**Results - Proposed changes to EDXL-HAVE-v2.0.xsd**

**Note: Results in green represent the results when changes were made to the main schema.**

**Replace**

<xs:import namespace="urn:oasis:names:tc:emergency:edxl:ciq:1.0:xpil" schemaLocation="./rim/edxl\_xPIL.xsd"/>

<xs:import namespace="urn:oasis:names:tc:emergency:edxl:ciq:1.0:xal" schemaLocation="./rim/edxl\_xAL.xsd"/>

**With**

<xs:import namespace="urn:oasis:names:tc:emergency:edxl:ciq:1.0:xpil" schemaLocation="./ImportedSchema/edxl-xPIL.xsd"/>

<xs:import namespace="urn:oasis:names:tc:emergency:edxl:ciq:1.0:xal" schemaLocation="./ImportedSchema/edxl-xAL.xsd"/>

**Result:** It was necessary in order for the package of included schema to validate for these changes to be made to the file ‘edxl-ct-v1.0-wd06.xsd’ as well as the ‘edxl-have-v2.0.xsd’ file. This was only an interim measure to be sure that ‘edxl-have-v2.0.xsd’ will indeed validate with this change, and it does.

So, the RIM SC will have to take up the issue of this change to the edxl-ct-v1.0 committee specification draft along with any other changes that now need to be made. I would still prefer to change that document of record in the <https://doc.oasis-open.org/emergency> directory rather than making a new wd07 just to have ‘edxl-have-v2.0.xsd’ validate.

Because this change is due to the change in camel case naming that the TC adopted several years ago, this is a good idea in keeping with the effort to normalize the whole edxl datamodel.

Note also that the rim directory name has been changed to ImportedSchema for all imported schema locations as decided by the TC April 4, 2017.

**Interim Result:** Invalid, error in associated file ‘edxl-xNL’ 'grNameKey' is already declared. This error was repeated for all elements. However, by deleting ‘xlink-2003-12-31’ and replacing with ‘xlink.xsd’ that problem was resolved. Success. This replacement is a change that is likely coming from the RIM SC as a recommendation.

**Reason:** replacements use Pascal Camel Case Capitalization Scheme the TC adopted. The correct files have/had to be added to the ‘ ./rim’ directory (a directory name that needs to be changed and standardized across the EDXL family that uses supporting elements from edxl-ct, edxl-ciq and edxl-gsf.)

(Note: The recommendation from the RIM SC was to have a structure consisting of the main specification folder named ‘schema’, including in it a folder/directory named “ImportedSchema” for the imported schema for edxl-ct (Common Types), edxl-gsf (Geographic Simple Features Profile), edxl-ciq (Customer Information Quality Profiles) and edxl-ext (Extension) and a folder/directory named “examples” for examples,. We’re just standardizing this structure across the whole suite/family of EDXL.

**Replace** (15 occurences)

<xs:element name="comment" type="FreeTextType" minOccurs="0">

**With**

<xs:element name="remarks" type="ct:RemarksType" minOccurs="0"/>

**Result:** These replacements caused no further changes.

**Result:** Error message: (15 occurrences some different cardinalities)

'have:ct-RemarksType' must refer to an existing simple or complex type.

However I noticed that the namespace prefix I actually needed to use was ‘edxl-ct:RemarksType’:

<xs:element name="remarks" type="edxl-ct:RemarksType" minOccurs="0"/>

It succeeded! (It validated.)

**Result:** We decided in the HAVE SC to leave this as is.

**Replace** (following lesson learned in preceding item) I decided to leave FreeTextType in because the precedent of explicitly providing for alternate languages in the creation and delivery of EDXL messages is something I think is worth initiating and continuing. However, I propose to eliminate LmitedString since it occurs in just this one specification and edxl-ct already has it covered in the restriction of edxl-ct:EDXLString

<xs:complexType name="FreeTextType">

<xs:sequence>

<xs:element name="defaultText" type="LimitedString">

<xs:annotation>

<xs:documentation>The text value that uses the message default language (defined at in the HAVE message defaultLanguage attribute).</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="alternateText" type="AlternateTextType" minOccurs="0" maxOccurs="unbounded">

<xs:annotation>

<xs:documentation>Alternate language representation.</xs:documentation>

</xs:annotation>

</xs:element>

</xs:sequence>

</xs:complexType>

<xs:complexType name="AlternateTextType">

<xs:simpleContent>

<xs:extension base="LimitedString">

<xs:attribute name="language" type="xs:string" use="required">

<xs:annotation>

<xs:documentation>Language code for the text in this element. Code MUST comply with RFC3066. </xs:documentation>

</xs:annotation>

</xs:attribute>

</xs:extension>

</xs:simpleContent>

</xs:complexType>

**With**

<xs:complexType name="FreeTextType">

<xs:sequence>

<xs:element name="defaultText" type="edxl-ct:EDXLStringType">

<xs:annotation>

<xs:documentation>The text value that uses the message default language (defined at in the HAVE message defaultLanguage attribute).</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="alternateText" type="AlternateTextType" minOccurs="0" maxOccurs="unbounded">

<xs:annotation>

<xs:documentation>Alternate language representation.</xs:documentation>

</xs:annotation>

</xs:element>

</xs:sequence>

</xs:complexType>

<xs:complexType name="AlternateTextType">

<xs:simpleContent>

<xs:extension base=" edxl-ct:EDXLStringType ">

<xs:attribute name="language" type="xs:string" use="required">

<xs:annotation>

<xs:documentation>Language code for the text in this element. Code MUST comply with RFC3066. </xs:documentation>

</xs:annotation>

</xs:attribute>

</xs:extension>

</xs:simpleContent>

</xs:complexType>

The substitution above validated with no problem.

We may want to eliminate “LimitedString” but for now I have commented it out as shown below:

</xs:simpleType>

<!--xs:simpleType name="LimitedString">

<xs:annotation>

<xs:documentation>Text block for preserving whitespace but limiting length to 1024 characters.</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:whiteSpace value="preserve"/>

<xs:maxLength value="1024"/>

</xs:restriction>

</xs:simpleType-->

<xs:complexType

Why? Because “EDXLStringType, from ‘edxl-ct-v1.0-wd06’ and the current official version of edxl-ct-v1.0.xsd seems to handle this consideration:

<xs:simpleType name="EDXLStringType">

<xs:restriction base="xs:token">

<xs:maxLength value="1023"/>

<xs:minLength value="1"/>

</xs:restriction>

I wouldn’t characterize any of these changes as major, but they are also not “Non-Material Changes” as OASIS recognizes such. Anyone who had implemented the current version would have their implementation broken by these changes, but would not have a difficult time adjusting some element names to come into conformance.