Subject: Re: RFE for connection, DE:Anschluss
Posted by on Fri, 09 Nov 2012 18:38:54 GMT
View Forum Message <> Reply to Message

> Do you mean "Lok umsetzen" with "formation reverse"?

Of course not. Running ‘round with the engine does not reverse the formation, it changes the formation.

Concerning the meaning of <ocpTT>.trainReverse:

It simply tells that the train(part) changes the direction - no matter whether the formation changes, reverses or neither of both.

trainReverse with change of formation = e. g. running around with the engine
trainReverse with reversing of formation = train(part) of several MUs or push-pull train
trainReverse with neither of them = train(part) consists of a single vehicle (MU or engine)

This information is mainly intended for passenger information (systems) which sometimes print a sign like <-> to notify the passenger where the running direction of the train changes.

Concerning the meaning of <formationTT>.orientationReversed:

It has nothing to do with a train changing the running direction. It simply shall avoid the necessity to create each formation two times for both orientations. A train does not need to change its running direction for <formationTT>.orientationReversed:

Let’s say a train with the formation
1. propelling control car
2. 2nd class carriage
3. 1st class carriage
4. engine

runs all day between Airport and a place called Pirna. For the one direction the <formation> is fine, but for the other direction - so for half of the trains in total - the formation would have to be created a second time:
1. engine
2. 1st class carriage
3. 2nd class carriage

4. propelling control car

To avoid this, the attribute <formationTT>.orientationReversed can be used at every second train.

Please note: None of the trains do ever change its running direction during a single run - as in practice between Airport and Pirna.

Until now, there was no need to use <formationTT>.orientationReversed at a formation consisting of one vehicle only. This would have been paradox since one cannot change the order of a list of one element.

> I'm sorry, I don't see the difference between a train and a formation
> reverse.

Additionally, one should be aware that <ocpTT>.trainReverse semantically applies to the whole train while <formationTT>.orientationReversed applies to the formation of one <trainPart> only. So, there is another way to change the orientation of the formation of a whole train w/o <formationTT>.orientationReversed: Each vehicle forms its own <trainPart>, may be due to different operating days or so.

```
<train>
  <trainPartSequence>
    <trainPartRef ref=’TP1.1’ position=’1’>
    <trainPartRef ref=’TP2.1’ position=’2’>
    <trainPartRef ref=’TP3.1’ position=’3’>
  </trainPartSequence>
  <trainPartSequence>
    <trainPartRef ref=’TP3.2’ position=’1’>
    <trainPartRef ref=’TP2.2’ position=’2’>
    <trainPartRef ref=’TP1.2’ position=’3’>
  </trainPartSequence>
</train>
```

Obviously the train reverses between both <trainPartSequences> but there would be no <formationTT>.orientationReversed at none of the <trainParts> if each consist of one MU only.

(This refers to the current situation in RailML. It changes if we declare <ocpTT>.trainReverse obsolete and declare <formationTT>.orientationReversed to be used by definition as recommended in the previous post.)

Summary:
1) A formation running w/o reversing in one direction (“forward”):

<ocpTT>.trainReverse: not used
<formationTT>.orientationReversed: not used

2) A formation of several vehicles running w/o reversing in the other direction (“backward”):
<ocpTT>.trainReverse: not used
<formationTT>.orientationReversed: shall be used

3) A formation of several vehicles reverses direction w/o ‘running around’:
<ocpTT>.trainReverse: shall be used
<formationTT>.orientationReversed: shall be used

4) A formation of several vehicles reverses direction with ‘running around’ of the engine:
<ocpTT>.trainReverse: shall be used
<formationTT>.orientationReversed: cannot be used since the formation changes

To avoid no. #4, the engine may be put in an own <trainPart> so that #4 becomes “two times #3”. This reduces the total number of necessary formations by trend.

Hope I was able to clarify the difference between trainReverse and orientationReversed.

Best regards,
Dirk.