**Issues in Harmonizing the EDXL Family**:

1. **EDXL-DE-v2.0**: When I loaded the file: ‘edxl-de-v2.0-wd11.xsd’ into XML Spy it failed to validate, with the following in the Messages window:

The schema doesn't appear to be valid by itself (as a part of another schema, it might still be OK).

File C:\OASIS\OASIS EM TC\EDXL-RIM\EDXL-DataModel\EDXL-RIM-Model-EA-1\EDXL-DE\v2.0\CS02\schema\other-supporting-schema\EDXLCT\_wd06\xlink-2003-12-31.xsd has errors.

'xlink:role' is already declared.

'xlink:arcrole' is already declared.

'xlink:title' is already declared.

'xlink:show' is already declared.

'xlink:actuate' is already declared.

'xlink:label' is already declared.

'xlink:from' is already declared.

'xlink:to' is already declared.

'xlink:href' is already declared.

I noticed that file: ‘xlink.xsd’ is found in the ‘other-supporting-schema’ folder and that the file: ‘xlink-2003-12-31.xsd’ is located in the ‘EDXLCT\_wd06’ inside the ‘other-supporting-schema’. However, in the file: ‘edxl-de-v2.0-wd11.xsd’ only the file: ‘xlink.xsd’ is specifically imported:

<xs:import namespace="http://www.w3.org/1999/xlink" schemaLocation="./other-supporting-schema/xlink.xsd"/>

From which I would expect ‘edxl-de-v2.0-wd11.xsd’ to validate against ‘xlink.xsd’ not ‘xlink-2003-12-31.xsd’ so I deleted ‘xlink-2003-12-31.xsd’ and when I reloaded ‘edxl-de-v2.0-wd11.xsd’ it validated with no problem.

I would like to have file ‘xlink-2003-12-31.xsd’ removed from <https://docs.oasis-open.org/emergency/edxl-de/v2.0/schema/other-supporting-schema/EDXLCT_wd06> by asking the EM TC to request a Special Majority Vote to approve EDXL-DE-v2.0 as a Committee Specification with Non-Substantive Changes. I would also like to declare this filename EDXL-DE-v2.0.xsd, without a working draft number, e.g.-wd12.

**Jeff, do you think there will be any impact from this change on the examples? I didn’t want to take the time to check the examples. Also, though I don’t recall why, I inadvertently used edxl-de-v2.0-csprd03 for the Enterprise Architect models wd01, wd02 and wd03, but I don’t think there are any significant changes to the datamodel.**

**I have since redone the model for edxl-de-v2.0-cs02 in EDXL-DE-v2.0-RIM-EA-wd04 which includes the removal of ‘xlink-2003-12-31.xsd’. We can always revert to wd03.—Later note: See Issue 7.**

**Spread throughout our specifications we use ‘xlink-2003-12-31.xsd’, ‘xLink.xsd’, ‘xlinks.xsd’ and my xlink-of-choice vanilla ‘xlink.xsd’, but at some point we need to figure this out.**

1. **EDXL-HAVE-v2.0 vs. EDXL-SitRep-v1.0**: We have a possible conflict that the EM RIM SC should step in to mediate before EDXL-HAVE-v2.0 or EDXL-SitRep-v1.0 proceed further. This issue is that we have two differently named XML elements which serve the same purpose

EDXL-HAVE-v2.0 uses a combination of specially defined strings of base type xs;string:

<xs:element name="comment" type="FreeTextType" minOccurs="0" maxOccurs="1">

<xs:annotation>

<xs:documentation>General comment/summary on the trauma center status</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="comment" is used many times in EDXL-HAVE-v2.0.xsd for comment in relation to various other elements. This element is constrained by limitations of the type="FreeTextType" below:

<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>

This type, in turn, is constrained by the restriction of “Limited String”:

<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>

EDXL-SitRep-v1.0 uses a similar, but much simpler and less limited construction:

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

This element is a part of the Common Types of EDXL declared at the beginning of EDXL-SitRep-v1.0.xsd:

xmlns:ct="urn:oasis:names:tc:emergency:edxl:ct:1.0"

And explicitly imported from the clearly specified directory:

<xs:import namespace="urn:oasis:names:tc:emergency:edxl:ct:1.0" schemaLocation="./supportingElements/edxl-ct-v1.0-csd06/edxl-ct-v1.0.xsd"/>

