EML - Bug #493

eml-software changes needed

05/01/2002 05:46 PM - Matt Jones

Status:	Resolved	**Start date:**	05/01/2002
Priority:	Immediate	**Due date:**	
Assignee:	Owen Eddins	**% Done:**	0%
Category:	eml - general bugs	**Estimated time:**	0.00 hour
Target version:	Beta9	**Spent time:**	0.00 hour
Bugzilla-Id:	493		

Description

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

1) Possible make software recursive for dependencies
2) add operating system
3) add processor architecture
4) remove location

possibly model it like:
software (versionNumber?, sourceModule*, documentation, dependencies*,
binaryFile*)

where sourceModule has a required attribute "langauge". binarFile would
probably need an attribute "encoding", but this is a hack.

this was a somewhat ad-hoc model from the discussion. Needs to be more
systematically reviewed. Consider the Open Software Description (OSD) note at
http://www.w3.org/TR/NOTE-OSD.html

Related issues:

Is duplicate of EML - Bug #145: revisions to software metadata standard	**Resolved**	**09/22/2000**

History

#1 - 05/01/2002 05:48 PM - Matt Jones

Missed a couple:

5) add publisher*
6) add pubPlace*
7) add contact* (depending on how this is resolved in resource

#3 - 05/15/2002 01:36 PM - Matt Jones

Hi Owen,

Your changes to software look good. Generally it looks much more comprehensive.
Thanks for looking over OSD. Here are a few issues I saw from my quick look:

1) You made several ComplexTypes, and I didn't understand why -- they don't seem
like they will/could be reused to me. In particular, "dependency", and
"codeBase". I also didn't understand what was going on with the "action"
SimpleType.

2) Is "codeBase" meant to point at external files? Is it somehow a physical
description of the file? Seems to me that "size" is an inadequate physical
description, and somewhat unneccessary. A checksum would be much more useful.
Any change we could utilize parts of eml-physical to describe the files, rather
than inventing this new "codeBase" structure? We might need to make changes to
eml-physical, or create a re-usable complex type for part of eml-physical that
is relevant. Also, how does this url/filename relate to the "distribution"
element in eml-resource? Aren't they the same?

3) What's the difference between language and programmingLanguage under
"implementation", and how do they both differ from "language" in eml-resource?

4) Is "implementation" a modular part of the whole software?

5) You haven't followed our type and element naming conventions. Elements should always start with a lower case letter and capitalize the first letter fo each subsequent word. Types should start with a capital and then capitalize each word. For example, the element "personalCommunication" should be of type "PersonalCommunication", and the element "map" should be of type "Map".

6) You are missing lots of documentation. We need the standard tooltip, summary, description, example, and lineage documentation tags to be present for every element and attribute. This documentation is what defines the spec. Without it people will have no guidance as to the contents of each field.

7) You broke the file formatting that we use. Please reformat the file with indentations for elements (2 spaces per level) and a maximum line length of 80 characters. This helps the files be more readable in various text environments.

8) The DTD needs to be updated to reflect these changes as well.

Thanks.

#4 - 05/15/2002 02:07 PM - Owen Eddins

Anyone. A big question I have right now with eml-software is as I have hacked it up is whether the

<softwarePackage>
<dependency>
<softwarePackage>
<dependency>
and so on...

as I have defined it is valid xml schema since dependency is defined as complex type of dependency which has as its sub-elements softwarePackage which has its sub-elements dependency and on. XML Spy does not complain about this but it does 'chop' it off at the second dependency. I changed dependency to optional (which it should be anyway) thinking that that would terminate the recursion. So for example, if the sub-element dependency, which is mandatory, defines its parent as a sub-element then there is no terminating the recursion. But Spy nevers shows the second dependency.

I'll address all your questions, make the changes where necessary. I currently am in the process of filling in the documentation.

#5 - 05/15/2002 03:12 PM - Owen Eddins

May have fixed the recursion issue by making the complexType dependency a global element instead. Spy stops showing the subtree after the dependency at least it is not truncating license choice and version. Question is still open as to whether this is valid xml schema?

#7 - 05/15/2002 04:12 PM - Owen Eddins

questions 1 & 2)

OSD was developed to be machine processable so the action element which I enumerated
as {install, assert} is meant either just assert the dependency or signal to some
program to do the installation. If install is set then it is implicit that the dependency is asserted. OSD treats this as an attribute I moved it up to the element
level.

