Subject: [railML2] adding an attribute for clearance on switches and crossings.
Posted by Torben Brand on Tue, 09 Mar 2021 13:12:17 GMT
View Forum Message <> Reply to Message

The Norwegian railway sector has the need to transfer information on the clearance of a switch or crossing. The clearance is the position where two train profiles can meet without just touching each other. UC is for signal planning.

Bane NOR and Jernbanedirektoratet currently use the extension attribute @nor:clearance of type xs:double on the <switch> and <crossing> elements.

The attribute is defined as "Distance in meters between the position of a switch/crossing (see @pos) and to the position of its clearance point behind the switch/crossing. The value is always positive."
PS. For a switch with @orientation="incoming" you need to calculate backwards for the position of the clearance.

Note that information on the formula used for the calculation of the clearance and the input values used, like choice of train profile, must be obtained from other external data sources.

Jernbanedirektoratet and Bane NOR are suggesting adding the extension into railML2.5 as

<switch>@clearance and <crossing>@clearance

Code example:
<switch absPos="200.0" code="KO-SPV-803680" id="sw41" name="1" clearance="50.0" normalPosition="straight" pos="0.0" trackContinueCourse="straight"/> <connection course="right" id="c41-43" maxSpeed="40.0" orientation="outgoing" ref="c43"/> </switch>

Subject: Re: [railML2] adding an attribute for clearance on switches and crossings.
Posted by christian.rahmig on Fri, 09 Apr 2021 14:20:49 GMT
View Forum Message <> Reply to Message

Dear Torben,

thank you for bringing up this topic into the forum discussion as the clearance point is important for various use cases dealing with railway operations.

Basically, I see two options for implementation:
1) explicit modelling of clearance point as own element, e.g. <clearancePoint>
2) modelling of distance between begin of switch and clearance point

Option 1 may be suitable for railML 3.x where we are refactoring the whole infrastructure model. For railML 2.x I prefer option 2 to stay as close as possible to the current implementation focusing only on small adaptations.

Therefore, your idea to introduce a new attribute @clearance (tLengthM) for <switch> and <crossing> sounds reasonable. Two things need to be considered:

a) It must be clarified that @pos always defines the same reference point of the <switch> and the <crossing>. For a <crossing> the reference point shall be its center point. The two clearance points can be derived by calculating @pos+@clearance and @pos-@clearance. For a <switch> the reference point can be the (topological) begin of the switch or the (virtual) center point. It is obvious that depending on the choice of the reference point, the value for @clearance will be different. Therefore, it is essential to fix the exact location of @pos relative to the switch. What are the best practices from the past?

b) Usually, the clearance point is marked by a small infrastructure element: the clearance post. This post marks the point, where the distance between two track center lines reaches 3.5 meters. Therefore, the clearance point is linked with the regular clearance gauge profile. If different clearance gauge profiles shall be considered (resulting in different clearance points), the attribute @clearance need to be repeatable and therefore transformed into a repeatable child element. Do you need to model different clearance points at once?

Any comments and ideas from the community are highly appreciated...

Best regards
Christian

Subject: Re: [railML2] adding an attribute for clearance on switches and crossings.
Posted by Jörg von Lingen   on Sun, 11 Apr 2021 04:25:22 GMT
View Forum Message <> Reply to Message

Dear all,

just to explain the situation/solution in railML3:

For movableElements like switches or crossings we have the reference by <hasGaugeClearanceMarker> to the physical marker near the track. However, for the interlocking purpose this info is not sufficient as one would need the explicit branch, which is possibly fouling the gauge of the other track. Therefore there is the reference <hasFoulingTrainDetector> in addition.

Once you have defined the clearance point it is fixed with the related clearance gauge. So if you would change using a different clearance gauge then you have to redefine the clearance point. Beside the physical marker this would also affect the interlocking.

Best regards,
Joerg v. Lingen - Interlocking Coordinator
Am 09.04.2021 um 16:20 schrieb Christian Rahmig:
> Option 1 may be suitable for railML 3.x where we are

> refactoring the whole infrastructure model. For railML 2.x I
> prefer option 2 to stay as close as possible to the current
> implementation focusing only on small adaptations.
>
> b) Usually, the clearance point is marked by a small
> infrastructure element: the clearance post. This post marks
> the point, where the distance between two track center lines
> reaches 3.5 meters. Therefore, the clearance point is linked
> with the regular clearance gauge profile. If different
> clearance gauge profiles shall be considered (resulting in
> different clearance points), the attribute @clearance need
> to be repeatable and therefore transformed into a repeatable
> child element. Do you need to model different clearance
> points at once?