EML - Bug #484

eml-attribute changes needed

05/01/2002 04:52 PM - Matt Jones

Status:	Resolved	**Start date:**	05/01/2002
Priority:	Immediate	**Due date:**	
Assignee:	Chad Berkley	**% Done:**	0%
Category:	eml - general bugs	**Estimated time:**	0.00 hour
Target version:	EML2.0.0rc1	**Spent time:**	0.00 hour
Bugzilla-Id:	484		

Description

Changes as decided upon at the Sevilleta EML meeting, April 24-25, 2002:
Responsible: Chad, Dan, David

1) rename dataType to storageType -- minimally suggest use fo XML Schema DT as base for storageType, add attribute "typeSystem" for referencing the system used
2) add "unitSystem" attribute to w/ default of STMML, make "unit" required with default of "dimensionless"
3) add measurementScale element for documenting scale (ordinal, ratio, interval, nominal). We discussed whether this was implied by dataType/unit, but decided to add it, even though it probably is somehow implied by the dimension of the measurement
4) add "accuracy" element, use FGDC "dataQuality" model for it
5) how do we express precision & accuracy -- need to be explicit in our field documentation
6) generally resolve the storageType/dimension/unit/measurementScale morass
7) add explanation/reason field to the "missingValueCode" so that people can explain what "-9999" means in their data set
8) move "textDomain" and "enumeratedDomain" up so that they are siblings of numeric domain, remove the choice
9) add "enforced" attribute to enumeratedDomain element with allowable values "yes", "no", defaults to "yes"
10) enumeratedDomain: need externally referenced codeset, reference to dataTable entity that contains codes (2 columns). below is a content model from my notes that doesn't make a lot of sense to me right now. Corinna says that the sequence after enumeratedDomain should be optional,repeatable
enumeratedDomain |- list | |- code | |- def | |- source | |- entity | |- entityID | |- codeColumn | |- codeDefinitionColumn | |- externalSource

History

#1 - 05/03/2002 10:59 AM - Chad Berkley

Finished with 1-9. Have implemented some changes in the dtd but not all. I don't fully understand nuber 10 so I will wait for clarrification before attempting it. Note that three of the FGDC elements that referenced other FGDC constructs reference eml or XMLSchema constructs in my implementation. The three are: citationInformation (I used eml-literature), timePeriodInformation (I used eml-coverage/temporalCoverage) and processContact (I used eml-party/responsibleParty). If anyone has a problem with this please let me know.

#2 - 05/10/2002 01:52 PM - Chad Berkley

I removed most of the FGDC dataquality stuff and just left the attribute accuracy parts. It seemed much more logical to just use a subset since a lot of the elements in FGDC:DataQuality are represented elsewhere in EML.

#3 - 05/28/2002 12:57 PM - Chad Berkley

in trying to figure out how to integrate the STMML stuff into attribute to handle units, I have thought about several different methods for including structured, parseable unit information in eml-attribute. The consensus that I have come to is that we need a unit dictionary marked up in compliance with STMML. this dictionary will be an extensible, structured list of all units that can be used in the 'unit' field of eml-attribute. The list will be

referenced within eml-attribute via an http URI that links the unit field to the unit type in the dictionary. As with namespace URIs, the referenced link need not be an active web service, rather it is just a unique, parseable identifier that allows the attributes unit to be linked into the dictionary. Of course, this will only be parseable by an external processor, which we hoped to avoid in the entire packaging scheme, however this seems to be a different type of problem than the packaging problem.

