EML - Bug #558

paragraph tag needs formatting structure

08/08/2002 08:11 AM - Matt Jones

Status:	Resolved	**Start date:**	08/08/2002
Priority:	Immediate	**Due date:**	
Assignee:	Matt Jones	**% Done:**	0%
Category:	eml - general bugs	**Estimated time:**	0.00 hour
Target version:	EML2.0.0rc1	**Spent time:**	0.00 hour
Bugzilla-Id:	558		

Description

Bergsma suggested that the paragraph tag needs structure to accomodate real-life text such as lists and hierarchies. Chapal seconds the need. I tend to agree, but am not sure currently how to accomplish this. The viable proposals from my perspective seem to be:

1) allow XHTML inside paragraph tags
2) allow Docbook or simple Docbook in paragraph tags

Other proposals that seem less viable include:
3) Decompose structured text into a series of <paragraph>.
4) Inject structured text, with its native markup, as a CDATA block in <paragraph>.
5) Make <paragraph> nestable

Three (3) and five (5) don't seem to solve the problem completely. Four (4) solves the problem but isn't at all standardized, and so interpreting paragraphs would be essentially impossible depending on what people used for their markup. Every author could use their own markup if we sanctioned (4).

Here's a copy of the email thread that lead to this bug:

---- Start message from scott.chapal at jonesctr org -------

Has there been any discussion regarding Tim's questions? I thought he pointed out an area that deserves clarification before EML 2 is released.

I too, am thinking there ought to be additional structure to represent prose.

Tim Bergsma <tbergsma@kbs.msu.edu> writes:

> The problem, as I have mentioned previously, is that prose metadata (text) is often highly structured. <paragraph> gives us no way of representing the structure of text, which is itself information. In many instances, of course, <paragraph> is repeatable, which allows us some leeway to represent sequential structure. But there is still no way to represent hierarchical structure. This has significant consequences. For example, a project-level abstract may include a short outline of purposes or hypotheses. A research protocol may include finely-grained outlines of contingencies and responses.
>
> Three alternative solutions have emerged from previous discussion.
> 1. Decompose structured text into a series of <paragraph>.
> 2. Inject structured text, with its native markup, as a CDATA block in <paragraph>.
> 3. Make <paragraph> nestable.

...

> I hope that the leadership of the eml development community will offer me some guidance on this issue. I really don't think number 1 is a viable option, but could make peace with either 2 or 3.

I don't particularly like any of those options.

How about deferring to some [SGML/XML] standard to represent prose?

Possibly DocBook; use the (W3C)Schema when it is complete?

Or even the simplified DocBook?
http://www.oasis-open.org/committees/docbook/specs/wd-docbook-simple-1.0-CR2.html

Other??

Necessarily, existant documentation would have to be deconstructed or converted to a markup language format. Or if visual formatting is paramount then it could point to a (quasi) neutral file format like .pdf, but that wouldn't accomplish textual indexing, querying and structure that EML aims to support.

This comment of Tim's really struck me:

> However, converting hierarchically-structured text to serially-structured text will require innovations by the data manager that raise him/her to the status of author, a status not necessarily sanctioned by those who contributed the original material.

I think this concern is largely unwarranted. Were it viable, Information Scientists, 'New' librarians and editors wouldn't be able perform their functions either. Fidelity to an original work should be mandatory, but the translation process should be made transparent and trivial. If an editing review is needed, then create one. Perhaps even a 'stamp of approval' or something. We're talking about metadata -- **documentation**, after all. If it's not structured it's really not very useful.

---- End message from scott.chapal at jonesctr org -------

Related issues:

Is duplicate of EML - Bug #557: paragraph tag needs formatting structure	**Resolved**	**08/08/2002**

History

#1 - 08/22/2002 08:05 AM - Chad Berkley

It was agreed in the conference call on 8/21/02 that Matt and Tim would look over this issue and post possible solutions to bugzilla and eml-dev.

#3 - 08/26/2002 01:50 PM - Matt Jones

- Bug 557 has been marked as a duplicate of this bug. ***

#4 - 08/26/2002 02:59 PM - Matt Jones

Tim, so I looked over your suggestions. Good overview. Thanks. Here's my take... Either we can go simple and include only a very few elements for paragraph structuring, or we can go complex and let most of docbook in. But a middle ground is the worst case. Here's why...

If we go simple, then we'll only need to define a text container type of tag that can be used throughout EML, and we can make its few elements correspond to docbook or XHTML (e.g., para, ol, ul, li, emphasis). This is easy to implement, and simple so easy to understand for EML authors and EML readers. And it would be a subset of docbook so the translation would be 1:1 to docbook.

If we go complex, we could wholesale import tags from the docbook namespace. nobody will understand the tags (there are far too many), but we can cut and paste from tools that output docbook format, and we can utilize already existing stylesheets that know how to format & present docbook. This is a far more complex but also more flexible solution.

If we go the middle road, then there are more tags than a user can understand, and we lose the ability to wholesale import an existing standard (so off the shelf stylesheets and software won't be usable, and its a pain to maintain). Worst case scenario to me.

In looking at what you think is needed/not needed, it seems that the following tags would be very useful:

ItemizedList, OrderedList, ListItem; (corresponds to ul,ol,li)
emphasis; (no italics in DocBook, predictably.)
procedure, step, substep; (Great! substeps nestable to any depth!)

You also say that sections would be useful, but aren't allowed in paragraphs in docbook:
section, title; (sections are great, but unfortunately cannot be in paragraphs.)

I would argue that the best course of action is for us to create a single generic container for structured text that is mixed content and can be re-used throughout EML, and is a direct subset of docbook. We might call it 'textBlock' or something like that, and it could contain titles, sections, and paragraphs. paragraphs could then in turn contain text, lists, and emphasis.

This would allow most of the formatting people would want, without importing the whole docbook namespace. The simplest case would devolve to <textBlock><para>This is some text<para></textBlock>, and so is basically what we have now. This has the disadvantage of being non-standard formatting, so requires us to have custom stylesheets for it (which would be a simple 1:1 mapping to docbook). But then, almost all of EML is non-standard and requires custom stylesheets, so that's not really an additional burden. I'll go ahead and create a candidate schema to demonstrate this proposal. What do you think?

#5 - 09/05/2002 04:30 PM - Matt Jones

OK, looks like people are pretty happy with TextType as it is, given that method will be changed to accomodate the recursive step definitions. So, all we need to do is go through and redefine all elements that contain paragraph to be of type "TextType". For example, in eml-resource, "abstract" would be of type "TextType". This is a little problematic for elements that currently allow mixed paragraphs with citations or other elements, so for those we could change the "paragraph" element to be named "text" and be of type "TextType". I'll do this tomorrow if no objections are raised.

#6 - 09/09/2002 02:03 AM - Matt Jones

RESOLVED FIXED. TextType incorporated into all modules that used to contain "paragraph" elements.

#7 - 03/27/2013 02:14 PM - Redmine Admin

Original Bugzilla ID was 558

Files

para.txt	2.28 KB	08/22/2002	Tim Bergsma