## Issues with Black Pear openEHR TDD XML

Review of the sample Black Pear openEHR Template Data Document example TDD (BP-TDD)

1. Language code\_string must be in lower case.

TDD:

<Initial\_Optometrist\_Glaucoma\_Assessment xmlns:oe="http://schemas.openehr.org/v1" xmlns:te="http://schemas.openehr.com/templates" xmlns="http://schemas.oceanehr.com/templates" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" archetype\_node\_id="openEHR-EHR-COMPOSITION.report.v1" type="COMPOSITION" template\_id="Initial Optometrist Glaucoma Assessment Encounter" >  
 <!-- All Archetyped nodes (anthing with an ‘archetypenodeid’ attribute), need a 'name' attribute - generally defaulted in the schema by the 'fixed' attribute. -->  
 <name>  
 <value>Initial Optometrist Glaucoma Assessment</value>  
 </name>  
 <language>  
 <terminology\_id>  
 <value>ISO\_639-1</value>  
 </terminology\_id>  
 <!-- EN needs to be lower-case -->  
 <code\_string>en</code\_string>  
 </language>

TDS:

<xs:element name="language">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="terminology\_id">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="value" fixed="ISO\_639-1" type="xs:token" />  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
 <xs:element name="code\_string" fixed="en" type="xs:string" />  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>

1. Composer is a mandatory attribute. This is equivalent to the CDA Author. The PARTY\_PROXY class must be sub-classed at run-time to a concerte class – usually PARTY\_IDENTIFIED

TDD:

<composer xsi:type="oe:PARTY\_IDENTIFIED">  
 <oe:name>Ms.A Optometrist</oe:name>  
 <oe:identifiers>  
 <oe:issuer>NHSWALES</oe:issuer>  
 <oe:assigner>WSDS</oe:assigner>  
 <oe:id>1234567</oe:id>  
 <oe:type>WNACSCODE</oe:type>  
 </oe:identifiers>  
 </composer>

TDS:

<xs:element name="composer" type="oe:PARTY\_PROXY" />

1. All archetyped nodes (anything carrying an archetypeNodeId XML attribute) must have a ‘name/value’ element. This is generally fixed in the TDS schema. The name defaults to the underlying archetype node name but can be overriden at template level.

TDD:

<Problem\_fslash\_Diagnosis archetype\_node\_id="openEHR-EHR-EVALUATION.problem\_diagnosis-uk.v1" type="EVALUATION">  
 <name>  
 <value>Problem/Diagnosis</value>  
 </name>

TDS:

<xs:element name="Problem\_fslash\_Diagnosis">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="name">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="value" type="xs:string" default="Problem/Diagnosis" />  
 <xs:element name="mappings" type="oe:TERM\_MAPPING" minOccurs="0" maxOccurs="unbounded" />  
 <xs:element name="defining\_code" type="oe:CODE\_PHRASE" minOccurs="0" />  
 </xs:sequence>  
 </xs:complexType>

1. Handling DV\_TEXT -> subclassed to DV\_CODED\_TEXT

This can be a little complex and is probably not handled cleanly by the proxy class generator.

Any DV\_TEXT element can be sub-classed to a DV\_CODED\_TEXT type at template or run-time. A good example is in the Problem/Diagnosis archetype where the Diagnosis name is modelled as a DV\_TEXT but needs to carry a SNOMED term as DV\_CODED\_TEXT at runtime.

In the BP-TDD, it appears that ‘null\_flavour’ has been used to try and capture this information.

The correct pattern in the XML is…

<data archetype\_node\_id="at0001" type="ITEM\_TREE">  
 <Problem\_fslash\_Diagnosis archetype\_node\_id="at0002" type="ELEMENT" valueType="DV\_CODED\_TEXT">  
 <name>  
 <value>Problem/Diagnosis</value>  
 </name>   
 <value>  
 <oe:value>Sight deteriorating</oe:value>  
 <defining\_code>  
 <terminology\_id>  
 <value>SNOMEDCT</value>  
 </terminology\_id>  
 <code\_string>22581002</code\_string>  
 </defining\_code>  
 </value>  
 </Problem\_fslash\_Diagnosis>

Note that this will generate a validation error when checking the TDD Instance data against the TDS schema, this must just be ignored.

1. All ENTRY archetype classes (Observation, Evaluation, Instruction, Action, Admin\_entry) have mandatory Language, Encoding and Subject attributes. This should be defined in the schema but is identical for every ENTRY.

TDD:

<Problem\_fslash\_Diagnosis archetype\_node\_id="openEHR-EHR-EVALUATION.problem\_diagnosis-uk.v1" type="EVALUATION">  
 <name>  
 <value>Problem/Diagnosis</value>  
 </name>  
 <language>  
 <terminology\_id>  
 <value>ISO\_639-1</value>  
 </terminology\_id>  
 <code\_string>en</code\_string>  
 </language>  
 <encoding>  
 <terminology\_id>  
 <value>IANA\_character-sets</value>  
 </terminology\_id>  
 <code\_string>UTF-8</code\_string>  
 </encoding>  
 <subject xsi:type="oe:PARTY\_SELF"/>

TDS:

1. ‘data’ elements inside an Event were missing the ItemType XML-attribute

TDD:

