Subject: Double switch crossing: 'crossingRef' attribute for the fictive switches
Posted by pierre.simon on Wed, 04 Jul 2012 18:02:41 GMT

View Forum Message <> Reply to Message

With the current standard (railML 2.1), the way of modelling a double switch crossing implies to create fictives elements (4 switches and 2 tracks).
For post-processing purpose, we suggest to add a 'crossingRef' attribute to the fictives switches, so that after parsing it could be considered as one single element.

Could you add the attribute 'crossingRef' for the <switch> element in the next release ?

[de: Fuer die Modellierung der EKW/DKW wird ein Attribut 'crossingRef' benoetigt.]

--
----== posted via PHP Headliner ==----

Subject: Re: Double switch crossing: 'crossingRef' attribute for the fictive switches
Posted by Christian Rahmig on Thu, 05 Jul 2012 04:12:35 GMT

View Forum Message <> Reply to Message

Hello Pierre,

> With the current standard (railML 2.1), the way of modelling a double
> switch crossing implies to create fictives elements (4 switches and 2
> tracks).

the way of modelling a simple or double switch crossing using railML, which you describe, is not the only one. The other - more macroscopic view - specifies a <crossing> element with the type attribute "simpleSwitchCrossing" or "doubleSwitchCrossing". In that case, there is only one element and a referencing is not necessary.

However, your approach of defining several switches and a crossing for a simple/double switch crossing as a combined element is much closer to a real node-edge model.

> For post-processing purpose, we suggest to add a 'crossingRef' attribute
> to the fictives switches, so that after parsing it could be considered as
> one single element.

Until now, it is not possible to explicitly group elements like switches being parts of a simple/double switch crossing. The proposed attribute "crossingRef" may close this gap and if we define this attribute being optional, it is possible to implement it with railML version 2.2.

Before creating a trac ticket dealing with that issue, I would like to ask the other users of railML infrastructure schema for their opinion about the "crossingRef" attribute. Any comments appreciated....

Regards

--
Christian Rahmig
railML.infrastructure coordinator

Subject: Re: Double switch crossing: 'crossingRef' attribute for the fictive switches
Posted by on Thu, 05 Jul 2012 13:23:40 GMT
View Forum Message <> Reply to Message

Dear Christian and Pierre,

> Before creating a trac ticket dealing with that issue, I would like to
> ask the other users of railML infrastructure schema for their opinion
> about the "crossingRef" attribute. Any comments appreciated....

In general, there is no objection against linking of two switches in RailML which are already linked in practice because it is one diamond crossing.

But we should stay as general as possible, which means: We should take into account that this is not bound to diamond crossings but also to other kinds of points such as three-way points. More extreme, it may also be used to implement a turntable or such in RailML as a grouped set of virtual points.

So, I would omit the term "crossing" but name it more as a grouping of virtual points to one physical element.

Best regards,
Dirk.

Subject: Re: Double switch crossing: 'crossingRef' attribute for the fictive switches
Posted by Christian Rahmig on Sat, 08 Sep 2012 09:00:56 GMT
View Forum Message <> Reply to Message

Hello Dirk and anyone interested,

> In general, there is no objection against linking of two switches in

> RailML which are already linked in practice because it is one diamond
> crossing.
>
> But we should stay as general as possible, which means: We should take
> into account that this is not bound to diamond crossings but also to
> other kinds of points such as three-way points. More extreme, it may
> also be used to implement a turntable or such in RailML as a grouped set
> of virtual points.
>
> So, I would omit the term "crossing" but name it more as a grouping of
> virtual points to one physical element.

thank you for your interesting remark. I agree that it might be useful to group "microscopic" infrastructure elements to "macroscopic" ones and in fact the double switch crossing consisting of 4 switches, 1 crossing and 4 short tracks is only one example.

However, in case we want to extend this approach, we need to think about the possible implementation again: In my previous post I suggested an attribute "crossingRef", which allows to refer to a crossing element. If we generalize the attribute, e.g. "infrastructureElementRef", we may get the problem that the destination's type of the reference is not clear by only evaluating the reference value. In particular, the "microscopic" switch may belong to a diamond crossing or a turntable, which are based on different types.

