Subject: XML list files in an element-centric structure (was: small issues on "register" and "tLineInfrastructureManagerCode")
Posted by Susanne Wunsch railML on Thu, 06 Dec 2012 09:53:51 GMT
View Forum Message <> Reply to Message

Dear Dirk and others,

I want to discuss the general structure of separate XML files as
replacement for schema-internal enumeration lists in this 'misc' forum
because it concerns all sub-schemas.

Susanne Wunsch <coord@common.railml.org> writes:
> Dirk Bräuer <dirk.braeuer@irfp.de> writes:
>>> concerning the file structure: I would prefer using named attributes
>> rather than default attributes with element names for shortness
>> (<version code='ENEE'> rather than <code>ENEE</code>).
>
> Yes, that is a break in the current railML encoding style.
>
> We often stumble over attributes that would need to occur more than
> once. That's obviously not possible with XML. We need to convert them to
> elements so far. That is a "structure-braking-change". In contrast
> changing the elements multiplicity from "1" to "unbounded" is a small
> change.
>
> That's the reason for changing the style concept starting with these new
> "separate XML file" additions.

I would like to introduce this new style with the separate XML
files. There are good reasons for attribute-centric and good reasons for
element-centric schemas. These discussions were held in dozens of
newsgroups, mailing lists and web-forums for various domains. It's not a
railway-specific issue. It's a general XML-style issue. Therefore I
would adopt the recommendations of these two conclusions of well-known
XML experts. [1] [2]

What would change for the railML style?

* Promote all attributes to elements, despite of the following

  @id, @code, @xml:lang

That change is currently only proposed to the XML list files. The railML
schema files keep unchanged.

Any comments, questions, concerns, +1, -1 ... appreciated.

Kind regards...

Susanne

[1] http://www.ibm.com/developerworks/xml/library/x-eleatt/index .html
[2] http://recycledknowledge.blogspot.de/2008/03/elements-or-att ributes.html

Crosspost & Followup-To: railML.misc
--
Susanne Wunsch
Schema Coordinator: railML.common

Subject: Re: XML list files in an element-centric structure (was: small issues on "register" and "tLineInfrastructureManagerCode")
Posted by   on Fri, 07 Dec 2012 14:17:27 GMT
View Forum Message <> Reply to Message

Dear Susanne,

thank you for providing the possibility to take part at this important
decision.

To make it short: I opt for keeping the current philosophy of usage of
attributes and elements. I opt for still using attributes for simple,
base-type information.

In general: The reason for my opinion is the file size: With elements a
file is bigger than with attributes because of the repeating of the
element names. Due to a simple file-size proportionality, the processing
of element-like files tends to be longer than that of attribute-like
files. We already have very large RailML files with processing times
sometimes of some minutes (!) and I do not want to get a problem with file
size or processing times in future.

But I am aware that there are many arguments for both solutions and I also
do not want to start this discussion again here. So this should be a
simple vote in a democratic manner without much arguing.

In specific: If RailML 3.0 or 10.0 or whatever would change all the
existing attributes to elements (even if except the three you named), we
possibly wouldn't be able to use that RailML schemes because I think it is
too much effort to change all the existing programming for iRFP and for
all our partners. So here in particular I opt a little bit "stronger" for
keeping the structures and philosophy we already have.

With best regards,
Dirk.