

---

Subject: Re: Double switch crossing: 'crossingRef' attribute for the fictive switches  
Posted by [Susanne Wunsch railML](#) on Wed, 10 Oct 2012 16:10:22 GMT

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

---

Christian Rahmig <coord@infrastructure.railml.org> writes:

>  
> in the discussion about the macroscopic infrastructure elements I set  
> up a first version regarding Dirk's good idea of generally allowing  
> for a grouping of (microscopic and macroscopic) infrastructure  
> elements.  
>  
> Here are the details of the concept, which can be also found in trac  
> ticket [1]:  
>  
> 1. The concept of macroscopic modelling of infrastructure elements is  
> not limited to switches and crossings. In particular, the following  
> elements might be of interest:  
>

- simple switch (de: Einfache Weiche)
- simple crossing (level junction, de: Einfache Kreuzung)
- > - simple switch crossing (de: Einfache Kreuzungsweiche)
- > - double switch crossing (de: Doppelte Kreuzungsweiche)
- > - three way switch
- > - turntable (de: Drehscheibe)
- transfer table (de: Schiebebühne)

I find multiple possibilities to group the basic railway elements (see above) into macroscopic objects.

- > - crossover (de: Gleisverbindung)
- > - double crossover (de: Doppelte Gleisverbindung)
  
- wye (triangular junction, de: Gleisdreieck)
- ??? (de: Gleisfünfeck)
- grand union (two double-track railway lines cross at grade)
- flying junction (grade separated crossing)
- double junction (double-track junction, de: zweigleisiger Abzweig)
- ??? (de: Ausweiche)
- ...

Do we really want to define this level of topology now?

- > 2. For an implementation of the macroscopic infrastructure element  
> feature in railML 2.2 the following solution is suggested:  
>  
> - In the type `tnfrastructure` a new container element  
> `<macroscopicInfrastructureElements>` is defined.

- > - This element contains a list of <macroscopicInfrastructureElement> objects.
- > - A macroscopic infrastructure element is defined by a list of references to other (microscopic and macroscopic) infrastructure elements.

+1 for all above mentioned

- > - The type of the macroscopic infrastructure element is specified in the parameter "elementType", which offers an (extendable) enumeration list of infrastructure elements, e.g. 'track', 'ordinarySwitch', 'threeWaySwitch', 'simpleCrossing', 'simpleSwitchCrossing', 'doubleSwitchCrossing' and 'turntable'.

Why are the values "insideCurvedSwitch" and "outsideCurvedSwitch" included? This geometric layout information should be at another layer, I mean.

Please add the "transferTable" to the enumeration list.

- > - The <macroscopicInfrastructureElement> inherits the parameters "id", "name" and "code" from the type tElementWithIDAndName.

+1

- > - The macroscopic infrastructure element contains several (at least one) <infrElementRef> reference objects.
- > - Each <infrElementRef> element provides the required parameters "elementType" for specifying the type of the referenced infrastructure element and "ref" for referencing the ID of the more detailed infrastructure element.

Please, do not abbreviate the element names.

Why not to allow all special element references and generic additions? That way, we could easily apply key-keyref constraints.

Do you really want the "sequence" attribute inside the \*Ref elements? I find it hard to define the sequence of the microscopic elements inside the macroscopic objects. The most important fact is, how are the microscopic elements connected with each other? How to ensure that in a consistent way?

```
<macroscopicInfrastructureElement id="mie1" code="sw12-14"
  elementType="other:crossover">
  <switchRef ref="sw12"/>
  <switchRef ref="sw14"/>
  <trackRef ref="tr1456"/>
```

```
</macroscopicInfrastructureElement>

<macroscopicInfrastructureElement id="mie2" code="tt1"
  elementType="turntable">
  <!-- The turntable tt1 consists of three tracks, that supposed to be
    defined using the default railML structure, each with two
    crossing elements to refer to, additionally the connection tracks
    are also listed, one "incoming", three "outgoing" -->
  <trackRef ref="tr1234"/>
  <trackRef ref="tr1235"/>
  <trackRef ref="tr1236"/>
  <crossingRef ref="cr1234-1235"/>
  <crossingRef ref="cr1234-1236"/>
  <crossingRef ref="cr1235-1236"/>
  <crossingRef ref="cr1235-1234"/>
  <crossingRef ref="cr1236-1234"/>
  <crossingRef ref="cr1236-1235"/>
  <genericRef ref="tb234" type="other:connection"/>
  <genericRef ref="tb235" type="other:connection"/>
  <genericRef ref="tb236" type="other:connection"/>
  <genericRef ref="te400" type="other:connection"/>
</macroscopicInfrastructureElement>
```

During hacking the above examples I feel that we should further think about the most common use cases and how to handle them with the new model.

Hope for any comments, remarks, questions, further ideas.

Kind regards...  
Susanne

--  
Susanne Wunsch  
Schema Coordinator: railML.common

---