The scheme is defined below:
STMML unit dictionary:

```
<?xml version="1.0"?>
<eml:unitList xmlns:eml="./eml-unit.xsd">
<!-- ========================================================================
-->
<!-- ========================= fundamental types ============================
-->
<!-- ========================================================================
-->
<eml:unitType id="length" name="length">
<eml:dimension name="length" />
</eml:unitType>

&lt;eml:unitType id="time" name="time"&gt;
    &lt;eml:dimension name="time" /&gt;
  &lt;/eml:unitType&gt;

&lt;eml:unitType id="dimensionless" name="dimensionless"&gt;
    &lt;eml:dimension name="dimensionless" /&gt;
  &lt;/eml:unitType&gt;

<!-- ========================================================================
-->
<!-- ========================== derived types ===============================
-->
<!-- ========================================================================
-->
<eml:unitType id="acceleration" name="acceleration">
<eml:dimension name="length" />
<eml:dimension name="time" power="-2" />
</eml:unitType>

<!-- ========================================================================
-->
<!-- ======================== fundamental SI units ==========================
-->
<!-- ========================================================================
-->
<eml:unit id="second" name="second" unitType="time">
<eml:description>The SI unit of time</eml:description>
</eml:unit>

&lt;eml:unit id="meter" name="meter" unitType="length" abbreviation="m"&gt;
    &lt;eml:description&gt;The SI unit of length&lt;/eml:description&gt;
  &lt;/eml:unit&gt;

&lt;eml:unit id="kg" name="nameless" unitType="dimensionless"
abbreviation="nodim">
    &lt;eml:description&gt;A fictitious parent for dimensionless
units&lt;/eml:description&gt;
  &lt;/eml:unit&gt;

<!-- ========================================================================
-->
<!-- ======================== derived SI units ==============================
-->
<!-- ========================================================================
-->
<eml:unit id="newton" name="newton" unitType="force">
<eml:description>The SI unit of force</eml:description>
</eml:unit>

<!-- multiples of fundamental SI units -->
<eml:unit id="g" name="gram" unitType="mass" parentSI="kg"
multiplierToSI="0.001" abbreviation="g">
<eml:description>0.001 kg.</eml:description>
```

</eml:unit>

```
&lt;eml:unit id="celsius" name="Celsius" parentSI="k" multiplierToSI="1"
constantToSI="273.18"&gt;
    &lt;eml:description&gt;
      &lt;p&gt;A common unit of temperature&lt;/p&gt;
    &lt;/eml:description&gt;
  &lt;/eml:unit&gt;
```

<!-- fundamental non-SI units -->
<eml:unit id="inch" name="inch" parentSI="meter" abbreviation="in"
multiplierToSI="0.0254">
<eml:description>An imperial measure of length</eml:description>
</eml:unit>

<!-- derived non-SI units -->
<eml:unit id="l" name="litre" unitType="volume" parentSI="meterCubed"
abbreviation="l" multiplierToSI="0.001">
<eml:description>Nearly 1 dm**3 This is not quite exact</eml:description>
</eml:unit>

```
&lt;eml:unit id="fahr" name="fahrenheit" parentSI="k" abbreviation="F"
multiplierToSI="0.55555555555555555" constantToSI="-17.777777777777777"&gt;
    &lt;eml:description&gt;An obsolescent unit of temperature still used in popular
meteorology&lt;/eml:description&gt;
  &lt;/eml:unit&gt;
&lt;/eml:unitList&gt;
```

URIs that could be used in the 'unit' field of eml-attribute:
Acceleration in m/s^2
http://ecoinformatics.org/unitDictionary?unitType=acceleration&length=meter&time=second

Acceleration in in/s^2
http://ecoinformatics.org/unitDictionary?unitType=acceleration&length=inch&time=second

Length in meters
http://ecoinformatics.org/unitDictionary?unitType=length&length=meter

Count of species
http://ecoinformatics.org/unitDictionary?unitType=dimensionless

note that the URIs follow the form:
http://ecoinformatics.org/unitDictionary?unitType=<unitType id>&<dimension1
id>=<unitM1 id>&...&<dimensionN id>=<unitM2 id>

with this style URI, the unitType gets defines as well as each dimension in
the unitType gets mapped to a unit. This would be easily parseable and
validateable against the unit Dictionary. In an application, these URIs
would get hashed to a user friendly list of units. For example
m/s^2 -->
http://ecoinformatics.org/unitDictionary?unitType=acceleration&length=meter&time=second

Of course, the unit dictionary will be much more inclusive than the example
that I pasted in above.

Please let me know ASAP if anyone has a problem with this scheme.