<Corrected\_visual\_acuity archetype\_node\_id="openEHR-EHR-OBSERVATION.visual\_acuity.v1" type="OBSERVATION">  
 <name>  
 <value>Corrected visual acuity</value>  
 </name>  
 <language>  
 <terminology\_id>  
 <value>ISO\_639-1</value>  
 </terminology\_id>  
 <code\_string>en</code\_string>  
 </language>  
 <encoding>  
 <terminology\_id>  
 <value>IANA\_character-sets</value>  
 </terminology\_id>  
 <code\_string>UTF-8</code\_string>  
 </encoding>  
 <subject xsi:type="oe:PARTY\_SELF"/>  
 <data archetype\_node\_id="at0001">  
 <origin>  
 <value xmlns="http://schemas.openehr.org/v1">20130112T103000</value>  
 </origin>  
 <Any\_event archetype\_node\_id="at0134" type="POINT\_EVENT">  
 <name>  
 <value>Any event</value>  
 </name>  
 <time>  
 <oe:value>20130112T103000</oe:value>  
 </time>  
 <data archetype\_node\_id="at0003" type="ITEM\_TREE">  
 <Test\_Name archetype\_node\_id="at0138" type="ELEMENT" valueType="DV\_CODED\_TEXT">  
 <name>  
 <value>Test Name</value>  
 </name>

1. Similar to the issue in (2) EVENT nodes inside OBSERVATION archetypes need a name/value element – again these should be ‘fixed’ in the TDS schema.
2. The OBSERVATION class has an origin attribute but you also need to populate a mandatory ‘time’ attribute for each EVENT within the OBSERVATION. In our use case, there is only ever a single EVENT and its time is identical to that held in origin.

<data archetype\_node\_id="at0001">  
 <origin>  
 <value xmlns="http://schemas.openehr.org/v1">20130112T103000</value>  
 </origin>  
 <Any\_event archetype\_node\_id="at0134" type="POINT\_EVENT">  
 <name>  
 <value>Any event</value>  
 </name>  
 <time>  
 <oe:value>20130112T103000</oe:value>  
 </time>

1. For DV\_MULTIMEDIA types, the mediatype is being recorded but the XML is not quite correct. We need to carry some other metadata about the terminology, which in this case is fixed to ‘openehr’.

TDD:

<media\_type xmlns="http://schemas.openehr.org/v1">  
 <terminology\_id>  
 <value>openehr</value>  
 </terminology\_id>  
 <code\_string>image/jpeg</code\_string>  
 </media\_type>

1. Where it is necessary to model e.g. ‘Right Eye’ and ‘Left eye’, this has not been done correctly.

It is a somewhat confusing structure in the schema.

First we have a CLUSTER node which will have an XML tag name <Right\_eye> but being an archetyped node, also needs a proper name/value element in the XML

e.g.

<Right\_eye archetype\_node\_id="at0053" type="CLUSTER">  
 <name>  
 <value>Right eye</value>  
 </name>

However this is just used as a human label and container, with the actual location semantics carried in a child ‘Eye’ or ‘Eye’ examined element.

The Eye element is a DV\_CODED\_TEXT element. Sine it is an archetype node it needs to carry the name/value as before, and then the real payload is carried in the Eye/value. In the openEHR template, I have (or should have) set the correct coded text as a default value. The TDS schema generator picks this up and adds the correct terminologyID, and definingCode as ‘fixed’ attributes in the schema. It does not add the value i.e. the term rubric “Right eye”. I am not sure if this a bug or if there is some technical reason which prevents this happening.

A correct XML fragment is

<data archetype\_node\_id="at0003" type="ITEM\_TREE">  
 <Right\_eye archetype\_node\_id="at0053" type="CLUSTER">  
 <name>  
 <value>Right eye</value>  
 </name>  
 <!-- Need to add the Eye element -->  
 <Eye>  
 <name>  
 <value>Eye</value>  
 </name>  
 <value>  
 <value>Right eye</value> <!-- Default value is not specified in the schema -->  
 <defining\_code>  
 <terminology\_id>  
 <value>local</value>  
 </terminology\_id>  
 <code\_string>at0013</code\_string>  
 </defining\_code>  
 </value>  
 </Eye>

The key issue here is how you can pick up the correct default value, since it is not in the TDS schema.

1. We need to add the ‘Test Name’ element to the Visual Acuity archetypes. Thisis a similar problem to the ‘Eye’ issue above. The kind of Visual acuity measure uncorrected, best corrected etc is carried by the XML tag and ‘name’ of the parent archetype but the actual semantics are fixed in the ‘Test Name element, which has a fixed list of test names. I will have selected one as a default to match the header.

e.g.

<Corrected\_visual\_acuity archetype\_node\_id="openEHR-EHR-OBSERVATION.visual\_acuity.v1" type="OBSERVATION">  
 <name>  
 <value>Corrected visual acuity</value>  
 </name>

… (omitted for brevity)  
 <data archetype\_node\_id="at0003" type="ITEM\_TREE">  
 <Test\_Name archetype\_node\_id="at0138" type="ELEMENT" valueType="DV\_CODED\_TEXT">  
 <name>  
 <value>Test Name</value>  
 </name>  
 <value>  
 <value></value>  
 <defining\_code>  
 <terminology\_id>  
 <value>local</value>  
 </terminology\_id>  
 <code\_string>at0136</code\_string>  
 </defining\_code>  
 </value>  
 </Test\_Name>

I should probably have made the Test Name and Eye nodes mandatory in the template to make sure that this was clearly expressed in the schema. I will do that now and send the updated schema to GitHub.

That’s it. Once these fixes are in place, the TDD instance validates correctly (other than the DV\_TEXT/DV\_CODED\_TEXT issue in (4), can be transformed to canonical openEHR XML and displayed with our generic openEHR xml->html viewer. I did have to make some changes to aspects of the namespacing. I suspect that might need to be a bit of a final fiddle for you to automate correctly.

Ian