Subject: small issues on "register" and "tLineInfrastructureManagerCode"
Posted by on Mon, 12 Nov 2012 12:09:32 GMT
View Forum Message <> Reply to Message

Dear Christian,

<ocp>.<designator>.register is of type string.
<line>.infrastructureManagerCode is of type tLineInfrastructureManagerCode (enumeration).

Both are of the same kind, so may be both should be harmonised in type (either string or enumeration).

In Wiki, the following enumerations for <ocp>.<designator>.register are defined so far:

> Currently predefined 'register' enumeration entries:- ENEE Station code
> of ENEE database managed by UIC.- IBNR ... (EFZ)...- DS100 German
> "Betriebsstellenkürzel" ...
> - DB640 Austrian "Dienstbehelf No. 640" ...
> - DIDOK Swiss "Dienststellendokumentation" of Schweizerische
> Bundesbahnen on behalf of Bundesamt für Verkehr."

In the XSD at an annotation it is written “e.g. IBNR, DB640, Ril100, DIDOK”. Please harmonise the XSD with Wiki (not Ril100 but DS100).

If tLineInfrastructureManagerCode keeps to be an enumeration: Please extend tLineInfrastructureManagerCode with ThE, EVB and other known European IM. Would it be advisable - for political correctness - to use some kind of ‘official’ abbreviation like “DB Netz AG”, wouldn’t it?

```
<xs:enumeration value="DB-Netz" />
<xs:enumeration value="Infrabel" />
<xs:enumeration value="ROeEE" />
<xs:enumeration value="ÖBB-Infra" />
<xs:enumeration value="SBB-Infrastruktur" />
<xs:enumeration value="BLS-Netz" />
<xs:enumeration value="SŽDC" />
```

Thank you,
Dirk.

Subject: Re: small issues on "register" and "tLineInfrastructureManagerCode"
Posted by Susanne Wunsch railML on Mon, 12 Nov 2012 14:02:56 GMT
View Forum Message <> Reply to Message

Dirk Bräuer <dirk.braeuer@irfp.de> writes:

> <ocp>.<designator>.register is of type string.
> <line>.infrastructureManagerCode is of type
> tLineInfrastructureManagerCode (enumeration).
>
> Both are of the same kind, so may be both should be harmonised in type
> (either string or enumeration).

I would prefer an extensible enumeration list for the register. :-)

> In Wiki, the following enumerations for <ocp>.<designator>.register
> are defined so far:
>
>> Currently predefined 'register' enumeration entries:- ENEE Station
>> code of ENEE database managed by UIC.- IBNR ... (EFZ)...- DS100
>> German "Betriebsstellenkürzel" ...
>> - DB640 Austrian "Dienstbehelf No. 640" ...
>> - DIDOK Swiss "Dienststellendokumentation" of Schweizerische
>> Bundesbahnen on behalf of Bundesamt für Verkehr."
>
> In the XSD at an annotation it is written “e.g. IBNR, DB640, Ril100,
> DIDOK”. Please harmonise the XSD with Wiki (not Ril100 but DS100).

According to Wikipedia [1] and Deutsch Bahn [2] it is abbreviated "RL
100" nowadays. So we should change to "RL100" instead of "Ril100".

I reopened the Trac ticket for this issue to clarify with railML 2.2:

http://trac.assembla.com/railML/ticket/112#comment:10

> If tLineInfrastructureManagerCode keeps to be an enumeration: Please
> extend tLineInfrastructureManagerCode with ThE, EVB and other known
> European IM.

Thanks for this additions. I found a Wikipedia page for some German IMs,
but this list seems to be not complete, 'ThE' is missing there. [3]
Switching the wiki language other European IMs may be found. ;-)

I reopened the Trac ticket for this issue to extend with railML 2.2:

http://trac.assembla.com/railML/ticket/152#comment:8

> Would it be advisable - for political correctness - to
> use some kind of ‘official’ abbreviation like “DB Netz AG”, wouldn’t
> it?
> <xs:enumeration value="DB-Netz" />

That would be possible, it results in possible changing all the other
IMs the same way. :-(

What to do, if they again change their company name?

How about your mentioned example: "DB Netz AG" (website) vs "DB Netze"
(PDF, [2])?