#4 - 05/29/2002 01:46 PM - Peter McCartney

Here is the sample attribute file based on today's discussion It defines four
complext types:

AttributeDefinitionType = definition of the content model for an attribute
description. has id &system attributes

AttributeType = contains a choice between a single element of type
AttributeDefinitionType or a reference to a previously defined
attributeDefinition

AttributeListDefinitionType = content model for an attribute list containing a
repeatable element attribute which is of type AttributeType. Has id & system
attributes

AttributeListType = a choice between a single element of type
AttributeListDefinitionType or a reference to a previously defined

AttributeListDefinition.

Usage: When defining an entity, you would create a local element of type AttributeListType. Insert either a <references> pointing to a previously defined element of AttributeListType or an <attributeListDefinition> element into which you insert one or more <attribute> elements. these in turn may each contain either an <attributeDefinition> element or a <references> elemnt.

Sample snippet for a table entity:

```
&lt;tableEntity&gt;
      &lt;attributeList&gt;
      &lt;attributeListDefinition id="111" scope="document"&gt;
          &lt;attribute&gt;
           &lt;attributeDefinition id="444" scope="document"&gt;
               &lt;attributeName&gt;siteID&lt;/attributeName&gt;
            ........
           &lt;/attributeDefinition&gt;
           &lt;/attribute
          &lt;attribute&gt;
           &lt;attributeDefinition id="445" scope="document"&gt;
               &lt;attributeName&gt;Sample Date&lt;/attributeName&gt;
               .......
           &lt;/attributeDefinition&gt;
           &lt;/attribute&gt;
          &lt;attribute&gt;
           &lt;references&gt;334&lt;/references&gt;
           &lt;/attribute&gt;
        &lt;/attributeListDefinition&gt;
    &lt;/attributeList&gt;
    &lt;/tableEntity&gt;
```

Another example from constraint would be as follows (assuming that in my eml-constraint.xsd I had defined the element Key to be of type attributeListType):

<primaryKey>

```
&lt;attributeListDefinition id="555" scope="document"&gt;
      &lt;attribute&gt;
     &lt;references&gt;444&lt;/references&gt;
       &lt;/attribute&gt;
    &lt;attributeListDefinition&gt;
&lt;primaryKey&gt;
```

#5 - 05/29/2002 01:48 PM - Peter McCartney

Sorry...typo in the last code snippet in my comments. should read:

<primaryKey>
<key>
<attributeListDefinition id="555" scope="document">
<attribute>
<references>444</references>
</attribute>
</attributeListDefinition>
</key>
</primaryKey>

#6 - 05/30/2002 01:09 PM - Peter McCartney

Here are some comments.

1) under storageType, the prefix xs: should not be expected since prefixes for content models are defined by the individual schemas that import them and this is not the context in which these will be used. so i would expect people to type in "string" and not "xs:string", or "xsd:string"

2) I like the spirit behind your proposed unit field, but i feel the same about is as i do about connection URLs - i dont believe people will understand it well enough to use it. Typing in "http://ecoinformatics.org/unitDictionary?" for every entry seems a bit awkward and unnecessary. Just like with connection URLs, users will require both a wizard processor to help them construct it as well as processing code to interpret it, and if the dictionary isn't shipped with eml, then youve now created a dependency on a web address that we cant

guarantee will always be there. What i thought was going to come out of the sevilleta discussion was an element that was either free choice or an enumeration based on a list of stmml type names that are taken from this directory.

The problem seems similar to me to the spatial reference module, in which projections that people frequently refer to by a name "UTM zone 12" actually require a fairly complex set of terms and references to standard algorithms. In eml-spatialReference, these are encoded as complex types that define which parameters need to be filled in. I understand that part of what you want to do is allow a syntax for people to build thier own data types using the established ontology, but i think they will not respond well to the URI model for doing that. I'm afraid of them simply not filling it in if they can either type in the name they use or pick it from a controlled list.

