EML - Bug #2054

use of <any> in additionalMetadata is invalid

03/31/2005 12:03 PM - Saurabh Garg

Status:	Resolved	**Start date:**	03/31/2005
Priority:	Immediate	**Due date:**	
Assignee:	Matt Jones	**% Done:**	0%
Category:	eml - general bugs	**Estimated time:**	0.00 hour
Target version:	EML2.1.0	**Spent time:**	0.00 hour
Bugzilla-Id:	2054		

Description

Johnoel from Hawaii also reported the same problem which Margaret reported earlier while using latest version of XMLSpy2005 or oXygen to parse XML Schema

The error from oXygen is:
E cos-nonambig: "":describes and WC[##any] (or elements from their substitution group) violate "Unique Particle Attribution". During validation against this schema, ambiguity would be created for those two particles. eml.xsd 246:27

More information about this can be found here: http://www.w3.org/TR/2000/WD-xmlschema-1-20000407/#non-ambig

Following text from the above page descibes what is happening:
We say that two non-group particles **overlap** if

- One is a wildcard and the other an element declaration, and the {target namespace} <http://www.w3.org/TR/2000/WD-xmlschema-1-20000407/#e-target_namespace> of the element declaration, or of any member of its equivalence class <http://www.w3.org/TR/2000/WD-xmlschema-1-20000407/#key-eq>, is schema-valid with respect to the {namespace constraint} <http://www.w3.org/TR/2000/WD-xmlschema-1-20000407/#namespace_constraint> of the wildcard.

So the schema that we have in EML is:
<xs:complexType>
<xs:sequence>
<xs:element name="describes" type="xs:string" minOccurs="0" maxOccurs="unbounded">
</xs:element>
<xs:any processContents="lax">
</xs:any>
</xs:sequence>
<xs:attribute name="describes" type="xs:string" use="optional"/>
<xs:attribute name="id" type="res:IDType" use="optional"/>
</xs:complexType>

This is a problem because <xs:any> is a wildcard and could be anything including <describes> itself. In particular a document which has following text can confuse the parser
<additionalMetadata>
<describes>1</describes>
<describes>2</describes>
<describes>3</describes>
</additionalMetadata>
So here the parser doesnt know if the last <describes> tag should be considered as <xs:any> or not.
(Though I think that as only one <xs:any> is possible, the last tag should be takes as <xs:any> by default. But I must be missing something as both oXygen and XMLSpy complain about this)

I was able to correct this error by doing the following
1. <describes> tag is required and can occur only once

<xs:complexType>
<xs:sequence>
<xs:element name="describes" type="xs:string">
</xs:element>
<xs:any processContents="lax">
</xs:any>
</xs:sequence>
<xs:attribute name="id" type="res:IDType" use="optional"/>
</xs:complexType>
2. describes is an attribute of additionalMetadata
<xs:complexType>
<xs:sequence>
<xs:any processContents="lax">
</xs:any>
</xs:sequence>
<xs:attribute name="describes" type="xs:string" use="optional"/>
<xs:attribute name="id" type="res:IDType" use="optional"/>
</xs:complexType>
3. <xs:any> is inside the <describes> tag
<xs:complexType>
<xs:sequence>
<xs:element name="describes" type="xs:string" minOccurs="0"
maxOccurs="1">
<xs:sequence>
<xs:any processContents="lax">
</xs:any>
</xs:sequence>
</xs:element>
</xs:sequence>
<xs:attribute name="id" type="res:IDType" use="optional"/>
</xs:complexType>

I think the first one would be the best in terms of minimum change to the schema.

Related issues:

Is duplicate of EML - Bug #2479: unable to validate eml.xsd and related schem...	**Resolved**	**06/28/2006**
Blocked by EML - Bug #3508: create a stylesheet for EML2.0.x to EML 2.1.0	**New**	**10/06/2008**

History

#1 - 03/31/2005 12:10 PM - Saurabh Garg

Actually 1 might not be best choice as Margaret told me that EML best practices document recommends using multiple describes. So some LTER sites might already be using this. Another option is to have something like this:

<additionalMetadata>
<describes>1</describes>
<describes>2</describes>
<describes>3</describes>
<metadata>anything_in_here</metadata>
</additionalMetadata>

#2 - 03/31/2005 12:24 PM - James Brunt

<additionalMetadata> is already unbounded though so making <describes> required and solo seems the best option. Wouldn't be to hard to take each <describes> and create a new <additionalMetadata> container for it to retrofit existing documents. How many documents in metacat currently use multiple <describes>?

