Subject: Re: Double switch crossing: 'crossingRef' attribute for the fictive switches
Posted by Christian Rahmig on Thu, 18 Oct 2012 15:33:57 GMT
View Forum Message <> Reply to Message

Dear Susanne and other railML users,

> 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?

It is not necessary to define a complete list of all possible
macroscopic topology elements, but I think it helps if our new approach
allows for an easy adaptation in future.

>> - 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.

Yes, you are right. From the topology view, an "insideCurvedSwitch" is
identical to an "ordinarySwitch".

> Please add the "transferTable" to the enumeration list.

+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?

If we implement special element references, we need to define a "complete" list of topologic elements. You are right, that this approach will help us much better regarding the key-keyref constraints. But as soon as we include a "genericRef" element, we have to check the attribute "type" to determine the type of referenced object. Therefore, we should try to minimize such genericRef cases.

```
> <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>
```

Thank you for these good examples!

Regards

--
Christian Rahmig
railML.infrastructure coordinator