

---

Subject: train parts

Posted by [Susanne Wunsch railML](#) on Wed, 02 Jul 2008 20:00:04 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Hello,

I try to explain, why we need some structure for distinguishing trains and train parts.

\* There are timetables for "destination coaches" ("Kurswagen" in German), that run as part of different trains from starting to destination station.

\* There are timetables for "multiple unit trains" ("Triebzüge" in German), that run some section together and split for further way, sometimes they join with other multiple unit trains.

\* In freight transit, carriages ("Güterwaggons" in German) run as part of different trains. Sometimes only the locomotive changes for some section.

\* The simplest case is a timetable for a whole train, that runs in one configuration without any rollingstock changes.

Furthermore there are additional constraints for the above defined cases:

+ Changes occur at different operating days, operating periods or timetable periods. (e.g. only on Mondays, only on days after a holiday, only during school holidays, only in winter timetable ...)

+ For some applications the train point of view is needed, with its timetable along the way and its train number. That represents the operational point of view ("Betreibersicht" or "betriebliche Sicht" in German). It is used by railway transportation companies, that calculate plans for staff and material.

+ For other applications the train part point of view is needed, with its detailed rolling stock definition and also the timetable along the way. That represents the user point of view ("Anwendersicht" or "verkehrliche Sicht" in German). It is used for communicating the "personal timetable" for passengers and goods, e. g. for booking systems or destination displays inside cars.

The same information can be defined from two points of view, so there is a risk for duplicating information in one file hence for allowing inconsistent data sets!

-> The documentation has to clarify the desired use of similar structure!

Some structure for the above discussed idea could look like following:

```
timetable
+- (timetablePeriod, operatingPeriod, ...)
+- trains
    +- train
        +- trainPartRefs
    +-trainPartRef
        +- (sections, times, refs to formation ...)
+- trainParts
    +- trainPart
        +- actualFormation
        +- (sections, times, ...)
```

The same in a more railML-way to define (no nested elements):

```
timetable
+- ...
+- trains
+- trainParts

trains
+- train

train
+- trainPartRefs
+- ...

trainPartRefs
+- trainPartRef

trainParts
+- trainPart

trainPart
+- actualFormation
+- ...
```

I hope, somebody can understand my statements. Feel free to ask, to comment and please DISCUSS the idea!

This idea isn't new to the railML community, Dirk Bräuer explained the need for it at 12th meeting in Dresden and I described the new structure at 13th meeting in Bern. You will find the minutes (in German) at <http://www.railml.org/de/public/conferences.html>

If users are interested in this development, I can help with an XSD for the idea.

Kind regards...  
Susanne

--

Susanne Wunsch  
Schema Coordinator: railML.common

---