

Dear all,

@Stefan: Thank you for the example file for further discussion. I do have some questions/feedback:

1. additional attribute for 2.3

I believe the agreement of our meeting was to create a new and TSI specific attribute - e.g. 'tapTsiCode' - so that the original railML 2.2 data representation is not affected. See #2 for sample.

2. Use of the number and not string code

Should we actually use a string as a code or should we use the actual number value of the code? I would suggest to use the actual number code value instead of a string because the code could be interpreted directly if the default TAP TSI codes are used. Furthermore, it would allow for at least a limited amount of verification - i.e. number from 1 to 999 as a valid code reference instead of any string. A set of default mapping for the existing railML enumeration categories could be provided in the Wiki.

```
<formationTT>
  <passengerUsage>
    <!-- 100 seats in 1st class -->
    <places category="class1" tapTsiCode="4" count="100"/>
    <!-- 300 seats in 2nd class -->
    <places category="class2" tapTsiCode="5" count="300"/>
    <!-- Snack trolley available -->
    <service name="trolley" tapTsiCode="25" count="1"/>
  </passengerUsage>
</formationTT>
```

3. Documentation and versions

For me it is still not clear where the code list is 'published' and how different versions can be managed between a railML provider and consumer. Within the railML release there should be a codelist-schema xsd for the file but the actual code list. It would then be up to the provider/consumer to agree on an actual code list version - either online or provided within their interface process.

Regarding external code lists in general I have found a similar approach used by the ISO 20022 used by the financial industry ([https://www.iso20022.org/external\\_code\\_list.page](https://www.iso20022.org/external_code_list.page)):

"... Unlike other ISO 20022 code sets, the codes are not included in the message schema with the message element they type. The purpose of externalising these codes is to be able to update the code sets (for example, add new codes) without impacting the messages themselves and, hence, without requiring the development of a new version of the messages that use these code sets."

Best regards,

Philip Wobst

Am 21.09.2015 um 18:02 schrieb Stefan Jugelt:

```
> Dear all,
>
> as agreed during our railml timetable meeting last week I have created for
> the proposed attributes one example for the coding of places and services
> for a train part:
>
> Example:
> <?xml version="1.0" encoding="UTF-8"?>
> <!--Sample XML file generated by XMLSpy v2015 sp2
> (http://www.altova.com)-->
> <railml xmlns="http://www.railml.org/schemas/2013"
> xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
> xsi:schemaLocation="http://www.railml.org/schemas/2013 railML.xsd">
>   <metadata>
>     <organizationalUnits>
>       <railwayUndertaking id="CompanyCode_1080" code="1080"/>
>     </organizationalUnits>
>   </metadata>
>   <infrastructure id="inf-0815">
>     <operationControlPoints>
>       <ocp id="enee_8010085">
>         <designator entry="8010085" register="ENEE"/>
>       </ocp>
>       <ocp id="enee_8010205">
>         <designator entry="8010205" register="ENEE"/>
>       </ocp>
>     </operationControlPoints>
>   </infrastructure>
>   <timetable id="tt-0815">
>     <trainParts>
>       <trainPart id="TrainPart-0815-1">
>         <formationTT>
>           <passengerUsage>
>             <!-- 100 seats in 1st class -->
>             <places category="seatingFirstClass" count="100"/>
>             <!-- 300 seats in 2nd class -->
```

> <places category="seatingSecondClass" count="300"/>  
> <!-- Snack trolley available -->  
> <service name="trolley" count="1"/>  
> </passengerUsage>  
> </formationTT>  
> <ocpsTT>  
> <ocpTT ocpRef="enee\_8010085" ocpType="begin"/>  
> <ocpTT ocpRef="enee\_8010205" ocpType="end"/>  
> </ocpsTT>  
> <organizationalUnitBinding>  
> <railwayUndertaking ref="CompanyCode\_1080"/>  
> </organizationalUnitBinding>  
> </trainPart>  
> </trainParts>  
> </timetable>  
> </railml>  
>  
> Some further explanations:  
> 1. The code values for the rail:places/@category shall be stored in an  
> external code list, not in an enumeration in railML.  
> 2. The code values are available in the files "Codelist\_B.4.7161.xml"  
> (servies) and "Codelist\_B.4.9039.xml" (places).  
> 3. The enumeration for the accomodation classes in "tPlaceCategory" and  
> "tServces" are not needed anymore.  
>  
> Please check the example and provide your comments.  
>  
> Kind regards,  
>  
> Stefan Jugelt  
>

---