

>> confusing.

>>

>> If nobody disagrees with good reasons I would file a Track ticket for

>> "deprecating" the "both" enumeration value from the "dir" attribute in

>> all "*Change" elements.

>>

>> New or already used practice is to define seperate elements for each

>> running direction. The above example would be:

>>

>> <speedChange pos="10" dir="up" vMax="40"/>

>> <speedChange pos="200" dir="up" vMax="80"/>

>> <speedChange pos="2xx" dir="down" vMax="80"/>

>> <speedChange pos="200" dir="down" vMax="60"/>

>>

>> That means that the semantic for the same XML content changes. That is a

>> very hard cut that can't be recognized by any parser!

>>

> It is a question of your position. Do I have to combine two speed

> restrictions which are shown at the same position to differnt dircetions

> into one element? I do not think so. So I can keep both of them in two

> elements and do not do any mistake in RailML-useage. So for me the cut looks

> not so heavy.

I agree with you, Carsten. The two interpretations of the usage of speedChange elements are not a cut within the railML infrastructure syntax. Nevertheless, I support the idea of you, Susanne, to mark the "both" value of the direction attribute as being deprecated. From the user's point of view, speedChange points need to be referenced with a specified direction (up or down) on the track segment.

>>>> If you have a speed restriction along a bridge, you may define different

>>>> <speedChange> elements in each direction refering to the same speed

>>>> profile with different speed aspects. The<speedChange> elements already

>>>> need the direction attribute. Why do we need to duplicate it?

>>>>

>>>

>>> Because of more clearance.

>>> In my sight a speedChange should be a child of a speedProfile.

>>> <speedProfile direction="up" ...>

>>> <speedChange position="0" speed="60" .../>

>>> <speedChange position="5" speed="120" .../>

>>> <speedChange position="123" speed="100" .../>

>>> <.../>

>>> </speedProfile>

>>>

>>> So you can see: the speedChanges if you run the track in one direction in

>>> a

>>> line.

>>> But this structure requires a break in downward compatibility which is
>>> not
>>> leagl at the moment.
>>
>> The compatibility break is only one side effect of the above mentioned
>> example. Another aspect is that speed profiles are much more general
>> definitions than the detailed speed changes along the tracks.
>>
>> You may define some speed profiles for different axle weights that are
>> referenced at multiple tracks. If you would define these speed profiles
>> at each track you get heavy redundancies and may "lose track of the big
>> picture".

I really appreciate the discussion about a possible speedProfile element, because it questions the current track-based structure of the whole infrastructure schema.

Focusing on the speedProfile idea, I see the same problem like Susanne: a direction attribute within a speedProfile is a redundancy to the direction attribute of the speedChange. Furthermore, it is not even unique in usage, e.g. when a speed profile contains speedChange points of several tracks, which are oriented differently. In other words: while the speedChange points can refer to the direction of the track they are placed on, a speedProfile cannot do likewise.

Defining speedChange points as "children" of a speedProfile brings up the question of the dimension of a speedProfile: Where does it begin and where does it end? From a user's point of view, this aspect depends on the application and by interpreting the speedChange points, a speedProfile that matches the special requirements of the application can be generated.

Kind regards

Christian Rahmig
railML.infrastructure coordinator
