Subject: Re: railML 2.3 infrastructure extension proposal route information at signal
Posted by Joerg von Lingen on Fri, 24 Aug 2018 08:57:02 GMT
View Forum Message <> Reply to Message

Dear Torben,

thanks for your extensive proposal. Within railML3.x IL we have the possibility to define train routes:

```
<rail3:knowsRoute name="Route_68N1_69A" id="rt_sig02_sig04" locksAutomatically="false"
delayForLock="PT1S">
 <rail3:handlesRouteType ref="rt_main"/>
 <rail3:isActivatedBy name="activation Route_68N1_69A" id="rt_act01" delayForLock="PT2S"
   automaticReleaseDelay="PT5S">
  <rail3:activationSection ref="A02T"/>
 </rail3:isActivatedBy>
 <rail3:requiresPointInPosition name="pt01 in left" id="rp_pt_swi01_li" inPosition="left"
   xsi:type="rail3:SwitchPointAndPosition">
  <rail3:refersToPoint ref="pt_swi01"/>
 </rail3:requiresPointInPosition>
 <rail3:routeEntry name="Start 68N1" id="rts_68N1">
  <rail3:refersTo ref="mb_sig02"/>
  <rail3:nonReplacement ref="A68W02T"/>
 </rail3:routeEntry>
 <rail3:hasReleaseGroup ref="prt02"/>
 <rail3:hasReleaseGroup ref="prt03"/>
 <rail3:hasReleaseGroup ref="prt04"/>
 <rail3:hasReleaseGroup ref="prt05"/>
 <rail3:hasReleaseGroup ref="prt06"/>
 <rail3:routeExit name="Dest 69A" id="rtd_69A">
  <rail3:refersTo ref="ls_sig04"/>
  <rail3:hasDangerPoint ref="dp01" />
  <rail3:hasOverlap ref="ov01" />
 </rail3:routeExit>
</rail3:knowsRoute>
<rail3:knowsOverlap name="Overlap 69A-P2" id="ov01" overlapValidityTime="PT60S"
speedInOverlap="0.0">
 <rail3:activeForApproachRoute ref="rt_sig02_sig04"/>
 <rail3:requiresAssetInPosition mustOrShould="should" proving="oneOff">
  <rail3:relatedAssetAndState xsi:type="rail3:SwitchPointAndPosition" inPosition="left">
   <rail3:refersToPoint ref="pt_swi02" />
  </rail3:relatedAssetAndState>
 </rail3:requiresAssetInPosition>
 <rail3:isLimitedBy ref="tde07"/>
 <rail3:overlapRelease name="ov01 Release" id="ov01_rl">
  <rail3:releaseTriggerSection ref="X02T"/>
  <rail3:overlapReleaseTimer timer="PT60S"
overlapReleaseCondition="startTimerUponOccupation" />
 </rail3:overlapRelease>
```

</rail3:knowsOverlap>

Route topology
@name [datatype: string] for example "A-1-L"
# Route name with @name

@routeEntry [datatype: ref to the <signal> starting the route]
# Route entry with <RouteEntry>

@routeExit, [all datatype: ref to any element ending the route usually a signal]
# Route exit with <RouteExit>

@routeVia/@switchAndPosition
# Route path definition with <requiresPointInPosition>

@setTime [datatype: time in seconds]
# Route name with @delayForLock

Speed profile(s) for the route
@proceedSpeed [datatype: integer in km/h]
# Speed for route in <aspectRelation> with @Vexpecting and @Vpassing
# additional reference to infrastructure with <signalsSpeedProfile> in <aspectRelation>

@releaseSpeed [datatype: integer in km/h]
# per signal @releaseSpeed

@approachSpeed [datatype: integer in km/h]
# per signal @approachSpeed

@approachPoint
# Route activation with reference in <activationSection>

Overlap in connection with the route
@overlapRef.
# Reference for route in <hasOverlap>

@overlapValidityTime [datatype: time in seconds]
# for <Overlap> in @overlapValidityTime

@speedInOverlap
# for <Overlap> in @speedInOverlap

@overlapReleaseTimer [datatype:ref]
# for <OverlapRelease> in <overlapReleaseTimer>

@overlapReleaseTimerHead [boolean]
# for <OverlapRelease> with combination of <releaseTriggerSection> and
@overlapReleaseCondition

