Subject: Release triggers for routes in railML 3.2
Posted by Terje Nordal on Thu, 14 Dec 2023 15:25:57 GMT
View Forum Message <> Reply to Message

Hello railML experts,

When modeling routes in railML 3.2, Bane NOR need the ability to define a trigger point for route release, typically a train detector that, when passed by the train, triggers the release of a train route. Up until recently we have used IL:routeReleaseGroupRear to model this, but have since found the description "One or more TVD sections as part of the route which can be released in a group in rear of passing train.". Since it refers to the release of TVD sections, and not routes, we think it does not fit our needs perfectly. We have also not found any other way to model it in railML 3.2. Is there something we have missed, which could be used to model it? Alternatively, are there/should there be plans to add this to railML 3.3?

For context, see the attached image where the route from MB 1 to MB 2 is released at the passing of train detector between TVD section 2 and 3, ie. when TVD section 2 is vacant.

As an addition: A similar trigger exists in railML 2.4nor; @releaseTriggerRef. Part of the description on this is "The trigger position used for releasing the complete route (and thus its overlap) while the train still occupies a track section(s) on the route.". Apart from the part of also releasing the overlap, which would not happen in our ERTMS system, this is exactly what we are trying to model.

File Attachments

1) Release route.png, downloaded 52 times

Subject: Re: Release triggers for routes in railML 3.2
Posted by Jörg von Lingen on Mon, 18 Dec 2023 06:11:51 GMT
View Forum Message <> Reply to Message

Dear Terje,

the intention for modelling route release is probably not so obvious but based on philosophy of conventional interlockings.

1) IL:routeReleaseGroupRear
Here the information is put for releasing parts of the running path of the route before the release trigger is reached. Especially some legacy interlockings do route release not per each element when the train passed it but using kind of intermediate triggers to release a defined group of elements at once.

2) final route release
To model the release trigger for overlap and final route release the IL:overlap element is intended. There are cases in conventional interlockings where routes do not have a physical overlap. But even then the overlap (with length 0) is modelled having all the information for route release.
The IL:overlap element is available in IL:routeExit and in IL:destinationPoint.

Best regards,
Joerg v. Lingen - Interlocking Coordinator

Subject: Re: Release triggers for routes in railML 3.2
Posted by Terje Nordal on Thu, 01 Feb 2024 12:19:10 GMT

View Forum Message <> Reply to Message

Dear Joerg,

Sorry for the late reply on this topic. Also, sorry for the amount of text in this post. I wanted to give some system background to

Bane NOR's use case for railML is, first and foremost, ETCS lines. These probably differ a bit from the conventional interlocking principles that (naturally) seem to be used as a basis for railML. So first, just a quick explanation on how release of train routes will work on ETCS lines in Norway (at least in the current state). In case of no specific design for release of train routes, the following principles are generic functions (taken from our interlocking guidelines):

- Objects in a train route and its flank protection are released when related TVP section is passed by the entire train.
- Objects in FS/OS-route and its flank protection are released when a train in the route has stopped and is deregistered.
- End point of a train route, that is extended with another route, is released when it is passed by the train's last axle.

Comments: A train route can release when train has passed TVP section specified in Ch 8.2.4. Releasing train route [Utløsning togvei], see the below points on specific release after a passed train detector.

- End point of a train route, at the border to a non-interlocked area, is released when it is passed by the train's last axle.
- FS/OS-route and subsequent overlap is released independently of each other when they are released by a train.
- FS/OS-route and subsequent overlap are released simultaneously when they are cancelled by command from the train dispatcher.
- An end point (signal) is released when both FS/OS-route to it and the overlap after it is released.

In addition there are separate rules for release of overlap in the generic applications, that are defined as follows (and do not necessarily correspond with the release of a train route to the signal that has the overlap):

- An overlap is not removed when FS/OS-routes are extended but is overtaken by the extended route.
- An overlap is released and release speed is set to 0 km/h when the last TVP section ahead of end point is occupied, and the train has stopped.
- An end point without an overlap is released when passed by the trains last axle. The release speed is kept even though the train stops ahead of the end point, as long as the overlap is not

released.

In addition, we are able to define specific train detectors for release of the entire train route (also the remaining part of the route in question) as explained in the point referring to Ch. 8.2.4 above. In such a case, the following applies:

- The train route is released without time delay.
- All train routes passing this axle counter in the specified direction are released.
- The overlap is still released using the generic principles above, typically when the last TVP section in the respective route is occupied and the train as stopped.

Moving on to the parameter discussion:

Most of our signals have overlaps of XX meters. Additionaly, since train route release and overlap release are not occurring at the same time in our system, it does not make sense to connect a train detector for route release to an overlap in railML. This means your option 2) is not optimal for our functionality.

Option 1) is a bit closer, I think, depending on how one interprets it. My understanding is that IL:routeReleaseGroupRear is meant to indicate the release of parts of a train route, and not the entire route (which would be the case in our system). One could say that, with the knowledge of how the specific signalling system works, such a parameter may be used to indicate the release trigger. And that the knowledge of the system functionality itself would help to know what is released and not. This may be the intention of the IL part of railML in general? If not, I think this option also does not cover our use case perfectly.

Either way, I think the closest railML parameter I have found to our system functionality is the @releaseTriggerRef for railML 2.4nor, as I mentioned in the original post. The only difference being whether the overlap is released at the same time.

Subject: Re: Release triggers for routes in railML 3.2
Posted by Jörg von Lingen on Tue, 27 Feb 2024 04:30:12 GMT
View Forum Message <> Reply to Message

Dear Terje,

thanks for your the description of your rules and background. Let me add
the following:

1) Objects in a train route and its flank protection are
released when related TVP section is passed by the entire
train.
That's the assumed standard behaviour why it does not need extra
definition. Refer https://wiki3.railml.org/wiki/IL:routeReleaseGroupRear.

<routeReleaseGroupRear> is forseen the cases where elements are released
from route as a group by extra command.

2) final route release by train
If route release shall be independent of any associated overlap a route
release trigger can be added to the <routeExit> referring to the trigger
position with mode information (occupation, dispatcher, trainStandstill).

--
Dr.-Ing. Jörg von Lingen - Interlocking scheme coordinator

Subject: Re: Release triggers for routes in railML 3.2
Posted by heidrun.jost@thalesgroup. on Tue, 12 Mar 2024 15:05:11 GMT
View Forum Message <> Reply to Message

Hello Terje and Jörg,

from Thales' perspective, for the TMS Norway project, for some routes we also need the route release trigger information as a reference to an element (either a track or a point).
That's why we would like to support the proposal from Terje.

Best regards,
Heidrun

Subject: Re: Release triggers for routes in railML 3.2
Posted by Terje Nordal on Wed, 13 Mar 2024 12:05:19 GMT
View Forum Message <> Reply to Message

Hello again,

The suggested addition of a route release trigger for <routeExit> seems to fit our use case very well. Thank you for the feedback.

Best regards,
Terje