> <xs:enumeration value="Infrabel" />
> <xs:enumeration value="ROeEE" />
> <xs:enumeration value="ÖBB-Infra" />
> <xs:enumeration value="SBB-Infrastruktur" />
> <xs:enumeration value="BLS-Netz" />
> <xs:enumeration value="SŽDC" />

[1] http://de.wikipedia.org/wiki/Bahnamtliches_Betriebsstellenve rzeichnis
[2] http://www.deutschebahn.com/site/dbnetz/zubehoer__assets/de/
anhaenge/betriebsstellen/betriebsstellen.pdf
[3] http://de.wikipedia.org/wiki/Eisenbahninfrastrukturunternehm en#Streckenl.C3.A4ngen

Kind regards...
Susanne

--
Susanne Wunsch
Schema Coordinator: railML.common

Subject: Re: small issues on "register" and "tLineInfrastructureManagerCode"
Posted by on Mon, 12 Nov 2012 14:22:13 GMT
View Forum Message <> Reply to Message

I would prefer it the other way 'round:

There are many infrastructure companies, it can become easily more or less
and they can change their names as their shirts: So I would prefer a
string here. I see no reason why a software would have to "understand"
that value - it is just a note for the user. (If there would be any
function behind the IM, it would be a matter for <operationModeChange> or
something like that.)

On the contrary, <designator>.register is something very important for a
software (may be a primary key). There are much less "registers" than IMs,
they are of much more "internal" nature and therefore do not change their
names so quickly, and with further computerisation we can expect less
"registers" in future. So I think an enumeration for "register" would be

very helpful when exchanging data between softwares.

Dirk.

Subject: Re: small issues on "register" and "tLineInfrastructureManagerCode"
Posted by Christian Rahmig on Sat, 24 Nov 2012 11:11:29 GMT
View Forum Message <> Reply to Message

Dear Susanne, Dirk and other railML users,

>> In Wiki, the following enumerations for <ocp>.<designator>.register
>> are defined so far:
>>
>>> Currently predefined 'register' enumeration entries:- ENEE Station
>>> code of ENEE database managed by UIC.- IBNR ... (EFZ)...- DS100
>>> German "Betriebsstellenkürzel" ...
>>> - DB640 Austrian "Dienstbehelf No. 640" ...
>>> - DIDOK Swiss "Dienststellendokumentation" of Schweizerische
>>> Bundesbahnen on behalf of Bundesamt für Verkehr."
>>
>> In the XSD at an annotation it is written “e.g. IBNR, DB640, Ril100,
>> DIDOK”. Please harmonise the XSD with Wiki (not Ril100 but DS100).
>
> According to Wikipedia [1] and Deutsch Bahn [2] it is abbreviated "RL
> 100" nowadays. So we should change to "RL100" instead of "Ril100".
>
> I reopened the Trac ticket for this issue to clarify with railML 2.2:
>
> http://trac.assembla.com/railML/ticket/112#comment:10

I agree with the idea to change the type of the register's name from string to an enumeration list. For the implementation and according to the wiki, I suggest the following values:

```
<xs:simpleType name="tRegisterName">
  <xs:union>
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="ENEE" />
        <xs:enumeration value="IBNR" />
        <xs:enumeration value="DB640" />
        <xs:enumeration value="RL100" />
        <xs:enumeration value="DIDOK" />
      </xs:restriction>
    </xs:simpleType>
    <xs:simpleType>
      <xs:restriction base="rail:tOtherEnumerationValue" />
```

```
    </xs:simpleType>
  </xs:union>
</xs:simpleType>
```

> [1] http://de.wikipedia.org/wiki/Bahnamtliches_Betriebsstellenve rzeichnis
> [2] http://www.deutschebahn.com/site/dbnetz/zubehoer__assets/de/anhaenge/betriebsstellen/betriebsstellen.pdf

Regards

--
Christian Rahmig
railML.infrastructure coordinator

Subject: Re: small issues on "register" and "tLineInfrastructureManagerCode"
Posted by Susanne Wunsch railML on Mon, 26 Nov 2012 11:10:43 GMT
View Forum Message <> Reply to Message

Christian Rahmig <coord@infrastructure.railml.org> writes:

>> I reopened the Trac ticket for this issue to clarify with railML 2.2:
>>
>> http://trac.assembla.com/railML/ticket/112#comment:10
>
> I agree with the idea to change the type of the register's name from
> string to an enumeration list.