James

#3 - 03/31/2005 02:00 PM - Margaret O'Brien

addendum to Sid's comment #1
Currently, it's the eml-access documentation that shows multiple describes, and it's not covered in the Best Practices doc yet.
http://knb.ecoinformatics.org/software/eml/eml-2.0.1/eml-access.html
However, I was wondering how best to describe the use of access trees for

individual tables in the additionalMetadata section. If the EML best practices document included a section recommending usage as it currently stands, it will probably be out of date soon. Can someone hazard a guess on what the solution will be? Or should BestPractices not touch this tree yet? I was under the impression that controlling access to tables separately from metadata was important to some groups.

#4 - 03/31/2005 02:08 PM - Matt Jones

I don't know how many use this feature, but the design was specifically done that way to allow one metadata snippet to describe multiple elements in the document. For example, someone might have a specialized metadata element in additional metadata that describes 4 different entities in a data set. Making "describes" have a cardinality of 1 would eliminate this feature, and force someone to repeat the metadata snippet multiple times. Same problem would exist for Sid's proposed solutions 2 and 3. Another potential solution is:

4. wrap the xs:any in another containing element, describes is repeatable
<xs:complexType>
<xs:sequence>
<xs:element name="describes" type="xs:string"
maxOccurs="unbounded">
</xs:element>
<xs:element name="metadata" type="xs:string" maxOccurs="1">
<xs:complexType>
<xs:sequence>
<xs:any processContents="lax">
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:any>
</xs:sequence>
<xs:attribute name="id" type="res:IDType" use="optional"/>
</xs:complexType>

This has the disadvantage of not preserving backwards-compatibility (but so would the other potential solutions).

#5 - 03/31/2005 02:28 PM - James Brunt

Matt's suggestion to create a <metadata> element seems OK to me - it's just as easy as any of the others to retrofit maybe easier because you could just dump the <any> into it without worrying about multiple <describes> that may or may not be there. I guess I was confused about the use of <describes> then since it was just a string - seems it was designed to be used like a triplet? seems like if this is the purpose then it (<describes>) should be more constrained?

#6 - 06/28/2006 11:10 AM - Matt Jones

- Bug 2479 has been marked as a duplicate of this bug. ***

#7 - 06/29/2006 12:11 PM - John Cree

Closing Comment on bug #2479 was the following:

"The workaround for the time being until a new version of EML is released is to modify eml.xsd as described in bug #2054 in order to be able to proceed with your schema mapping activity."

A number of workarounds have been suggested under this bug, #2054, so exactly which workaround is the recommended one, and exactly which portion of the existing schema should be replaced? Thanks in advance for any assistance.

#8 - 06/29/2006 12:24 PM - Matt Jones

My intention was that you would follow the schema outlined under Comment #4 which preserves the ability to have multiple describes and still solves the schema ambiguity problem. That schema snippet in Comment #4 would replace the existing definition for additionalMetadata in the eml.xsd file.

#9 - 06/30/2006 08:18 AM - John Cree

If I replace the xml starting at:
<xs:element name="additionalMetadata" minOccurs="0" maxOccurs="unbounded">

and ending with:
</xs:appinfo>

</xs:annotation>
</xs:any>
</xs:sequence>
<xs:attribute name="id" type="res:IDType" use="optional"/>
</xs:complexType>
</xs:element>

with the snippet from Comment #4.

<xs:complexType>
<xs:sequence>
<xs:element name="describes" type="xs:string"
maxOccurs="unbounded">
</xs:element>
<xs:element name="metadata" type="xs:string" maxOccurs="1">
<xs:complexType>
<xs:sequence>
<xs:any processContents="lax">
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:any>
</xs:sequence>
<xs:attribute name="id" type="res:IDType" use="optional"/>
</xs:complexType>

, then eml.xsd, is still not well formed or valid. Validation in XMLSpy returns the following message: "xs:any closing element name expected.". Looking at the schema, I can see where this problem arises, but, not being an XML expert by any means, any of my attempts to correct to correct the lack of a closing element for <xs:any> result in more serious validation errors. Could somebody else try this out to verify the problem and provide a resolution. Thank-you.

#10 - 06/30/2006 09:04 AM - Matt Jones

Your right, there was a mistake in the snippet in comment #4 (xs:any end tag was mismatched). I've gone ahead and patched eml.xsd with a documented version of this soluttion in comment #4 and checked the patch into CVS. I will also attach a copy of the new modified eml.xsd and the diff with the previous version as an attachment to this bug for reference.

#13 - 03/21/2008 04:47 PM - Margaret O'Brien

changed summary ony

#14 - 09/22/2008 10:10 AM - Margaret O'Brien

changing status to "fixed"

#15 - 03/27/2013 02:19 PM - Redmine Admin

Original Bugzilla ID was 2054

Files

eml.xsd.patch	5.92 KB	06/30/2006	Matt Jones
eml.xsd	19.1 KB	06/30/2006	Matt Jones