As defined in OSD codebase is just the file or external link to it and its size.
I would agree that a checksum is appropriate but I also think size is important also.
Since this model is designed to be machine processable a lot of the elements at the
implementation level are appropriate if that is to occur. I looked over distribution
and it is the same thing as codebase. I'd like to propose that distribution be a
global element or complexType so in can plug it in to implementation. For

software
it is not appropriate as I have defined it to have distribution at the resource level
since a package could have multiple implementations. We could do away with this and
move the implementation level stuff up one level to softwarePackage so if there are
multiple implementations of some version of softwarePackage then you'd just have to
have multiple eml modules defined for each implementation. I think a case can be
made to support multiple implementations of a software package within one instance
of eml-softwarePackage. So for every hardware/os platform the java app 1.0 runs on all
the info in implementation could be filled out for platform specific differences.

So I'd like to propose that we nuke codebase altogether, since as you've pointed out
its redundent, and put checksum and size as subelements to implementation and add
Distribution as a global element and a subelement of implementation.

See new attachment.

#8 - 05/16/2002 10:57 AM - Matt Jones

OK, I like your proposal. As far as distribution goes, its details are being worked out. Its still unclear exactly how it will work, especially in terms of retrieving multiple objects. I think Peter envisions the distribution/connection element as a way to point at a general connection, and then information in eml-physical is used to provide the exact name/id of an entity that can be retrieved from a particular connection. Dan's comments on eml-physical show that this is still unclear. Nevertheless, however we decide to structure distribution, I think we can make it so that it supports retrieval of the software implementations as well. Is there a distinction between a code distribution and a binary distribution in terms of an implementation?

Some Bugzilla usage notes: 1) Please "accept" this bug. 2) Please mark previous patches as obsolete when you add a new version (check the checkbox) so that people will know which attachment is the most current. Thanks.

#9 - 05/16/2002 12:50 PM - Owen Eddins

Question:

Is there a distinction between a code distribution and a binary distribution in terms of an implementation?

Answer:

Not as I'm envisioning it. The model makes no distinction between binary and source distribution and if one wanted to logically group the binary and source distributions together it would support that OR breaking it up in to two separate instances of an eml-softwarePackage compliant document.

Side note: I added the dependency element as a sub-element to implementation AND left it at the package level because it occurs to me you can have package and implementation level dependencies.

#10 - 05/17/2002 09:00 AM - Owen Eddins

Matt,

Question:

```
3) What's the difference between language and programmingLanguage under
    "implementation", and how do they both differ from "language" in eml-
resource?
```

Answer:

```
A &lt;softwarePackage&gt;/&lt;implementation&gt; could have several supported
languages so
```

```
    langugae at the eml-resource level does work if we are going to support
multiple
    implementations within a software package.  Rather than just making
language a
    simple element of simple string type I structured it so it will support
simple
    string types and/or some standard like ISO 639-2 or 639-1.

The structure I came up with is

&lt;xs:element name="Language" minOccurs="0" maxOccurs="unbounded"&gt;
      &lt;xs:complexType&gt;
        &lt;xs:sequence&gt;
        &lt;xs:element name="LanguageValue" type="xs:string"&gt;

&lt;/xs:element&gt;
        &lt;xs:element name="LanguageCodeStandard" type="xs:string"
minOccurs="0">

&lt;/xs:element&gt;
        &lt;/xs:sequence&gt;
      &lt;/xs:complexType&gt;
    &lt;/xs:element&gt;

Both sub-elements could be gotten rid of and CodeStandard could be made
an attribute
    of Language element, either one works for me.

I had forgotten that language was going in to eml-resource.  Whatever
you come up
    with at resource level could we make it either a global element or
complex type like
    distribution and I can plug it in at the
&lt;softwarePackage&gt;/&lt;implementation&gt; level
    also.
```

Thanks,

Owen

#11 - 05/24/2002 02:15 PM - Owen Eddins

still need to clean up comments

#13 - 06/14/2002 08:22 AM - Matt Jones

Completed as per discussion.

#14 - 06/14/2002 08:25 AM - Matt Jones

- Bug 145 has been marked as a duplicate of this bug. ***

#15 - 03/27/2013 02:14 PM - Redmine Admin

Original Bugzilla ID was 493

Files

osd.xsd	5.87 KB	05/15/2002	Owen Eddins
eml-softwarePackage.xsd	6.5 KB	05/15/2002	Owen Eddins
eml-softwarePackage.xsd	6.34 KB	05/15/2002	Owen Eddins
eml-softwarePackage.xsd	17.9 KB	05/24/2002	Owen Eddins
eml-software.xsd	22.3 KB	05/29/2002	Owen Eddins