
Subject: Re: Mapping of code and abbreviation for ocps
Posted by on Mon, 26 Mar 2012 07:31:37 GMT
[View Forum Message](#) <> [Reply to Message](#)

We discussed about some aspects Of Simon's original post from one year ago again on last Monday, March 19th 2012.

There are normally several abbreviations and/or numbers for the same station, even in one country. So, the writing and reading software of a RailML file can have different external primary keys for the same station. During the evaluation process of the last year we got the new 'code' and 'tsi' as additional external primary keys but somehow we lost 'abbrev[i]ation' and 'number' as well. So where to put the abbreviations and EFA-numbers (needed in Germany) now? We cannot put both into 'code'.... So, we now have this certain amount of confusion which Simon did warn us against...

We now intend to allow a kind of enumeration of two-valued elements (elements with two attributes) per station. Each one can handle one external primary key of the station, which may be either a string (abbreviation) or a number.

I try to explain with a simple example (which is not valid RailML nor agreed in any kind, so key words or syntax may differ later):

```
<ocp ... name='Passau Hbf.' ...>
  <externalPrimaryKeys>
    <externalPrimaryKey register='DS100' value='NPA'/>
    <externalPrimaryKey register='DB640' value='Pa'/>
    <externalPrimaryKey register='UIC:80' value='80.7.33.165.9059'/>
    <externalPrimaryKey register='UIC:81' value='81.4.1744'/>
    <externalPrimaryKey register='EFA' value='8000298'/>
  </externalPrimaryKeys>
</ocp>
```

The first attribute 'register' means something like catalogue, index, directory. It shall be an enumeration of predefined values but this would mean, if someone needs a new register, he would need to call the Scheme Coordinator first. So I guess we have to allow a free string there. But we should strongly recommend and agree that each new 'register' has to be 'registered' at the Scheme Coordinator...

The second attribute 'value' has also to be defined as a string but may contain a number also depending on the 'register'. (This means, some 'registers' require a number which is not forced by XML.)

With this principle, there is no need to use 'code' for the abbreviation and/or the number. 'Code' will still be there since it is inherited but

(by recommendation) not specially to be used with OCPs.

I herewith apply for the following 'registers' to be defined from the very beginning:

- 'DS100' for the German "Betriebsstellenkürzel" (referring to the former "Dienstvorschrift"; I would not agree with "Richtlinie" since it is not a recommendation to use them but a directive!)
- 'DB640' which is the Austrian pendent to DS100 (DB="Dienstbehelf" - has nothing to do with Deutsche Bahn nor Dirk Bräuer).
- 'EFA' for the numbers used in some German public timetable databases and some RaiLML-reading programmes (EFA="elektronische Fahrplanauskunft" - or however they are called officially - Vasco know how).

The other values I used in the example above are really existing but we do not use them in RailML so far and I do not know the exact name of their origin.

It is intended to introduce the new principle with the first pre-launch RailML 2.2.

--

Erstellt mit Operas revolutionärem E-Mail-Modul: <http://www.opera.com/mail/>
