

---

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

Posted by [christian.rahmig](#) on Fri, 19 Oct 2018 14:42:10 GMT

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

---

Dear Thomas, dear @Dir\_k ;-)

Am 19.10.2018 um 10:34 schrieb Dirk Bräuer:

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

I agree with Dirk: Thomas, your notes on the @dir topic are very detailed and useful for further railML (2 and 3) development. Thank you!

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

I take your advice very serious for railML 3.1 implementation. The railML 2 attribute @dir is named @applicationDirection in RTM and railML 3.

>> \* 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 (Einfahrtsignal). 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.

And what about track circuit borders, where there is a track circuit only on one side (like we have it in the Simple Example at signal 69A)?

The attribute @applicationDirection has been used there to define on which side(s) of the track circuit border track circuits can be found.

Do you consider this being logical?

> 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.

In railML context, `<speedChange>` elements may have a direction while `<speedProfile>` don't have this attribute. However, Dirk is right when he says that `<speedProfile>` elements have a direction information, too. It is given implicitly by the `<speedChange>` elements referencing this `<speedProfile>`. So, if elements `<speedChange>@dir=up` and `<speedChange>@dir=down` reference the same `<speedProfile>`, this `<speedProfile>` is de-facto valid for both directions.

>> 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.

Absolutely no objections from my side! I am thankful for any contribution. I suggest to do it on the discussion page of element `<speedChange>` in the railML wiki (<https://wiki.railml.org/index.php?title=IS:speedChange>). Once, we have an agreement on the final solution, we may integrate it as "Best Practice" on the main wiki page of `<speedChange>`.

>> Suggested changes:

>> \* DEPRECATE `@dir` for `bridge`, `levelCrossing`, `platformEdge`,

>> `serviceSection` (?) and `tunnel`

>

> I think it 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.

For final answering this question I would like to receive more feedback from the community. Do we want to introduce a rule that says: "infrastructure elements with length have to be defined always in direction of the track orientation (increasing `@pos`)"? If there is a majority supporting this approach, we may think about an implementation (in railML 2.5?).

Best regards  
Christian

--

Christian Rahmig - Infrastructure scheme coordinator

railML.org (Registry of Associations: VR 5750)

Phone Coordinator: +49 173 2714509; railML.org: +49 351 47582911

Altplauen 19h; 01187 Dresden; Germany [www.railml.org](http://www.railml.org)

---