But still I am a big fan of the idea of grouping infrastructure elements. Therefore I want to suggest an alternative approach, which defines macroscopic infrastructure elements such as diamond crossings or turntables and let them refer to microscopic elements. The example of the double switch crossing mentioned above might look like this (simplified syntax):

```
<doubleSwitchCrossing id="dkw01">
   <elementRef type="crossing" ref="c01">
   <elementRef type="switch" ref="s01">
   <elementRef type="switch" ref="s02">
   <elementRef type="switch" ref="s03">
   <elementRef type="switch" ref="s04">
   <elementRef type="track" ref="t01c01">
   <elementRef type="track" ref="t02c01">
   <elementRef type="track" ref="c01t03">
   <elementRef type="track" ref="c01t04">
</doubleSwitchCrossing>
```

Please feel free to comment on that suggestion, but keep in mind that it might not be compatible with a railML 2.2 coming up soon.

Regards

--
Christian Rahmig
railML.infrastructure coordinator

Subject: Re: Double switch crossing: 'crossingRef' attribute for the fictive switches
Posted by on Tue, 02 Oct 2012 17:21:54 GMT
View Forum Message <> Reply to Message

Dear Christian,

> But still I am a big fan of the idea of grouping infrastructure
> elements. Therefore I want to suggest an alternative approach, which
> defines macroscopic infrastructure elements such as diamond crossings or
> turntables and let them refer to microscopic elements.

In general, I totally agree with you.

In particular, I would prefer not to force it to very special (limited) macroscopic elements.

The theory is in my opinion:
1. There is a limited number of natural microscopic elements: tracks, points, may be crossings (but even not necessarily crossings - could be two tracks). We should be able to enumerate all allowed microscopic elements.
2. There is a much more greater possible number of macroscopic elements, and may be we do not even know all possible macroscopic elements.

That's why I would prefer to use your 'grouping' idea in a very much generic way:

- No pre-defined macroscopic element type 'doubleSwitchCrossing'/'diamondCrossing' or 'turntable' or such.
- Macroscopic elements can refer to other macroscopic elements - there can be a hierarchy just as we have allowed it with OCPs (which I think is very good generic).

Your example would then be:

```
<macroscopicTrackElement id="kr01" type="crossing" >
    <elementRef type="track" ref="s01s03">
    <elementRef type="track" ref="s02s04">
</macroscopicTrackElement>

<macroscopicTrackElement id="dkw01" type="diamondCrossing" >
```

```
   <elementRef type="macro" ref="kr01">  <-- one macroscopicTrackElement
to another
   <elementRef type="switch" ref="s01">
   <elementRef type="switch" ref="s02">
   <elementRef type="switch" ref="s03">
   <elementRef type="switch" ref="s04">
   <elementRef type="track" ref="t01c01">
   <elementRef type="track" ref="t02c01">
   <elementRef type="track" ref="c01t03">
   <elementRef type="track" ref="c01t04">
</macroscopicTrackElement>
```

The attribute <macroscopicTrackElement>."type" is the compromise: It is pre-defined, but it is an enumeration which can always and easily be extended (and which can allow non-predefined enumeration values).

With this generic principle of grouping infrastructure elements, I think we are very flexible, very general and therefore have much advantages compared with the current infrastructure model, so that it is worth the effort of change. I would welcome such a change in 3.0.

Best regards,
Dirk.

Subject: Re: Double switch crossing: 'crossingRef' attribute for the fictive switches
Posted by Christian Rahmig on Sun, 07 Oct 2012 14:28:14 GMT

View Forum Message <> Reply to Message

Dear Dirk and dear railML community,

>> But still I am a big fan of the idea of grouping infrastructure
>> elements. Therefore I want to suggest an alternative approach, which
>> defines macroscopic infrastructure elements such as diamond crossings
>> or turntables and let them refer to microscopic elements.
>
> In general, I totally agree with you.
>
> In particular, I would prefer not to force it to very special (limited)
> macroscopic elements.
>
> The theory is in my opinion:
> 1. There is a limited number of natural microscopic elements: tracks,
> points, may be crossings (but even not necessarily crossings - could be
> two tracks). We should be able to enumerate all allowed microscopic
> elements.
> 2. There is a much more greater possible number of macroscopic elements,
> and may be we do not even know all possible macroscopic elements.

