
Subject: Re: the use of @dir in railML.

Posted by on Fri, 19 Oct 2018 08:34:05 GMT

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

Dear Thomas,

I found your composition on @dir very refreshing and fundamental and welcome the detailed work. Hopefully, it was not for nothing... ;-)

In many aspects I see a kind of rather philosophical background in it which we also often had and have in <timetable> discussions: Whether to make railML rather "flexible" or rather "strong, exact".

To make it short: I don't have general objections against your suggestions. I also would welcome to make @dir more exact.

As often, there is unfortunately a "but":

> * DEPRECATE @dir for border and trackCircuitBorder

I can imagine borders which need a direction. As I mentioned earlier in a post with Torben about station limits, in Germany we have direction-dependant station limits. For instance, from the view of interlocking, for a train entering a station, the limit is the home signal (Einfahrsignal). For a train of the opposite direction, that leaves the station, the limit is the shunting-limit marker (Rangierhalttafel Ra10).

So, if a <border> can for instance encode the interlocking or operational station limits, we may need a @dir attribute.

Concerning <speedChange>s:

> BUT, it is important what the common practice is.

For many years, we use <speedChange> in the way you describe it, with @dir=up or =down dependant to the validity direction of speedProfiles:

- We always encode speedProfiles in the direction of (relative) mileage, means in order of raising @pos attributes, independent of the validity direction (of travel) of the speedChanges.
- In cases of a speedProfile is valid for the =down direction, our <speedChange>@speed value is valid from that @pos until the previous (!)(not next!) <speedChange>.
- The <speedChange>@dir never encodes the validity direction of the <speedProfile>. It only says in which direction (next or previous element) the @speed is valid. The <speedProfile> has its own direction and can be valid for both directions.
- This is documented in our own documentation of FBS-railML-interface, which is a kind of extension and concretion of the general Wiki.

> Suggested change:

- > * Document what part of the track the new values apply to
- > when using the @dir attribute

I can do it if Christian Rahmig agrees and/or if there will be no objections here.

> Suggested changes:
> * DEPRECATE @dir for bridge, levelCrossing, platformEdge,
> serviceSection (?) and tunnel

I think is is a misunderstanding:

The intention behind @dir was never to encode that the tunnel or bridge is not visible for the other direction.

The intention may have been:

<tunnel @pos=1234 @length=200 @dir=up> = a tunnel from km 1,234 to 1,434

<tunnel @pos=1234 @length=200 @dir=down> = a tunnel from km 1,034 to 1,234

This would fit in a certain way to the way <speedChanges> are to interpret from @dir.

However, I have no objections against killing this redundancy and deprecate @dir from bridges, tunnels etc.

With best regards,
@Dir_k.
