
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

Subject: Re: train parts
Posted by [Joachim Rubröder railML](#) on Wed, 09 Jul 2008 12:21:36 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Susanne,

I agree, that the distinction between "operational view" and "commercial view" is an important missing point of the timetable schema.

Your trainParts would cover this problem, but there are some points I'd like to discuss about them:

- * You are creating a new sub structure called "trainParts" but I'm not sure that this is the atomic part of a train. Maybe there will be the need for trainPartParts next time.
- * The "train" can by now be further grouped to a trainGroup or an intervalGroup. For some users the "intervalGroup" is what he calls a train, for others it is the "trainPart"
- * A simple railML user (without operatingPeriods and formations) knows about its "trains" but doesn't want to have to deal with "trainParts" because there is no reason for him.

I therefore suggest to identify the most basic structure of a train as "train". Changing formations will of course be forbidden for this new "train".

In addition to this "train" structure, there could be several different groupings or high level understandings of a train like "commercialTrain", "operationalTrain" or "intervalTrain".

We will have to discuss if attributes like "trainNumber" or "trainKind" belong to the basic "train" or to the "operationalTrain"

What do you think about this approach? - Please comment!

Kind regards...
Joachim

--

Joachim Rubröder
Schema Coordinator: railML.timetable

Subject: Re: train parts

Posted by [Susanne Wunsch railML](#) on Tue, 15 Jul 2008 20:04:32 GMT

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

coord@timetable.railml.org wrote:

> * You are creating a new sub structure called "trainParts" but I'm not
> sure that this is the atomic part of a train. Maybe there will be the
> need for trainPartParts next time.

So let's define a "train" as something that is unique with its train number on a day and that goes as on "thing" on rails. That means that people, who handle the train doesn't know about its parts.

In order to avoid redundancy, the "train" knows about its sections and all according stopping (and passing) times and positions. That can be stations, signals, reference points or anything else.

Instead of full defining times and positions inside "trainParts", the "train" defines its "parts". That means, along a defined section the "train" runs without changing its formation. Data about "current formation" should be saved along the whole route, maybe only in points, where formation changes (for avoiding redundancy).

Example:

Some Train 18345 runs from A via B to C.

From A to B we've two cars, in B one car is decoupled, from B to C only one car runs.

A -->-- B ->- C

21---21 1---1 (1 and 2 are cars)

```
<train number="18345">
  <parts>
    <part id="1" from="A" to="C" posInTrain="1"/>
    <part id="2" from="A" to="B" posInTrain="2"/>
  </parts>
  <!--...timesAndPositions...-->
</train>
```

If somebody wants to describe the whole run of some part, he can define a "composition". In order to get detailed timetable for such a part, you have to follow the references.

Example:

Some destination coach ("Kurswagen" in German) t1 runs in train 18345 as

part "2" (unique id in train) and in train 12345 as part "b".

```
<compositions>
  <composition id="t1">
    <trainRef="18345" partRef="2"/>
    <trainRef="12345" partRef="b"/>
  </composition>
</compositions>
```

> * The "train" can by now be further grouped to a trainGroup or an
> intervalGroup. For some users the "intervalGroup" is what he calls a
> train, for others it is the "trainPart"

Interval trains means, that there is a "master train" and some "offset
trains". That has to be clarified in documentation.

> * A simple railML user (without operatingPeriods and formations) knows
> about its "trains" but doesn't want to have to deal with "trainParts"
> because there is no reason for him.

The above mentioned version helps simple users, which doesn't handle
with train parts. They can use trains as they are by now.

Thanks for your comments.
Feel free to discuss this approach.

Kind regards,
Susanne

--

Susanne Wunsch
Schema Coordinator: railML.common

Subject: Re: train parts
Posted by [Joachim Rubröder railML](#) on Fri, 18 Jul 2008 15:01:38 GMT
[View Forum Message](#) <> [Reply to Message](#)

I suggest to create an elementary train part structure called "train".
This "train" has no changement of formations. For most users it will be
the same "train" as before. There are different views and interpretations
of a train, but they all consist of these elementary trains.

- an "operational" train is the one seen standing along the track, which
is known by th signal box and shown on the graphical timetable.
- a "commercial" train is the one used by a passenger sitting on his
reserved seat, which is shown on the published timetable or planned in a
formation rostering.

Example:

Two formations (a) and (b) are driving together from A to B. We split the train in B. The first part (a) will go on to C and the second part (b) to D. The same scenario occurs at 8:00 and 9:00.

There will be 8 "trains" in the xml. Every system could deal with them:

8ABa, 8ABb, 8BCa, 8BDb and
9ABa, 9ABb, 9BCa, 9BDb

There are 4 trainGroups as "operationalTrain", with optional position information. This is for the signal box, a simulating programm or a graphical timetable:

{8ABa-1, 8ABb-2, 8BCa}, {8BDb} and
{9ABa-1, 9ABb-2, 9BCa}, {9BDb}

There are 4 trainGroups as "commercialTrain". These are the trains for the reservation system and the rostering plan:

{8ABa, 8BCa}, {8ABb-2, 8BDb} and
{9ABa, 9BCa}, {8ABb-2, 9BDb}

There are 2 trainGroups as "intervalTrains" for the conceptional timetable planner:

{8ABa-1, 8ABb-2, 8BCa, 9ABa-1, 9ABb-2, 9BCa} and
{8BDb,9BDb}

With this new structure, we could get rid of the "operation" element (same as an operational train) and the two grouping train attributes "trainGroupID" and "intervalGroupID".

There are still some open questions:

- Are there attributes, belonging to the trainGroup not to the train? Maybe the "operational" trainNumber is different from the "commercial" train number.
- Is the "operatingPeriod" and the "trainKind" fix for an elementary train? I'll think so, but this will lead to even more grouping.
- Are there more different views on a train?
- Does this cover all problems with the existing train structures?

More Ideas? More Comments?

Kind regards,
Joachim

--

Joachim Rubröder
Schema Coordinator: railML.timetable