Subject: Export linear location information for switch tip/left/right branch and crossings
Posted by heidrun.jost@thalesgroup. on Fri, 22 Mar 2024 15:16:19 GMT
View Forum Message <> Reply to Message

Dear all,

we have a question for modelling location information at points and crossings.

For insertion of a TemporarySpeedRestriction starting at a point or crossing we need to export the point/crossing extensions:

[BILD]

In our model the tip-branch, left-branch and right-branch are not engineered as track elements and therefore they cannot be exported as rail3:track elements.

Our current solution looks like this:

```
  <switchIS id="Dhk3FD1ZZsZZfKdMmH8xO7V9">
     <name name="6" language="en"/>
     <spotLocation id="SL_Dhk3FD1ZZsZZfKdMmH8xO7V9"
netElementRef="Z3GkdhHfEBn3Xvh6bZZpx2V9" intrinsicCoord="0">
      <linearCoordinate positioningSystemRef="lpsID_23" measure="31410"/>
     </spotLocation>
     <leftBranch netRelationRef="Z3GkdhHfEBn3Xvh6bZZpx2V9_VyATIZHrOBtcgXQ7DBI5Cfa"/>
     <rightBranch
netRelationRef="Z3GkdhHfEBn3Xvh6bZZpx2V9_Z3q46YPEowm4bZHzS4G2Ysl8"/>
  </switchIS>
```

Our solution is to add linear location information for each point branch to the export. The implicit association between the linear location and the tip-, left and right branch can be determined by the branch specific net relations.

```
  <switchIS id="Z7YkbysVoqh1S5aZD44nOAz8">
     <!-- tip branch /-->
     <linearLocation id="LL_T_Z7YkbysVoqh1S5aZD44nOAz8">
      <associatedNetElement netElementRef="ypQhOfgM0r7fWvQ4mjsVL9"
intrinsicCoordBegin="1" intrinsicCoordEnd="0.9" keepsOrientation="true">
       <linearCoordinateBegin positioningSystemRef="lpsID_91" measure="168869"/>
       <linearCoordinateEnd positioningSystemRef="lpsID_91" measure="168874"/>
      </associatedNetElement>
     </linearLocation>

     <!-- left branch /-->
     <linearLocation id="LL_L_Z7YkbysVoqh1S5aZD44nOAz8">
      <associatedNetElement netElementRef="kDAg4skv4haCdKBBnFZZyWa"
intrinsicCoordBegin="0" intrinsicCoordEnd="0.1" keepsOrientation="true">
       <linearCoordinateBegin positioningSystemRef="lpsID_91" measure="168874"/>
       <linearCoordinateEnd positioningSystemRef="lpsID_91" measure="168939"/>
      </associatedNetElement>
     </linearLocation>

     <!-- right branch /-->
     <linearLocation id="LL_R_Z7YkbysVoqh1S5aZD44nOAz8">
      <associatedNetElement netElementRef="sQAouLaZHRDbYgyqEhEj4Xa"
intrinsicCoordBegin="1" intrinsicCoordEnd="0.9" keepsOrientation="true">
       <linearCoordinateBegin positioningSystemRef="lpsID_91" measure="168874"/>
       <linearCoordinateEnd positioningSystemRef="lpsID_91" measure="168914"/>
      </associatedNetElement>
     </linearLocation>

     <spotLocation id="SL_Z7YkbysVoqh1S5aZD44nOAz8"
netElementRef="ypQhOfgM0r7fWvQ4mjsVL9" intrinsicCoord="1">
      <linearCoordinate positioningSystemRef="lpsID_91" measure="168874"/>
     </spotLocation>
     <leftBranch netRelationRef="kDAg4skv4haCdKBBnFZZyWa_ypQhOfgM0r7fWvQ4mjsVL9"/>
     <rightBranch netRelationRef="sQAouLaZHRDbYgyqEhEj4Xa_ypQhOfgM0r7fWvQ4mjsVL9"/>
</switchIS>
```

