
Subject: Re: How to model ETCS BL2 speed restrictions and gradients, in railML v2.x

Posted by [Thomas Nygreen JBD](#) on Tue, 08 Jan 2019 21:07:50 GMT

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

Quote:

- > For data to be useful for ETCS the slope of each part of the
- > track must be known.
- > I question whether both these approaches should be
- > supported:
- > 1. Deduce from connected tracks. The last gradientChanges in
- > routes through the network ending up in the track in
- > question would specify the slope. A gradientChange would
- > then only be defined when contradicting values are deduced
- > from the different routes.
- > 2. Always define a gradientChange at pos=0 of a track, even
- > if the value is unchanged compared to the connected tracks.
- > Or in other words limit the effect of a gradientChange
- > element to the track it is defined in.
- >
- > Approach 1. (above) would require processing of both readers
- > and writers of the railML data. I propose that approach 2.
- > (above) is promoted to be the only valid approach.

Both approaches have their advantages and disadvantages. However, it should be used consistently in import and export interfaces. So, we need a clarification based on "best practices" here. This, for sure, has substantial effects on railML modelling, because the current railML version leaves it up to the user how to use the <*Change> elements.

@all: Do you prefer approach 1 or approach 2?

Approach 1 was chosen for railML2.4nor, so it is the preference of the Norwegian sector.

Quote:

- > @slope is given in relation to @pos and @dir so that: if
- > standing at @pos, looking along the track in the direction
- > of @dir, a positive value of @slope will be considered as
- > seeing that the track continues uphill.
- > The slope in down direction should for ETCS purposes be
- > allowed to have a different value than in up direction.
- > But if the slope value in down direction is omitted, then
- > the inverse value of the slope in up direction is assumed
- > (as slope value in the down direction).

Using the @dir attribute allows to implement direction dependent gradient profiles. The slope value (positive or negative) shall be interpreted like you mentioned it: in the orientation defined by the

@dir attribute.

This would contradict the current definitions, which is that the value is in track direction. This is not documented on the wiki page for gradientChange, but it is clearly stated for radiusChange: "the parameter radius describes the new value of the radius that is valid from there into the direction of track orientation".

If you do not have different gradient profiles for the two directions, you should not use @dir. The way that the documentation defines @dir, having only gradientChanges with @dir="up" would mean that you do not know anything about the gradients in the "down" direction. There is a separate thread on what we should do with @dir.

Quote:

- > 4. speedChange
- > Similar to gradientChange, whether or not to deduce values
- > from speedChanges of connected tracks, is a relevant
- > question.
- > And I propose to limit the effect of a speedChange element
- > to the track it is defined in.
- > Additionally I propose that a track must be fully covered by
- > speedChanges, unless an infraAttrGroupRef is defined.

Like with the <gradientChange> issue above, we should come to a unique and consistent solution. Therefore, I would like to ask the community again: which approach do you prefer?

* option 1: track features (gradients, speeds, ...) are valid even beyond the end of track

* option 2: track features are only valid within the range of the track; every track needs to have appropriate <*Change> elements at the begin and the end.

Again, approach 1 was chosen for railML2.4nor, so it is the preference of the Norwegian sector.
