

Dear Milan,
Dear all,

I gave a presentation at the TT telco on 12.11.2020, where I tried to answer some of the questions.

Regarding the elements inherited from <train>, not all are actually desired (such as trainNumber, tafTapTsilD, cancellation). Ideally the <train> and <patternTrain> extend a common parent class that contain the shared elements and attributes. In railML2.4 nor the closest alternative was to extend the existing eTrain class used for <train>. On page 4 of the presentation linked above, a possible redistribution of elements and attributes defined in the tTrain and eTrain classes is shown. The resulting eTrain class is identical to railML 2.4.

The suggested <patternTrain> class uses <trainPartSequence>s to describe the train path in the same manner as for <train>s. Christian Rößiger suggested in the telco to at least define clearly (e.g. in the wiki) how to interpret the properties of a <trainPart> depending on whether it is referenced from a <train> or a <patternTrain>. Possibly, a new type of train parts should be created and used by <patternTrain>s. Even though this increases the number of changes in the schema, it may narrow down the scope and reduce the complexity. One example is that we could skip using full <operatingPeriods> and use only <operatingDay>s instead.

I think the new train number attributes suggested for trainGroups were meant to be used with individual trains and as part of the yearly timetable process. So I'm not sure they fit on <patternTrain>. We do however already have a use case for some kind of train number pattern. In the different planning software I have used, it is normally given as an initial number (e.g. 501) and an increment (e.g. 2, resulting in 501, 503, 505, ...).

It has also been mentioned that the use of xs:list is uncommon in railML. That is true, but it is already used for @coord in <geoCoord>, and was chosen because it is simple, compact and readable. When using a schema-aware parser, the attribute value will be split automatically into a numeric integer array, so no text parsing by the receiving software should be necessary. As element values (e.g. <interval>1200</interval>) are not used in railML, the alternative would be something like

```
<intervals>  
  <interval seconds="1200"/>  
  <interval seconds="2400"/>  
</intervals>
```

to replace interval="1200 2400". And in many (or most?) cases the value will be a single number.

Best regards,
Thomas