In contrast to the elaborately constrained definition in EDXL-HAVE-v2.0 the declaration in edxl-ct-v1.0.xsd and used in EDXL-SitRep-v1.0 is a remarkably simple definition:

<xs:simpleType name="RemarksType">

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

</xs:simpleType>

**Jeff, I obviously favor SitRep’s use of remarks over HAVE’s use of comment(s) and propose that we recommend that it be adopted for EDXL-HAVE-v2.0 without the unnecessary restriction to 1024 characters. Like I’ve said several times, I’m not an expert in XML Schema, but I’m getting there with every time I have to get down in these weeds—sheeesh! And I’m likely to be the one stuck with making these changes, though they should be straightforward enough for me to be able to do it without screwing up. Your thoughts?**

1. **File Directory Structure and Zip packages for EDXL specifications**: At present we have several different, but very similar file directory structure schemes for our various EDXL specifications and I think they should be reworked to

* use the same nomenclature
  + for where the specification schema are located,
  + where their supporting schema are located and
  + how those directories are named.
* use the latest official versions on the supporting schema, especially important with regard to Issue 1 where, for instance
  + EDXL-HAVE-v2.0 uses file: ‘xlink-2003-12-31.xsd’ while EDXL-SitRep-v1.0 uses file: ‘xlink.xsd’).
  + EDXL-TEP-v1.1 also uses file: ‘xlink-2003-12-31.xsd’.

However, this is just one case of several similar dissonances, in particular for ‘edxl-ct-v1.0-csd05 and ‘edxl-ct-v1.0-csd06’. Most of these are due to the fact that specifications started later typically use later versions of the supporting schema.

* use the same uniform practices for populating the Zip packages that are assembled for standards, committee specification public review drafts and committee specifications
* use the same file directory structures and naming practices for examples across the individual EDXL specifications.

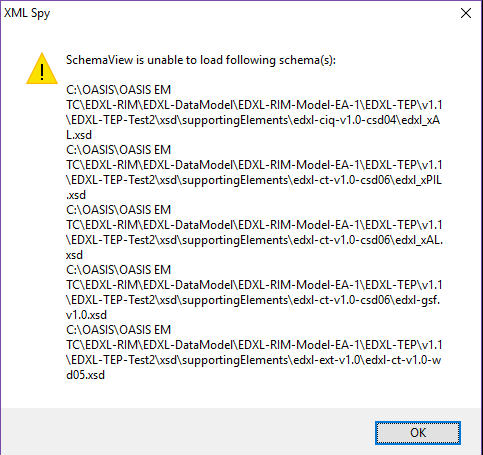
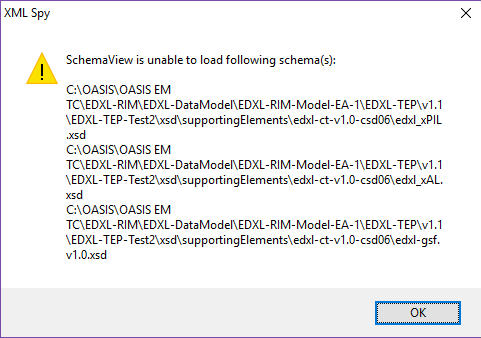
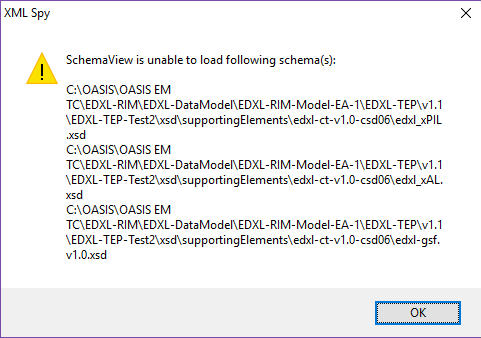
The time of compiling this issues list (mid-February 2017) is advantageous for this because the RIM SC is working on a unified Framework/Toolkit for the entire overarching EDXL family or suite of specifications. The RIM SC should revisit these practice issues to determine how to ensure that the existing EDXL OASIS Standards and EM TC Committee Specifications are updated to use the latest versions of the supporting specifications in the least disruptive way in order to keep our EDXL specifications current.

This kind of uniformity will greatly simplify the overall implementation of the EDXL family or suite of specifications.