Related to this section, i have a question regarding some fields from ISO that i am trying to eliminate on the grounds that we cover them elsewhere. for raster cells, ISO defines cellattributedescription, cellvalueunits, tonegradation, scalefactor and offset. the first, and perhaps all of these are covered in eml-attribute.xsd. cellvalueunits has a list of codes that i think could be indetified as an externalcodeset domain if that is brought back (see below). tone gradation is the number of colors(64 colors, 256 colors, etc). i think this could be gotten from storageType, but maybe we need something in enumeratedDomain for numberOfUniqueValues?. scale factor and offset are for any scale multipliers or delta constants that have been applied to the values. i think they mean transformations done to allow expression of values that are either larger or have a broader range than can be accomodated by the storage type used (ie using a byte data type for annual accumulation in thousands of inches). does stmml have a way of deal with this or do we need to leave these in?

3) the sequence portion of enumeratedDomain needs to repeat so that multiple codes can be entered. I dont thinke each code needs a separate source, but thats a minor point. My recollection from sevilleta was that we were going to let enumerated domain include a choice between providing a value list, providing a reference to an external codeset (codeSetName, codeSetURI?,codeSetCitation?) or a reference to an entity within the dataset whose data define the domain (entity, codeAttribute, codeDefinitionAttribute,

#7 - 06/13/2002 06:52 PM - Matt Jones

Changes completed and checked into cvs.

#8 - 06/27/2002 09:29 AM - Owen Eddins

I'm passing the following comments from Tim Bergsma the data manager at Kellog Biological Station in Michagen. He made them in a eml-dev email. I posting to bugzilla just to make sure they don't fall through the cracks.

8. Regarding <attribute>: there is in science a classic distinction between precision and accuracy. <accuracy> is used here in that sense; we should be aware that <precision> is not, at least not strictly. <precision> is used here in the sense of 'least significant digit', which may be related to but is not identical to the classical sense in which precision represents the repeatability of a measurement, and is statistically qualified. Precision is a messy issue. Suppose rain fall is measured to the nearest quarter of an inch. Converted to a decimal, quarter inches are represented as 0.25, which misleadingly suggests that precision is at the level of hundredths of an inch. Perhaps EML should allow a statement of precision that is not decimal-oriented.
9. Should <unit> be optional under <attribute>? Many attributes do not have units, such as <skyCondition>sunny</skyCondition>.

#9 - 08/20/2002 03:40 PM - Matt Jones

Regarding precision (issue 8):

I think decimal precision works fine based on the example given. To the nearest 1/4 inch is to the nearest 0.25 inch. If someone had meant to the nearest hundredth of an inch, they would write 1/100 inch or .01 inch for the precision. I don't understand the issue.

Regarding unit (issue 9):
The unit is dimensionless for those types of measures. What exactly dimensionless means is fairly open to debate.

#10 - 09/03/2002 05:17 PM - Peter McCartney

In responding to a request from Chad to provide a sample metadata file from our Xylographa tool, im finding a number of stumbling blocks as i go through filling in the blanks to make the file valid. Many of these are in attribute and result from the fact that unit, measurementScale and attributeDomain are all mandatory. This is forcing me to put in nonsense information for some attributes (in fact, in order to get my sample file to validate a couple weeks ago, this was exactly what Matt put in for these fields - nonsense). I dont see "dimensionless" as a choice under unit, although its valid if i just make the element but leave it blank. I can answer nominal for measurement scale, but how many people are going to recognize this as the right choice for a field they feel doesnt "measure" anything? finally, many attributes dont have a domain other than the general topic that the information is supposed to be about. I could fill in a textDomain entry that simply says "any text" but i would regard someone that did this with the same contempt i do people that put "N/A" in spread sheet cells. I think all of these need to be optional or we come up with an easier and meaningful way to generate default entries. If there really is no domain information, id rather the element not be there. same with unit and scale.

im also drawing blanks for things i should be able to put in unit - dates and times, photosynthetically active radiation (umol/m2/s), pressure in kiloPascals, and of course lots of ratios. we're going to need to have a really rich list as well as some primers on how to fill in custom unit definitions.
,

#11 - 09/13/2002 03:51 PM - Chad Berkley

done

#12 - 03/27/2013 02:14 PM - Redmine Admin

Original Bugzilla ID was 484

Files

eml-attribute.xsd	40.3 KB	05/29/2002	Peter McCartney