

Hello,

I am working on a Thales railway infrastructure project for Norway. An important topic is the mapping of the infrastructure by using railML v3.1. When modelling the railway network (on the Micro Layer), we have a problem with the crossing element of railML. In my view the element 'crossing' (xsd:type rail3:Crossing) should be expanded in its definition. I miss the possibility of referencing to 'netRelation' (rail3:NetRelation) from this element. What I mean can be shown well using the example of the definition of a double slip in railML v3.1. In contrast to a crossing, this very similar infrastructure element supports referencing to 'netRelation'.

Simplified example of a double slip by railML v3.1

```
<switchIS id="Dhk3FD1ZZsZZfKdMmH8xO7V9" type="doubleSwitchCrossing" >
  <name name="KAM 2" language="en" />
  <spotLocation id="spotlocationId"
    netElementRef="Z3GkdhHfEBn3Xvh6bZZpx2V9" intrinsicCoord="0"/>
  <straightBranch netRelationRef="VyATIZHrOBtcgXQ7DBI5Cfa"/>
  <straightBranch netRelationRef="Z3q46YPEowm4bZHzS4G2Ysl8"/>
  <turningBranch netRelationRef="Z0C40IIZZSEq6eSXY9sakbU8"/>
  <turningBranch netRelationRef="iEkqg5X8LFb8Opwn0KNYmb"/>
</switchIS>
```

It would be very helpful for our project if two 'straightBranch' elements would be available, comparable to the 'switchIS' element.

Argumentation:

A crossing can be considered topologically as a simplified switch crossing.

From the safety perspective of interlockings, the branching, merging and crossing of tracks must be modelled in the form of a netRelation. Precisely these structures in the network are to be protected by interlocking measures. The 'natural' identification of such a structure occurs via the topology.

From my perspective, railML v3.1 offers two levels of abstraction for mapping the railway network. Each layer has its own root element/ type.

- a. a 'graph layer' represented by the 'topology' element (rail3:Topology)
- b. an element oriented 'infrastructure layer' represented by 'functionalInfrastructure' element (rail3:functionalInfrastructure)

The linking from the 'infrastructure' layer to the 'graph layer' must be supported when it comes to elementary properties, such as crossing of tracks.

Mit freundlichen Grüßen/Best regards

--

Heidrun Jost  
Data Manager  
Transportation Systems  
Thales Deutschland GmbH

Phone: +49 (0) 30 688306 423  
Schützenstr. 25 10117 Berlin Germany

---

---

Subject: Re: [railML3] Request for extension of the 'crossing' infrastructure element  
Posted by [christian.rahmig](#) on Fri, 29 May 2020 11:20:24 GMT

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

---

Dear Heidrun,

thank you for your feedback on railML 3.1 implementation. The issue you raised is very relevant.

Current implementation in railML 3.1 does not contain child elements for the various branches of a crossing. However, since <crossing> is (like any other functional infrastructure element) extendable, you are free to create such child elements for branches. From side of railML.org we may think about adding these missing child elements for release with railML 3.2, especially if further partners from the community share your opinion. Therefore, my question to the railML community: Do you have a need for modelling branches at crossings?

For the background discussion: there are different answers to the question whether a crossing (not a switch crossing!) can be considered as a topology relevant element. Some say "yes", because there is a (physical) connection of rails based on different NetElements and some say "no", because there is no "topological choice" at a crossing (you may only go one way and have no chance to choose a branch). Any comments on this (rather philosophical) discussion are highly appreciated, too.

Best regards  
Christian

---

---

Subject: Re: [railML3] Request for extension of the 'crossing' infrastructure element  
Posted by [Jörg von Lingen](#) on Sat, 30 May 2020 07:05:58 GMT

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

---

Hi,

just a remark on the issue:

An interlocking always needs to set a particular (virtual) position to a crossing in order to clearly define the path. This is needed even if the trackwork outside does not move at all, i.e. no physical switching of the

crossing. But that is why the counterpart in IL subschema is named <movableCrossing>.

