

Odds and ends

Definitions of attribute groups `a.style` and `a.vocabularySource` are never referenced in the schema.