
Subject: [railml3.1] speed profiles and @etcsTrainCategoryNumber
Posted by [Thomas Nygreen JBD](#) on Mon, 24 Dec 2018 15:55:38 GMT
[View Forum Message](#) <> [Reply to Message](#)

The categorisation of speed profiles has been greatly improved from 2.x to 3.1. Still, (at least) one issue remains:

speedProfile/trainType/@etcsTrainCategoryNumber is currently implemented the same way in 3.1 as in 2.22.4, namely as an integer in the range 0-15. The wiki points to the value table for NC_TRAIN in ERA_ERTMS_040001 "Assignment of values to ETCS variables" v1.21 from 2016-10-20. The document is now in v1.26, released just a few days ago, but the relevant content is not changed. As is stated in the same document, since v1.4 of 2011-04-08, "The variables previously referred in sections [...] 3.7 (NC_TRAIN) [...] considered as baseline dependent [...]". However the list of their previously assigned values is kept in this document for recording purpose." (paragraph 1.3.4). So, even if this document is still actively maintained, it has not been a reference for train categories since 2011, and should not be used as such by railML.

Train categories used in ETCS are defined on at least two levels. First, on a legislative level, through of the "technical specification for interoperability relating to the 'operation and traffic management' subsystem of the rail system in the European Union" (TSI OPE). Since 2011, the train categories have been specified in its Appendix A, maintained by ERA. Second, in the System Requirements Specification (SRS) for each baseline of ETCS. As far as I can determine, each version of TSI OPE Appendix A is based on the "current" baseline of ETCS, so there is some synchronisation of the two.

Currently, the TSI (Annex B on page 79 in Appendix A) specifies 18 operational train categories. The categories are not numbered, but labelled PASS 13, TILT 17, FP 14 and FG 14 respectively, each corresponding to a combination of type of train (passenger, tilting passenger, freight), type of brake (P, G) and cant deficiency (11 levels from 80 to 300 mm).

The SRS subset 26 chapter 3 (current version 3.6.0 of 2016-05-13) defines how a train should choose the correct Static Speed Profile (SSP).

The cant deficiency categories and passenger/freight categories that were previously mixed in ERA_ERTMS_040001 have now been separated into two different sets of categories:

3.11.3.3.1 A maximum of 31 train categories is defined to match the SSP categories. 16 "Cant Deficiency" train categories and 15 "other international" train categories.

3.11.3.3.2 A train shall always belong to one and only one "Cant Deficiency" train category and may optionally belong to one or more "other international" train categories.

Consequently, there is no atomic "ETCS train category number". Suggestion: delete @etcsTrainCategoryNumber

If we move to chapter 7, it defines the packets used to transmit information to and from the trains. Here, the variables NC_CDTRAIN (integer 015) and NC_TRAIN (bitset, 16 bits) describe respectively the "cant deficiency" and "other international" train categories that a train belongs to.

The corresponding NC_CDDIFF (integer 015) and NC_DIFF (integer 015) are used to specify which categories an SSP applies to. Multiple categories may be given for one SSP.

The current options for the variables NC_TRAIN and NC_DIFF are

- * Freight train braked in "P" position
- * Freight train braked in "G" position
- * Passenger train
- * 13 currently unused spare values/bits

The distinction between freight and passenger trains is already covered by speedProfile/trainType/@type, but the "type of brake" (P/G) is not covered in railML 3.1.
Suggestion: add @brakePosition

The current cant deficiency options for the variables NC_CDTRAIN and NC_CDDIFF already covered by speedProfile/trainType/@cantDeficiency. However, I am not sure if the current restriction to a value in the range 80300 is necessary. I think we should either just require a positive integer or use the enumerated set of values from the SRS.

Best regards and happy holidays!
Thomas
