Posted by Joachim Buechse on Thu, 25 Sep 2003 16:36:08 GMT
View Forum Message <> Reply to Message

As part of on ongoing project Ergon will contribute several drafts for
other schemas of 'operational data' to railml. We have a strong
interest, that thoose schemas are officially published by railml.org.
Also we will have to implement upwards and downwards compatibility -
i.e. a client written to process timetable data encoded with schema 0.93
should be able to read data encoded with schema 1.02 by downloading and
applying an xslt transformation script from a predefined location [as
long as this is semantically possible of course].

[1] All current railml main tags (rollingstock, timetable, etc) have a
version attribute which is very good for compatibility and automated
conversions. This same rule should be followed for all future schemas as
well.

[2] I suggest, that we define the url pattern(s) for RailML schema(s).
My suggestion is to use the schema version as part of the name. (In the
currently published examples the namespace definitions refer to files
rather than urls). I additionally suggest to use separate (sub-)schemas
for infrastructure, timetable and rollingstock. This allows the usage of
separate namespaces for the tags. [It will also get more important once
we contribute the operational data schema drafts which might otherwise
cloak the schedule for railml 1.0, 1.1, etc].

http://www.railml.org/schemas/<version>/<name-without-xsd>

  http://www.railml.org/1.00/railml
  http://www.railml.org/1.00/infrastructure
  http://www.railml.org/1.00/timetable
  http://www.railml.org/1.00/rolingstock

[3] I suggest that we define urls where XSLT transfomations scripts can
be located that tranfsorm data encoded against different schema versions.

http://www.railml.org/transformers/<oldversion>-<newversion>/<name.xsl>

  http://www.railml.org/transformers/0.94-1.00/infrastructure. xsl

[I realize that automated transformations will not be generally possible
based on XSLT, especially if semantical changes are involved.]

In our application update mechanism we use a system where several

conversion might have to be performed in sequence i.e.

```
if not exists 0.76 -> 3.12 then
  use 0.76 -> 1.00 -> 2.00 -> 3.00 -> 3.12
if not exist 0.76 -> 1.00 then
  use 0.76 -> 0.8 -> 0.9 -> 1.0 -> ...
...
```

This might be a reasonable approach for the xsl transformers as well. If this is considered an idea worthwile following, I will provide some java code as a reference implementation.

[4] I suggest that we define a url where a conversion servlet is hosted

  http://www.railml.org/converter

This URL may redirect the client to another URL (including an https URL). The servlet may require an HTTP Basic-Authorization header with username and password [in case such service would be commercially used].

The servlet should accept a POST request with a multipart-formdata (MIME) encoded data stream with the fields

TARGET_VERSION=
DATA=

The source version is included in the XML data beeing passed and hence doesnt need to be provided as a parameter.

[Such a servlet might include semantic value transformations that are not possible with XSLT. The servlet may also implement the same functionality as a (SOAP) Webservice].

PS: I fully realize that creation of XSLT transformers [3] or a converter servlet [4] should currently not be a priority for railml.org. However defining them now will allow implementors to include the mechanisms in their application for future use. Theese two mechanisms provide goods fields for IT student or apprentice work, so I am confident they will be implemented eventually.

Best regars,
Joachim Buechse

--
buechse@ergon.ch, Phone +41 1 268 89 58, Fax +41 1 260 20 65

Ergon Informatik AG, Kleinstrasse 15, 8008 Zuerich, Switzerland
http://www.ergon.ch

_____

e r g o n   smart people - smart software

Subject: Re: RFC: schema locations, transformations
Posted by Nils Poldrack on Mon, 19 Apr 2004 15:01:14 GMT
View Forum Message <> Reply to Message

Hello,

I refer to posting of Joachim Buechse 2003-09-25, 18:36

Joachim Buechse schrieb:
> [2] I suggest, that we define the url pattern(s) for RailML schema(s).
> My suggestion is to use the schema version as part of the name. (In the
> currently published examples the namespace definitions refer to files
> rather than urls). I additionally suggest to use separate (sub-)schemas
> for infrastructure, timetable and rollingstock. This allows the usage of
> separate namespaces for the tags. [It will also get more important once
> we contribute the operational data schema drafts which might otherwise
> cloak the schedule for railml 1.0, 1.1, etc].
>
> http://www.railml.org/schemas/<version>/<name-without-xsd>
>
>   http://www.railml.org/1.00/railml
>   http://www.railml.org/1.00/infrastructure
>   http://www.railml.org/1.00/timetable
>   http://www.railml.org/1.00/rolingstock
>

There was no comment to this proposal. But now I suggest to define the
URL patterns the other way round as Joachim proposed. My favourite URL
pattern is
 http://www.railml.org/<scheme>/<version>
Reasons for this rotation:
1. There is more order in this system.
2. We do not have "empty" directories with only one scheme. (This might
appear when there is only a v1.05 from timetable, but even a 2.x from
all other schemes.)
3. It is immediately visible with is the newest version of a scheme.

I'm just creating a pattern document for the documentation of changing a
railML-scheme. There MUST be written down the storage of all documents
and schemes. And a uniform path is REQUIRED.

Thank you for your comments,

best regards
Nils P.

PS. If there is no comment or veto against, I will write as proposed.