EML - Bug #3508

create a stylesheet for EML2.0.x to EML 2.1.0

10/06/2008 04:33 PM - Margaret O'Brien

Status:	New	**Start date:**	10/06/2008
Priority:	Immediate	**Due date:**	
Assignee:	Matt Jones	**% Done:**	0%
Category:	utilities	**Estimated time:**	0.00 hour
Target version:	Unspecified	**Spent time:**	0.00 hour
Bugzilla-Id:	3508		

Description

there are 4 changes in EML 2.1.0 which need to be addressed in a transformation stylesheet. Summarized:
1. datetime -> dateTime (bug #1152)
2. new additionalMetadata/metadata element (bug #2054)
3. method -> methods (bug #2568)
4. access trees moved (bug #1132)

Detailed:
1. xpath in instance docs:
/eml/dataset/dataTable/attributeList/attribute/measurementScale/datetime
becomes .../dateTime

2. xpath in instance docs (* neq <describes>):
/eml/additionalMetadata/* becomes /eml/additionalMetadata/metadata/*

3. method was declared in entity group, so all these paths become "methods":
eml/dataset/dataTable/method
eml/dataset/spatialRaster/method
eml/dataset/spatialReference/method
eml/dataset/spatialVector/method
eml/dataset/view/method
eml/dataset/storedProcedure/method

4. access trees moved:
case 1: access tree at resource level:
/eml/dataset/access and /eml/software/access moved to eml/access

case 2: access tree(s) in additionalMetadata sections:
Current behavior:
Do not recognize <access> siblings of additionalMetadata/describes that reference
/eml/dataset/distribution
/eml/software/distribution

Recognize <access> siblings of additionalMetadata/describes that reference
/eml/dataset/dataTable/physical/distribution
/eml/software/implementation/distribution

recognizable cases:
4.2.1.one additionalMetadata/describes element pointing a recognized node, with one sibling access tree
action: move the access tree, remove additionalMetadata section

4.2.2. two (or more) additionalMetadata/describes elements pointing to recognized nodes, with one sibling access tree
action: copy the access tree to both locations, remove additionalMetadata

4.2.3 two (or more) additonalMetadata/describes
4.2.3.1 references a distribution node
action: remove the <describes>, copy the access tree to the referenced node
4.2.3.2 references some other non-recognized node
action: leave the <describes>,
additionalMetadata/access -> additionalMetadata/metadata/access

4.3. one additionalMetadata/describes element pointing to a recognized node,

with multiple sibling access trees (in eml 2.0.x, these will need to have been wrapped in another element in order to be valid)
action: move nothing. can't deal with elements that arent 2.0.1 [aside: addtionalMetadata section will have <metadata> tag added per item 2. above]

4.4 the eml-author created access trees in additionalMetadata that contain non-eml material. we cannot move these to other parts of the EML instance document. [aside: If metacat has encountered additional not-eml elements alongside access elements, it would have ignored them.]

To be sure output is valid EML, a script should include the parser to check the doc against the schema. Can the stylesheet do this alone? Jing and margaret not sure.

4.1. assuming stylesheet finds the non-eml material (ie, it can parse), possible actions include:
a? ignore access tree, but wrap it in <metadata> </metadata>
or b? reject document, tell use that the resulting eml will be invalid
or c? rebuild access tree in the new location (ie, try to interpret what author wanted)

Related issues:

Blocks EML - Bug #2568: Inconsistent naming of "method"(s) element	**Resolved**	**10/18/2006**
Blocks EML - Bug #1152: datetime element does not use consistent naming	**Resolved**	**09/23/2003**
Blocks EML - Bug #2054: use of <any> in additionalMetadata is invalid	**Resolved**	**03/31/2005**
Blocks EML - Bug #1132: fix access control rule ambiguities	**Resolved**	**08/15/2003**

History

#1 - 10/07/2008 11:46 AM - Margaret O'Brien

Regarding dateTime:
the path for this template can be shortened to
attributeList/attribute/measurementScale/datetime

since attributeList appears in these schemas
eml-dataTable.xsd
eml-entity.xsd
eml-spatialRaster.xsd
eml-spatialVector.xsd
eml-storedProcedure.xsd
eml-view.xsd

#2 - 10/07/2008 03:32 PM - Jing Tao

The access subtree for data file level can be:

/eml/software/implementation/distribution
/eml/dataset/dataTable/physical/distribution
/eml/dataset/spatialRaster/physical/distribution
/eml/dataset/spatialVector/physical/distribution
/eml/dataset/view/physical/distribution
/eml/dataset/otherEntity/physical/distribution
/eml/dataset/storedProcesure/physical/distribution

We propose to use two paths:
//physical/distribution
//software/implementation/distribution

do you see any problem?

#3 - 10/15/2008 02:46 PM - Margaret O'Brien

This is another case for how access trees could have been used in eml201
case 4.3

/dataset/access
and
/dataset/distribution with an id attribute and additionalMetadata fpresumably for overriding document-level access for this node.

?? Mike says that metacat recognizes these nodes on eml201 docs. But in EML210, there is no access tree available to resource-level distribution elements (only at entity-level). So the question is what to do with this case.
Options:
a. ignore it (and lose access control at this level)
b. create an entity, and move the distribution tree to it, and copy the access tree from additonalMetadata as well

#4 - 10/15/2008 03:15 PM - Jing Tao

I guess the access control on resource-level should be a mistake. EML 2.0.1, we should only have two level - document level and entity level. This is reason why eml 2.1.0 only has the two levels.

#5 - 10/15/2008 04:56 PM - Margaret O'Brien

regarding case 4.2 (from irc)
daigle that is the case that we use in all the test cases for 2.0.1 inline access
daigle bunches of tests
daigle in fact, that's the only case that I'm positive works :)

So metacat does recognize /dataset/distribution nodes. But there is a variant of this case that we cannot transform to 2.1.0. If the doc also has access rules in additionalMetadata that override the /dataset/distribution with another access tree, then there is no place to move that second access tree.

There might be the possiblity of this action:
create an otherEntity, with a distribution tree, and copy the access tree from additonalMetadata to there. This would also require adding a otherEntity/physical/distribution/references node which referred to dataset/distribution (who's id attribute would have to be generated if it didn't exist). All this is possible (although is it advisable?) But the deal breaker is this: the entity group has some other required elements whose content we know nothing about. So I think this sub-case falls into the category of those docs we cannot transform.

In the eml/test directory, see the doc eml-datasetWithAccessOverride.xml, also attached to this comment. This test is similar to the doc we would be trying to create.

#6 - 10/21/2008 10:56 AM - Jing Tao

We should add otherEntity/method to this group:

3. method was declared in entity group, so all these paths become "methods":
eml/dataset/dataTable/method
eml/dataset/spatialRaster/method
eml/dataset/spatialReference/method
eml/dataset/spatialVector/method
eml/dataset/view/method
eml/dataset/storedProcedure/method
eml/dataset/otherEntity/method

#7 - 04/02/2009 10:36 AM - Margaret O'Brien

Open stylesheet bugs are now associated with EML's Utility component

#8 - 03/27/2013 02:23 PM - Redmine Admin

Original Bugzilla ID was 3508

Files

eml-datasetWithAccessOverride.xml	3.47 KB	10/15/2008	Margaret O'Brien