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

Posted by [Jürgen Strandberg](#) on Fri, 07 Dec 2018 08:33:18 GMT

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

For clarity, here is the content of the word document attached in the initial posting:

Posting on forum.railml.org about ETCS profile data

2018-11-29

In search of a best practice for, in railML, modeling ETCS BL2 speed restrictions and gradients (slope) for the complete track.

These elements and attributes can be used to model speed and gradient data of a track:

1. gradientChange
 - a. @slope
2. infrastructure.infraAttrGroups.infraAttributes
 - a. @id
 - b. speeds.speed@vMax
 - c. speeds.speed@etcsTrainCategory
 - d. speeds.speed@profileRef - to define dependent on e.g. axle load
3. track.infraAttrGroupRefs
 - a. infraAttrGroupRef@ref
4. speedChange
 - a. @pos
 - b. @dir
 - c. @vMax
 - d. @etcsTrainCategory
 - e. @profileRef
5. speedProfile
 - a. @maxAxleLoad

1. gradientChange

This seems to be the only way to enter slope of track.

Description of @dir value range on wiki.railml.org is incorrect. Only up and down are valid according to schema, and that is what the wiki should say too.

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.

@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).

2. infrastructure.infraAttrGroups.infraAttributes

Provides a way to define speeds (static, train category dependent and axle load dependent) that can be referenced for a whole track as default values. See speedChange below for suggestions regarding @vMax, @etcsTrainCategory, and @profileRef.

3. track.infraAttrGroupRefs

Provides a way to reference a number of predefined speed elements that make up the default values for a whole track. These defaults are then possible to override with speedChanges. When an infraAttrGroupRef is defined all speedChanges stating that data can be removed. A speedChange that is not immediately followed by another speedChange needs to be terminated with an additional speedChange that sets @vMax=end (@vMax=999), and then the default values of the track shall be valid.

The type of element to reference with track.infraAttrGroupRefs is incorrectly documented in railML 2.2, however there is a Key/KeyRef definition that correctly describes the reference, which should make this a valid construction.

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.

Wiki should also describe how to represent the end of a speedChange: with @vMax=999 (for railML v2.2) or @vMax=end

Description of @dir value range on wiki.railml.org is incorrect. Only up and down are valid according to schema, and that is what the wiki should say too.

Static speed

Is defined with the @pos, @dir, and @vMax values.

Train category dependent speed

Is defined with the @pos, @dir, @vMax, and @etcsTrainCategory values. Setting @etcsTrainCategory defines that the speedChange in fact is a train category dependent speed.

Axle load dependent speed

Is defined with the @pos, @dir, @vMax, and @profileRef values.

To make a speedChange dependent on axle load, a speedProfile element (that specifies @maxAxleLoad) is referenced with the @profileRef value.

5. speedProfile

Provides a way to define certain criteria (such as @maxAxleLoad of train) that need to be met for a referencing speedChange or speed to be valid.

Additionally @influence must be set to tell which speedChange+speedProfile to give precedence when several are valid for a specific train.

Wiki should describe precedence among multiple speedProfiles using @influence