Me not ... sorry to change the opinion to this aspect.

As agreed at the meeting in Zurich we will implement a free string attribute in the XML Schema, but provide an additional XML file with pre-defined values for this attribute.

Kind regards...
Susanne

--
Susanne Wunsch
Schema Coordinator: railML.common

Subject: Re: small issues on "register" and "tLineInfrastructureManagerCode"
Posted by on Mon, 26 Nov 2012 12:02:26 GMT
View Forum Message <> Reply to Message

Dear Susanne and Christian,

> As agreed at the meeting in Zurich we will implement a free string
> attribute in the XML Schema, but provide an additional XML file with
> pre-defined values for this attribute.

Despite of having suggested an enumeration here, I would accept Susannes solution. Here, anything is better than a pure string. It is ok, as long as the spelling of well-known registers is defined anywhere in RailML. In this sense, an enumeration and a string with pre-defined values makes no difference for me.

Please don't forget the other extreme already mentioned: The /infrastructureManagerCode/. As long as nobody wants to enumerate (and maintain!) all the known IMs (ThE, EVB, HzL,...): Please define it as a string (possibly with some pre-defined values) before it is too late for 2.2.

Thanks,
Dirk.

Subject: Re: small issues on "register" and "tLineInfrastructureManagerCode"
Posted by Christian Rahmig on Sun, 02 Dec 2012 10:18:45 GMT
View Forum Message <> Reply to Message

Hello Susanne,

> As agreed at the meeting in Zurich we will implement a free string
> attribute in the XML Schema, but provide an additional XML file with
> pre-defined values for this attribute.

Agreed.

Regards

--
Christian Rahmig
railML.infrastructure coordinator

Subject: Re: small issues on "register" and "tLineInfrastructureManagerCode"
Posted by Susanne Wunsch railML on Mon, 03 Dec 2012 13:06:09 GMT
View Forum Message <> Reply to Message

Christian Rahmig <coord@infrastructure.railml.org> writes:

>> As agreed at the meeting in Zurich we will implement a free string

>> attribute in the XML Schema, but provide an additional XML file with
>> pre-defined values for this attribute.

I would suppose the following structure for the pre-defined registers in
the extra XML file:

```
<registers xmlns="http://www.railml.org/lists">
  <register id="d1e3">
    <version>
      <code>ENEE</code>
      <name>European Railway Location Database</name>
      <validity/>
      <remarks/>
    </version>
  </register>
  <register id="d1e51">
    <version>
      <code>RL100</code>
      <name>Richtlinie 100</name>
      <validity begin=""/>
      <remarks/>
    </version>
    <version>
      <code>DS100</code>
      <name>Drucksache 100</name>
      <validity begin="1951" end=""/>
      <remarks/>
    </version>
    <version>
      <code>DV100</code>
      <name>Dienstvorschrift 100</name>
      <validity end="1951"/>
      <remarks/>
    </version>
  </register>
</registers>
```

This would be used in the 'ocp':

```
<ocp>
  ...
  <designator register="RL100" entry="...">
</ocp>
```

Kind regards...
Susanne

--

Susanne Wunsch
Schema Coordinator: railML.common

Subject: Re: small issues on "register" and "tLineInfrastructureManagerCode"
Posted by on Mon, 03 Dec 2012 15:33:45 GMT
View Forum Message <> Reply to Message

Dear Susanne,

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>).

concerning the contents: Please do not provide redundant elements for the same register ("RL100", "DS100", "DV100" are all the same). You find examples for different registers in Wiki. If you feel necessary to provide different names for the same register, disjunctive validities should be enforced. This is not the case in your example.

Best regards,
Dirk.

Subject: Re: small issues on "register" and "tLineInfrastructureManagerCode"
Posted by Susanne Wunsch railML on Mon, 03 Dec 2012 21:08:23 GMT
View Forum Message <> Reply to Message

Dear Dirk,

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.

> concerning the contents: Please do not provide redundant elements for
> the same register ("RL100", "DS100", "DV100" are all the same).

