
Subject: Re: Panels in general

Posted by [Susanne Wunsch railML](#) on Fri, 18 Jan 2013 11:49:06 GMT

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

Dear Christian, Dirk and others,

Christian Rahmig <coord@infrastructure.railml.org> writes:

> Am 03.12.2012 16:17, schrieb Dirk Bräuer:

>>> - A <speed> signal information further contains the parameters

>>> "kind" (values: announcement, execution and end) and "speedChangeRef".

>> A list of <speedChangeRefs> please. (Several speed profiles, each

>> referencing the same <speedChange>.)

+1, implemented. [1]

>> I suggest not to distinguish between "execution" and "end" because of

>> this may depend on the speed profile. Rather, an attribute

>> /kind/="announcement"|"execution" should be enough also for the "ends".

+1, to be implemented.

>> Additionally, we need both the "valid for head of train" (not the same

>> as /kind/!) and "virtual" information as well, as already discussed and

>> agreed.

>

> The attribute @virtual already exists for the <signal> element and

> does not have to be declared again in the sub-element.

+1

> For the information about the point of validity of the signal/panel, I

> added the attribute @trainRelation to the signal's sub-element

> <speed>. Probably, it is even useful to implement the attribute

> @trainRelation in the <signal> element as it might be of interest for

> other types of signals as well.

Disagree. The @trainRelation is already defined in the 'stopPost' and 'speedChange' elements. The 'trainRelation' of a "speed signal" may be found following signal/speed/@speedChangeRef.

What other kinds of signals may need a @trainRelation?

>>> - A <milepost> signal information further contains the parameters

>>> "shownValue" and "realValue".

>> Is "realValue" the same as the /pos/ of this element at the track, isn't

>> it?

+1

>>> - A <braking> signal information further contains the parameter
>>> "trackConditionRef", which refers to a <trackCondition> where the
>>> attribute "type" allows for the values 'nonStoppingSection',
>>> 'noRegenerativeBraking', 'noEddyCurrentBraking',
>>> 'noMagneticShoeBraking'.
>> I suggest to use a brake type already existing (tVehicleBrakes or
>> such) and an attribute declaring "not for vehicles with the following
>> brakes...".

-1, there are no signals/panels next to the track that define the possible braking characteristics for this track. But there are sometimes panels for such rough characteristics like the list above (copied from the ETCS specification for "track conditions").

>> Please do already think at the German places for "Betriebsbremsung" or
>> "Zwangshalt" - should we use such a <signal> there, possibly with
>> /virtual/=true'? If so, we would also use a /type/ therefore.

>
> Don't we have implemented the <speedChange> attributes @mandatoryStop
> and @mandatoryBraking for that? Consequently, we need to add a
> reference from a braking signal to a <speedChange> element as we
> defined it for all speed signals.

I'm a bit confused about this.

Where should one define the "mandatory stop" in front of a level crossing?

trackElements/speedChanges/speedChange/@mandatoryStop and
ocsElements/signals/signal/speed/speedChangeRef/@ref

or

trackElements/speedChanges/speedChange/@mandatoryStop and
ocsElements/signals/signal/braking/speedChangeRef/@ref

I would not add an subelement 'speedChangeRef' to signal/braking, but refer to "signal/speed/speedChangeRef" for the mandatory stop and braking at the "speed/braking" wiki page.

Does somebody should define a 'virtual' signal if the characteristic is already defined somewhere else?

I mean that the 'speedChange' definition may be enough, the 'signal/speed' could be only used in case of "known speed panels".

There is already @signalised in the element 'speedChange' that has the

same meaning like `signal[@virtual=true]/speed`.

I would welcome to clarify the use cases for
`speedChange[@signalised=false]` and `signal[@virtual=true]/speed`.

Any comments appreciated.

Kind regards...
Susanne

--

Susanne Wunsch
Schema Coordinator: railML.common
