
Subject: Aspects of timetable 3.0

Posted by [Burkhard Franke](#) on Thu, 16 Oct 2014 17:26:31 GMT

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

some reminder of the discussion at the railML-conference in Paris:
getting rid of optional elements.

Problem: the railML standard features a lot of flexibility, as it covers a wide range of use-cases. That's basically fine, but it makes it hard to write import interfaces and to validate files as you never know what to expect...

The idea to ease this problem is to predefine "use-cases".

Favourite example of my colleague Bernhard are the timetable and operating periods. It is required to have them, everything else is optional...

The "use-case"-approach defines several sets of mandatory and optional data:

use-case 0: no information on an operating period (for schematic/long-term planning)

use-case 1: startDate and bitmask mandatory, bitmask covers seven days to specify a sample week (to define operating patterns in a schematic timetable), no holidays, deviances or offsets

use-case 2: real timetable: start date, bitmask, holidays, ... and all the complex stuff with deviances, specialServices ...

use-case n: user-defined description of a timetable/operating period based on the klingon calendar ;-)

The use-cases can also be applied to other elements, for instance it could ease the discussion on vehicle data in the timetable (uc0 - no vehicle information; uc1 - only sample vehicles; uc2 - detailed technical data ;...)

This approach (maybe the term "use-case" is not the best) will limit flexibility or rather guides the flexibility in an orderly manner. This is meant to help create a better structured railML-timetable in a 3.x version.

Comments welcome
