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

**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="./rim/edxl-xPIL.xsd"/>

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

**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.

**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.)

**Follow-up:** For filepath consistency, I propose this directory structure:

EDXL-Name-VersionNumber **/** schema **/** supportingElements ( ct/ciq/gsf/ext files may just share ‘supportingElements’ or be broken out into their own subdirectories).

Actually EDXL-DE-v2.0, EDXL-TEP-v1.1, EDXL-RM-v1.0 and EDXL-SitRep-v1.0 follow this scheme with differing directory names now. So does EDXL-HAVE-v2.0, I just prefer ‘supportingElements’ to ‘rim’ and I think we need a standard directory structure. This required making the same changes to the schema location filepaths in ‘edxl-have-v2.0-alt.xsd’.

**Replace** (15 occurences)

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

**With**

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

**Result:** Error message: (15 occurences 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!

**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 continuing. However, I also wanted to eliminate LmitedString in just one specifications when 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

Because “EDXLStringType, from ‘edxl-ct-v1.0-wd06’ seems to handle this consideration:

<xs:simpleType name="EDXLStringType">

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

 <xs:maxLength value="1023"/>

 <xs:minLength value="1"/>

 </xs:restriction>