**I think most of these suggestions are just common sense, but I am obviously biased since they are my recommendations. However, we have such a lot of issues that need to be addressed at the same time that I think it behooves us to get EDXL properly aligned in one large effort even though it may be a little more painful than to continue with haphazard bandages. Your thoughts?**

1. **EDXL-SitRep-v1.0 and EDXL-TEP-v1.1**:

EDXL-TEP-v1.1 raw from docs.oasis-open.org: It opened with 3 warning windows citing schema that XMLSpy was unable to load. However I ignored that and noted the highlighted portions of top of the schema below these images.

<?xml version="1.0" encoding="UTF-8"?>

<!--

Snip

-->

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"

xmlns:ct="urn:oasis:names:tc:emergency:edxl:ct:1.0"

xmlns:xal="urn:oasis:names:tc:emergency:edxl:ciq:1.0:xal"

xmlns:tep="urn:oasis:names:tc:emergency:EDXL:TEP:1.1"

xmlns:ns1="urn:oasis:names:tc:emergency:edxl:tep:ct:1.0"

xmlns:ext="urn:oasis:names:tc:emergency:edxl:extension:1.0"

targetNamespace="urn:oasis:names:tc:emergency:EDXL:TEP:1.1" elementFormDefault="qualified">

<xsd:import namespace="urn:oasis:names:tc:emergency:edxl:ct:1.0"

schemaLocation="./supportingElements/edxl-ct-v1.0-csd06/edxl-ct-v1.0.xsd"/>

<xsd:import namespace="urn:oasis:names:tc:emergency:edxl:ciq:1.0:xal"

schemaLocation="./supportingElements/edxl-ciq-v1.0-csd04/edxl\_xAL.xsd"/>

<xsd:import namespace="urn:oasis:names:tc:emergency:edxl:extension:1.0"

schemaLocation="./supportingElements/edxl-ext-v1.0/edxl-ext-v1.0.xsd"/>

From previous experience with EDXL-SitRep-v1.0, it was apparent on examination that the highlighted schema import pointed to a schema version prior to our decision to adopt Pascal CamelCase naming and design rules.

Below is the version of EDXL-TEP-v1.1 that eventually validated with XMLSpy supplying namespace abbreviations for two namespaces on loading, then edited until it validated with no changes to the supportingElements file directory structure and contents. This editing builds on what was learned in the last stages of working on the Committee Specification of EDXL-SitRep-v1.0.Note also that there is an ongoing problem with the extension element shown and discussed in **Issue 6**.

<?xml version="1.0" encoding="UTF-8"?>

<!--

Snip

-->

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"

xmlns:ct="urn:oasis:names:tc:emergency:edxl:ct:1.0"

xmlns:xal="urn:oasis:names:tc:emergency:edxl:ciq:1.0:xal"

xmlns:tep="urn:oasis:names:tc:emergency:EDXL:TEP:1.1"

xmlns:ns1="urn:oasis:names:tc:emergency:edxl:tep:ct:1.0"

xmlns:ext="urn:oasis:names:tc:emergency:edxl:extension:1.0"

xmlns:ns2="urn:oasis:names:tc:emergency:edxl:ciq:1.0:xpil"

xmlns:ns3="urn:oasis:names:tc:emergency:edxl:gsf:1.0"

targetNamespace="urn:oasis:names:tc:emergency:EDXL:TEP:1.1" elementFormDefault="qualified">

<xsd:import namespace="urn:oasis:names:tc:emergency:edxl:ct:1.0" schemaLocation="./supportingElements/edxl-ct-v1.0-csd06/edxl-ct-v1.0.xsd"/>

<!--xsd:import namespace="urn:oasis:names:tc:emergency:edxl:ciq:1.0:xal"

schemaLocation="./supportingElements/edxl-ciq-v1.0-csd04/edxl\_xAL.xsd"/-->

<xsd:import namespace="urn:oasis:names:tc:emergency:edxl:extension:1.0"

schemaLocation="./supportingElements/edxl-ext-v1.0/edxl-ext-v1.0.xsd"/>

<xsd:import namespace="urn:oasis:names:tc:emergency:edxl:ciq:1.0:xpil"

schemaLocation="./supportingElements/edxl-ciq-v1.0-csd04/edxl-xPIL.xsd"/>

<xsd:import namespace="urn:oasis:names:tc:emergency:edxl:ciq:1.0:xal"

schemaLocation="./supportingElements/edxl-ciq-v1.0-csd04/edxl-xAL.xsd"/>

<xsd:import namespace="urn:oasis:names:tc:emergency:edxl:gsf:1.0"

schemaLocation="./supportingElements/edxl-gsf-v1.0-csd02/edxl-gsf-v1.0.xsd"/>

EDXL-SitRep-v1.0 below shows a similar correction on which the changes above were based:

<?xml version="1.0" encoding="UTF-8"?>

<!--

(Snip)

-->

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"

xmlns:ct="urn:oasis:names:tc:emergency:edxl:ct:1.0"

xmlns:xpil="urn:oasis:names:tc:emergency:edxl:ciq:1.0:xpil"

xmlns:xal="urn:oasis:names:tc:emergency:edxl:ciq:1.0:xal"

xmlns:edxl-gsf="urn:oasis:names:tc:emergency:edxl:gsf:1.0"

targetNamespace="urn:oasis:names:tc:emergency:edxl:ct:1.0" elementFormDefault="qualified" attributeFormDefault="qualified">

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

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

<!--xs:import namespace="urn:oasis:names:tc:emergency:edxl:gsf:1.0" schemaLocation="edxl-gsf.v1.0.xsd"/-->

<xs:import namespace="urn:oasis:names:tc:emergency:edxl:ciq:1.0:xpil"

schemaLocation="../edxl-ciq-v1.0-csd04/edxl-xPIL.xsd"/>

<xs:import namespace="urn:oasis:names:tc:emergency:edxl:ciq:1.0:xal"

schemaLocation="../edxl-ciq-v1.0-csd04/edxl-xAL.xsd"/>

<xs:import namespace="urn:oasis:names:tc:emergency:edxl:gsf:1.0"

schemaLocation="../edxl-gsf-v1.0-csd02/edxl-gsf-v1.0.xsd"/>

<xs:annotation>

Above, see what was done for EDXL-SitRep-v1.0 for it to validate by commenting out the incorrect declared schema locations and filenames for edxl-xAL.xsd, edxl-xPIL.xsd and the locations for all three and replacing those faulty import statements with correct ones.

EDXL-TEP-v1.1 was approved without having these things noticed, but we now we can correct it. Also, I propose that we repair the schema of EDXL-SitRep by removing the commented out matter and do the same for EDXL-TEP-v1.1. There are a few other items to look at in EDXL-TEP-v1.0 in following issues.

I believe we should recommend that the EM TC request TC Admin conduct Special Majority Ballots for Approval of these edited specifications as Committee Specifications With Non-Substantive Changes. This holds for EDXL-DE-v2.0, EDXL-SitRep-v1.0, and EDXL-TEP-v1.1. We should consider short public reviews that we can use to bring attention to our work. EDXL-HAVE-v2.0 has the issues noted above but will need to conduct a whole new round of public reviews unless we opt to live with it in its current state. I advise against that.

This way we can get the necessary changes in place in the schema, and harmonize the naming/location of xsd schema, supporting schema and example directories so we can get the latest versions of edxl-ciq, edxl-gsf and edxl-ct implemented uniformly across the EDXL family/suite.

1. **EDXL-HAVE-v2.0 and EDXL-TEP-v1.1**: The same problem outlined in issue 1. Both of these specifications use ‘xlink-2003-12-31.xsd’. Though both now validate (with the changes noted above and below, and I would like to replace those with ‘xlink.xsd’.
2. **EDXL-TEP-v1.1**: I found that yet another faulty expression in EDXL-TEP-v1.1.xsd. After all the other changes, it did not validate in XMLSpy, producing the following message in the Message window:

File C:\OASIS\OASIS EM TC\EDXL-RIM\EDXL-DataModel\EDXL-RIM-Model-EA-1\EDXL-TEP\v1.1\xsd\edxl-tep-v1.1.xsd is not valid.

'ext:ExtensionType' must refer to an existing simple or complex type.