That is one example for showing how this register renaming may be defined with the newly created structure.

> You find examples for different registers in Wiki.

Yes, we will fill out the other values, too. But I tried to only cite an excerpt from the file for two use cases - two registers, one with historic names, one with only one name.

> If you feel necessary to provide different names for the same
> register, disjunctive validities should be enforced. This is not the
> case in your example.

Yes, you are right. I tried to find out, when the renaming from "DS100" to "RL100" was done, but I couldn't find it. That's the reason for the blank "begin" and "end" attributes. It was my intention to show this use case with disjunctive validities. I'm sorry for that.

Kind regards...
Susanne

--
Susanne Wunsch
Schema Coordinator: railML.common

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: Identification in the XML list files and its references (was: small issues on "register" and "tLineInfrastructureManagerCode")
Posted by Susanne Wunsch railML on Thu, 06 Dec 2012 10:29:03 GMT
View Forum Message <> Reply to Message

Dear Dirk, Christian and others,

I would like to discuss the general content structure of the separate

XML list file in the 'misc' forum, because it concerns all sub-schemas.

The separate XML list files should be an easier to maintain replacement
for schema-internal enumeration lists.

I will change the already proposed example a bit taking care of the
parallel discussion about attribute- or element-centric XML
styles. Please don't discuss this issue here. The following XML
structure may be easy changed into an attribute-centric one, if that
comes as consensus from the neighbouring thread.

```
<registers xmlns="http://www.railml.org/lists">
  <register id="d1e3">
    <version code="ENEE">
      <name>European Railway Location Database</name>
      <validity/>
      <remarks/>
    </version>
  </register>
  <register id="d1e51">
    <version code="RL100">
      <name>Richtlinie 100</name>
      <validity begin="xxxx"/>
      <remarks/>
    </version>
    <version code="DS100">
      <name>Drucksache 100</name>
      <validity begin="1951" end="xxxx"/>
      <remarks/>
    </version>
    <version code="DV100">
      <name>Dienstvorschrift 100</name>
      <validity end="1951"/>
      <remarks/>
    </version>
  </register>
</registers>
```

* Each register would be identified by a file-wide unique 'id'
  attribute.

* Each version of a register would be identified by a domain-specific
  'code' attribute. This is not a unique string.

The purpose of this list is to constraint the strings for the same
register name, e.g.

  recommend "RL100",

not to use "Ril100", "KoRil100", "Ril 100"...

But how to use it?

For the following code snippets the <ocp> context is assumed.

* Should a railML file be meaningful without this list file?

  That would mean to refer to the /meaningful/ 'code' value, that is not unique.

  <designator register="RL100" entry="..."/>

* Should a railML file be meaningful only with knowledge of the list file, only in cases, where its attributes are used?

  That would mean to refer to the _unique_ but not /meaningful/ 'id' value.

  <designator registerRef="registers.xml#d1e51" entry="..."/>

* Should both possibilities be provided? If the list file is present, it may be looked up for further details, if not, the value is /meaningful/ anyway.

  That would mean to refer to both values.

  <designator register="RL100" registerRef="registers.xml#d1e51" entry="..."/>

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

Kind regards...
Susanne

Crosspost & Followup-To: railML.misc

--
Susanne Wunsch
Schema Coordinator: railML.common

Subject: Re: small issues on "register" and "tLineInfrastructureManagerCode"
Posted by on Fri, 07 Dec 2012 14:37:07 GMT
View Forum Message <> Reply to Message

Dear Susanne,

> Yes, you are right. I tried to find out, when the renaming from "DS100"

> to "RL100" was done, but I couldn't find it. That's the reason for the
> blank "begin" and "end" attributes. It was my intention to show this use
> case with disjunctive validities.

As far as I know: There never was a renaming of DS100 to RL100. Only last week I got a document from DB Kommunikationstechnik (!) where it is still named "DS100". I think these are two different naming philosophies at Deutsche Bahn.

So please do not try to create examples which assume there was a naming brake of DS100 and RL100. This could be misunderstood in a way that both values are allowed. I sent you an example with a naming brake from DR to DB and ThE, so possibly you could use that.

> I'm sorry for that.

No need to be sorry. We have to be thankful for your support. Thank you!

With best regards,
Dirk.