Thus all items can be modelled in railML3.x

Regards,
Jörg von Lingen - Interlocking scheme coordinator

Torben Brand wrote on 02/03/2017 16:20:
> Dear railML infrastructure forum,
> This posting contains the discussion to an extension towards
> the Signal
> For use case and naming conformity I refer to the topic
> "railML 2.3 infrastructure extension for capacity planning
> and network statement usecases".
> We need a simple route description for capacity modeling
> purposes. I see this as not yet interlocking, but a simple
> list of route attributes.
> The extended route attributes are placed in the <NO:route>
> element under the signal that forms its starting position.
> There can be multiple <NO:route> elements under one <signal>
> element. There is one route element per route starting from
> the signal.
> The new extended attributes are:
>
> route topology
>
> This to place the individual routes in the topology. One
> signal can have multiple routes starting from routeEntry,
> passing routeVia (or switchAndPosition) and ending at
> routeExit. One specific route can only start from one
> signal, and is placed under that signal in the railML
> structure.
>
> @name [datatype: string] for example "A-1-L"
> @routeEntry [datatype: ref to the <signal> starting the
> route]
> @routeExit, [all datatype: ref to any element ending the
> route usually a signal]
> @routeVia ([datatype: ref to any element that the route
> passes and uniquely describes the route from other routes
> starting from and ending at the same signal.]
> alternatively use @switchAndPosition. SwitchandPosition is
> used in the future interlocking schema to define route via
> topology. As this is somewhat complex and requires more data
> (all switch id's and their course along the route) we
> suggest using a simpler approach to just define any via
> element to distinguish the routes from each other.
> @setTime [datatype: time in seconds]
> Time for the route to be set by the CTC and interlocking.

> This is from the command is given by the dispatcher at the
> CTC/OCS terminal and to the signal light lights up or the
> movement authority is displayed in the MMI in the CAB.
>
> Speed profile(s) for the route
>
> @proceedSpeed [datatype: integer in km/h]
> Proceed speed is a speed restrictions by the route. This is
> valid for the whole route (from signal to signal). No
> value=track speed
> @releaseSpeed [datatype: integer in km/h]
> Release speed is the speed at which the brake curve
> intervention is removed and the train driver is unsupervised
> except from SPAD. This is valid for the whole route (from
> signal to signal).
> @approachSpeed [datatype: integer in km/h] used together
> with @approachPoint
> @approachPoint [ref to any object from where the speed must
> be lower than the appraochSpeed this is usually a signal]
> Used for a speed restriction on an approach zone in front of
> the route (Before RouteEntry) a train must obey if the route
> is closed (red light). The approachPoint refers to where the
> approach zone starts in front of the route. This is in
> Norway the distant signal in front of the route. The message
> is relayed to the train by a balise of type "fremskutt
> forsignal (FF)". The connection does not need to be
> modeled.
> ApproachSpeed can also be used for multiple route approach
> speed profiling. This is outside our use case as it is not
> implemented in Norway.
>
> Overlap in connection with the route
>
> overlap conditions by the route. The overlap condition needs
> to be placed on the route and not on the ending signal as
> the overlap condition can vary per what route is set.
> @overlapRef. The overlap always starts at the RouteExit
> signal. The reference is to the end of the overlap/slip.
> This is usually a train detection element.
> @overlapValidityTime [datatype: time in seconds]. Duration
> the overlap is active blocking potential overlapping routes
> from forming. The overlap is formed together with the route
> and is released after overlapValidityTime has run out after
> the overlapReleaseTimer value has been triggered.
> @speedInOverlap [in km/h; if the attribute is not set the
> default value is 0 km/h]. Not in our use case as speed in
> overlap is always 0 in Norway. This is a whish by Bob
> Jansen.

>
> The overlap release condition (overlapReleaseTimer) can be
> generic determined by the @system defined in the
> <controller>. To distinguish between a slip and an overlap,
> is achieved with overlapValidityTime. If the value is set
> it's an overlap. If the value is not set the overlap is a
> slip.
> The type of system will be mapped against attributes
> @overlapReleaseTimer and @overlapReleaseTimerHead. These
> will then not be needed in our use case. But I will include
> them here for completeness:
>
> @overlapReleaseTimer [datatype:ref]
> Reference to the trigger point for the overlap release
> timer. This is usually a train detection element.
> If no overlapReleaseTimer is set, but a overlapValidityTime
> is set, the default value is that the timing starts from
> when the train has stopped on the route.
> @overlapReleaseTimerHead [bolean]
> "yes"= valid for first axel (head) of the train
> "no"=valid for last axel (tail/rear) of the train
>
> Use cases - The overlapValidityTime is triggered in
> different ways:
> "occupyRoute"
> @overlapReleaseTimer=ref to routeEntry signal with
> @overlapReleaseTimerHead=yes
> For when the first axle of the train occupies the first TVD
> section of the route. For when the head of the train is at
> the routeEntry signal. Applicable for the following systems
> in Norway: SIMIS.
> "releasePreviousRoute"
> @overlapReleaseTimer=ref to routeEntry signal with
> @overlapReleaseTimerHead=no
> For when the first axle of the train occupies the first TVD
> section of the route and the last axle releases the TVD
> section in front of the route. For when the rear of the
> train is at the routeEntry signal. Applicable for the
> following systems in Norway:NSI-63.
> "stopOnRoute"
> @overlapReleaseTimer and @overlapReleaseTimerHead not set.
> For when the train has stopped on the route under ETCS L2 or
> on a station with a local dispatcher (that can verify that
> the train has stopped).
> "slip"
> @overlapReleaseTimer=ref to train detector element behind
> routeExit signal with @overlapReleaseTimerHead=no
> for when the last axle releases the TVD section of the

> slip/overlap behind the route (DE:"Durchrutschweg").
>