Error location: xsd:schema /xsd:element /xsd:complexType /xsd:sequence /xsd:element / @type

Details

src-resolve: Value 'ext:ExtensionType' of attribute 'type' doesn't resolve to a type definition

The offending expression:

<xsd:element name="TEPMessage">

<xsd:annotation>

<xsd:documentation>Group of elements used to uniquely identify a TEP message and its source.</xsd:documentation>

</xsd:annotation>

<xsd:complexType>

<xsd:sequence>

<xsd:element name="messageID" type="ct:EDXLStringType"/>

<xsd:element name="systemID" type="ct:EDXLStringType" minOccurs="0"/>

<xsd:element name="patient" type="tep:PatientType"/>

<xsd:element name="extension" type="ext:ExtensionType" minOccurs="0" maxOccurs="unbounded"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

A similar expression from EDXL-SitRep-v1.0.xsd is commented out, and replaced:

<!--xs:element name="extension" type="ext:ExtensionType"

minOccurs="0" maxOccurs="unbounded"/-->

<xs:element ref="ext:extension" minOccurs="0" maxOccurs="unbounded"/>

and it validated, so I tried that and it, too, validated . So my recommendation is that we comment out the incorrect or invalid expression and replace it as above or else simply replace the incorrect expression with the correct one.

1. **‘xlink-2003-12-31.xsd’ et al revisited**: While working on trying to get EDXL-HAVE-v1.0 to validate in order to produce a viable JSON Schema for it, I discovered that ‘xlink-2003-12-31” has namespace declaration that throws parsers off kilter since our later specifications which do validate cite the second example below:

<schema xmlns:xlink="http://www.w3.org/1999/xlink1" xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.w3.org/1999/xlink1" elementFormDefault="qualified" attributeFormDefault="qualified">

From EDXL-SitRep-v1.0:

xmlns:xlink=<http://www.w3.org/1999/xlink> …

…

<xs:import namespace="http://www.w3.org/1999/xlink"

schemaLocation="./other-supporting-schema/xlink.xsd"/>

/xlink1” appears to be the culprit. I thought it should be /xlink” as the error message indicated was “expected” so I made the correction to test the hypothesis and the error disappeared on reload, but unfortunately, this is the product of a TC that is closed and we can’t edit it. I don’t know how we can repair this. This may be the root of what is causing this file to tangle up our work everywhere it turns up. But there are probably other concerns since it was, in essence, borrowed from xbrl.org by the CIQ TC.

Later, I realized that as long as we remove ‘xlink-2003-12-31” wherever we can do so, **provided we** **achieve validation without getting more than a warning note that it couldn’t be loaded**, we should be all right with this. However, several EDXL specifications import CIQ and edxl-ciq namespaces linked to files at specific locations and date from before our change of Camel Case naming practice, which in their turn import this same darn ‘xlink-2003-12-31’ file. See the bulleted item below. We run into this problem with ‘edxl-gsf-v1.0.xsd’ and ‘edxl-gsf-base.xsd’.

**Question: Do we need to put in a note that these schema are valid and users should ignore the warning messages?**

That leaves us with no choice but to include the file in the directory that the imported namespace declaration points to. The CIQ or edxl-ciq won’t validate properly if it isn’t present, even if it is not being used directly. And that prevents our main specification from validating. It’s a nice little tangle.

I have discovered other problems such different filenames in the CIQ specification set between different EDXL specifications. For instance

* EDXL-SitRep-v1.0 uses the following files with dashes between parts of the filenames:
  + edxl-xAL.xsd. edxl-xNL.xsd, edxl-xPIL, but
* EDXL-DE-v2.0 uses virtually the same files with underscores between parts of the filenames:
  + edxl\_xAL.xsd, edxl\_xAL.xsd, edxl\_xPIL.xsd.

SitRep was started later and used a later updated version of the same file using the naming and design rules for Capitalization that we adopted after EDXL-DE-v2.0 was started.

EDXL-TEP-v1.0 is consistent with EDXL-DE-v2.0—wrong capitalization in three edxl-ciq files but now validates.

EDXL-TEP-v1.1 is consistent with EDXL-SitRep-v1.0

EDXL-HAVE-v1.0 probably can’t be repaired—uses geo-oasis.xsd, w/ errors in namespace

EDXL-HAVE-v2.0 is currently consistent with EDXL-DE-v2.0—wrong capitalization in three edxl-ciq files but needs to be reworked so it is a good time to address these issues.

EDXL-RM-v1.0 doesn’t use those three files, but does use geo-oasis.xsd and properly uses gml-oasis.xsd located locally and all message type schema validate