Note: The determination which linear location belongs to which branch can be done be evaluating the rail3:switchIS branch net relations.

```
<leftBranch netRelationRef="kDAg4skv4haCdKBBnFZZyWa_ypQhOfgM0r7fWvQ4mjsVL9"/>
<rightBranch netRelationRef="sQAouLaZHRDbYgyqEhEj4Xa_ypQhOfgM0r7fWvQ4mjsVL9"/>
```

The tip branch can be found by comparing both netRelations and taking the netElement which is referenced in both.
The other netElement is the branch specific (left/right) net relation.

```
    <netRelation id="kDAg4skv4haCdKBBnFZZyWa_ypQhOfgM0r7fWvQ4mjsVL9"
positionOnA="0" positionOnB="1" navigability="Both">
     <!-- tip /-->
     <elementA ref="kDAg4skv4haCdKBBnFZZyWa"/>
     <!-- left branch /-->
     <elementB ref="ypQhOfgM0r7fWvQ4mjsVL9"/>
    </netRelation>
    <netRelation id="sQAouLaZHRDbYgyqEhEj4Xa_ypQhOfgM0r7fWvQ4mjsVL9"
positionOnA="1" positionOnB="1" navigability="Both">
     <!-- tip /-->
     <elementA ref="sQAouLaZHRDbYgyqEhEj4Xa"/>
     <!-- right branch /-->
     <elementB ref="ypQhOfgM0r7fWvQ4mjsVL9"/>
    </netRelation>
```

After finding the branch specific netElement the linearLocation can be found by comparing against linear locations associatedNetElement/netElementRef value.
right-branch linear location -> right branch km values:

```
 <linearLocation id="LL_R_Z7YkbysVoqh1S5aZD44nOAz8">
```

<associatedNetElement netElementRef="sQAouLaZHRDbYgyqEhEj4Xa" intrinsicCoordBegin="1" intrinsicCoordEnd="0.9" keepsOrientation="true">

First question: Is this solution correct or is there a better proposal?
Second question: We need the same for crossings. But this solution does not work for crossings.
A crossing consists of 4 netElements (upper-left, lower-left, upper-right and lower-right).
The netRelations for crossings (and switchable crossings) are provided via straight branch and turning branch netRelations.
It is not possible to find out based on these netRelations, which netElement belongs to upper-left, lower-left, upper-right and lower-right.

Best regards,
--
Heidrun Jost
Data Manager TMS Norway
Transportation Systems
Thales Deutschland GmbH

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

Subject: Re: Export linear location information for switch tip/left/right branch and crossings
Posted by Larissa Zhuchyi on Fri, 22 Mar 2024 16:37:32 GMT

View Forum Message <> Reply to Message

Regarding the first question: The addition of three linear locations I do not understand at fool extent, because the topology of a switch is already determined by left, right branches and application direction. If they are needed to model temporary speed restrictions maybe they should be children of <speedSection>. Then again if the netElementRef of the linear location of <speedSection> is the same as the net element of netRelationRef of right/<leftBranch> then it applies to the left, right or tip branch.

If you want to state that all the three topology net elements are covered by a railway switch and you have the exact coordinates to indicate to which extent then <areaLocation> seems to be a better solution.

Regarding the second question: please see the guide on crossings of the Wiki of railML3 [1] (section 2) for modelling of a crossing as a combination of switch/@type="doubleSwitchCrossing" and two elements switch/@type="switchCrossingPart" with @applicationDirection "normal" and "reverse". I think this approach was developed specifically for your problem to determine whether "netElement belongs to upper-left, lower-left, upper-right and lower-right".

Please let me know whether:

(1) linear locations can be children of <speedSection> in your model or there is still a need to

place them inside a switch;

(2) approach [1] meets your requirements for the crossings.

[1] https://wiki3.railml.org/wiki/Dev:Double_and_single_switch_c rossing