Best regards,

Joerg v. Lingen - Interlocking Coordinator

Am 29.05.2020 um 13:20 schrieb Christian Rahmig:

- > For the background discussion: there are different answers
- > to the question whether a crossing (not a switch crossing!)
- > can be considered as a topology relevant element. Some say
- > "yes", because there is a (physical) connection of rails
- > based on different NetElements and some say "no", because
- > there is no "topological choice" at a crossing (you may only
- > go one way and have no chance to choose a branch). Any
- > comments on this (rather philosophical) discussion are
- > highly appreciated, too.

---

Subject: Re: [railML3] Request for extension of the 'crossing' infrastructure element  
Posted by [Dominik Looser](#) on Tue, 02 Jun 2020 11:43:45 GMT

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

Dear all,

We would also very much appreciate if there was a straightBranch subelement for crossings. The crossing-element would then have the same logic with straightBranches as a doubleSwitchCrossing modeled as a switchIS. As we are looking at crossings in a similar way than doubleSwitchCrossings, a similar implementation in railML 3.1 would be very useful.

Best regards,

Dominik Looser - traFIT solutions gmbh

---

Subject: Re: [railML3] Request for extension of the 'crossing' infrastructure element  
Posted by [Fabiana Diotallevi](#) on Fri, 26 Jun 2020 13:46:16 GMT

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

Dear all,

we have managed to handle the railML 3.1 import/export of crossings in our software Rail-AiD using the <external> tag of the <crossing> tag to store the reference to the netRelations.

For example, the railML export of crossing Sc01 would be:

```
<crossing id="scr1">
  <name name="Sc01" language="en"/>
  <spotLocation id="scr1_sloc01" netElementRef="ne6" applicationDirection="normal"
intrinsicCoord="1.0000"/>
```

```
<designator register="_Example" entry="CROSSING Sc01"/>
<external id="scr1_1" ref="nr_ne6ne9_scr1"/>
<external id="scr1_2" ref="nr_ne3ne10_scr1"/>
</crossing>
```

The <external> tags contain the references to the netRelations that correspond to the straight branches of the crossing.

Do you think that this approach is correct?

## File Attachments

---

1) [Sc01.PNG](#), downloaded 482 times

---

---

Subject: Re: [railML3] Request for extension of the 'crossing' infrastructure element  
Posted by [christian.rahmig](#) on Fri, 03 Jul 2020 11:04:34 GMT

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

---

Dear Fabiana,

Fabiana Diotallevi wrote on Fri, 26 June 2020 15:46

...

The <external> tags contain the references to the netRelations that correspond to the straight branches of the crossing.

Do you think that this approach is correct?

that's an interesting approach you present, which shows that the railML schema syntax provides a lot of flexibility. However, this flexibility needs to be limited by semantic constraints, and from my perspective, we should define such constraints for your model approach :-)

These are my arguments against your approach:

1. The element <external> was discussed as an element for file external linking, e.g. to be used when cutting the infrastructure into several pieces for different railML files (see forum discussion <https://www.railml.org/forum/index.php?t=msg&th=657&start=0> and Trac ticket #363)

2. The above mentioned discussion in the forum concluded with the opinion that this <external> element is not needed as its functionality can be realized with existing elements/parameters of id and designator. Therefore, element <external> shall be marked deprecated for future railML 3.x versions.

3. Linking of NetElements (and NetRelations) is the task of topology in order to form a consistent and navigable railway network based on a graph. Therefore, the proposed solution formulated in Trac ticket #380 aims at implementing the missing link between <crossing> and underlying topology without re-definition of topological relations on higher infrastructure levels.

What do you think, Fabiana, does the proposed solution in Trac ticket #380 fulfil your needs?

Best regards

Christian

---

---

Subject: Re: [railML3] Request for extension of the 'crossing' infrastructure element  
Posted by [Fabiana Diotallevi](#) on Mon, 06 Jul 2020 08:16:53 GMT

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

---

Dear Christian,  
the proposed solution (Trac ticket #380) is perfect for us.

As far as railML 3.1 import/export is concerned, if it is ok for you, we will keep using the  
<external> tag as it is the only way we found to store that information.

Kind regards,  
Fabiana

---