>
> That's why I would prefer to use your 'grouping' idea in a very much
> generic way:
>
> - No pre-defined macroscopic element type
> 'doubleSwitchCrossing'/'diamondCrossing' or 'turntable' or such.
> - Macroscopic elements can refer to other macroscopic elements - there
> can be a hierarchy just as we have allowed it with OCPs (which I think
> is very good generic).

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 crossing (de: Einfache Kreuzungsweiche)
- double switch crossing (de: Doppelte Kreuzungsweiche)
- three way switch
- crossover (de: Gleisverbindung)
- double crossover (de: Doppelte Gleisverbindung)
- turntable (de: Drehscheibe)

2. For an implementation of the macroscopic infrastructure element feature in railML 2.2 the following solution is suggested:

- In the type tInfrastructure 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.
- 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'.
- The <macroscopicInfrastructureElement> inherits the parameters "id", "name" and "code" from the type tElementWithIDAndName.
- 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.
   - Since macroscopic infrastructure elements may include not only
microscopic, but also other macroscopic infrastructure elements, the
attribute "elementType" provides the same enumeration list for the
infrastructure element's type as described above for the element type of
the <macroscopicInfrastructureElement> object.

[1] https://trac.assembla.com/railML/ticket/168

Regards

--
Christian Rahmig
railML.infrastructure coordinator

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 tInfrastructure 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

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

Subject: Re: Double switch crossing: 'crossingRef' attribute for the fictive switches
Posted by Christian Rahmig on Mon, 29 Oct 2012 16:38:40 GMT
View Forum Message <> Reply to Message

Dear railML users,

> 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 crossing (de: Einfache Kreuzungsweiche)
>   - double switch crossing (de: Doppelte Kreuzungsweiche)
>   - three way switch
> [...]

the term "three way switch" should replaced by "tandem turnout". Thanks
to Carsten for pointing out that mistake.

Regards

--
Christian Rahmig
railML.infrastructure coordinator

Subject: Re: Double switch crossing: 'crossingRef' attribute for the fictive switches
Posted by Christian Rahmig on Sat, 24 Nov 2012 12:33:12 GMT
View Forum Message <> Reply to Message

Dear railML users,

> 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.
> [...]

since the concept of macroscopic infrastructure elements includes many unanswered questions and apparently a lot of further detail work, we (Susanne and me) agreed on moving this proposed enhancement into the future, e.g. railML 3.0. Therefore, macroscopic infrastructure elements as described in Trac ticket [1] won't be available in railML 2.2.

[1] https://trac.assembla.com/railML/ticket/168

Regards

--
Christian Rahmig
railML.infrastructure coordinator

Subject: Re: Double switch crossing: 'crossingRef' attribute for the fictive switches
Posted by Susanne Wunsch railML on Mon, 26 Nov 2012 11:01:15 GMT

View Forum Message <> Reply to Message

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

> since the concept of macroscopic infrastructure elements includes many
> unanswered questions and apparently a lot of further detail work, we
> (Susanne and me) agreed on moving this proposed enhancement into the
> future, e.g. railML 3.0. Therefore, macroscopic infrastructure
> elements as described in Trac ticket [1] won't be available in railML
> 2.2.
>
> [1] https://trac.assembla.com/railML/ticket/168

Thanks for reverting this implementation in order to reduce the number of un-harmonized aspects for the future. ;-)

How about implementing the minimum variant as proposed by Pierre at the beginning?

New attribute 'crossingRef' into the 'switch' element.

Additionally 'anyAttribute's for 'switch' and 'crossing' elements, as there aren't any at the moment.

Kind regards...
Susanne

--
Susanne Wunsch
Schema Coordinator: railML.common

Subject: Re: Double switch crossing: 'crossingRef' attribute for the fictive switches
Posted by Christian Rahmig on Sun, 02 Dec 2012 12:51:09 GMT
View Forum Message <> Reply to Message

Dear Susanne and other railML users,

> How about implementing the minimum variant as proposed by Pierre at the
> beginning?
>
> New attribute 'crossingRef' into the 'switch' element.
>
> Additionally 'anyAttribute's for 'switch' and 'crossing' elements, as
> there aren't any at the moment.

I agree with your suggestion to implement any-attributes for switches and crossings. Regarding the parameter "crossingRef" I do not want to open the discussion again ;) Let's move it to 3.0. Thank you.

Regards

--
Christian Rahmig
railML.infrastructure coordinator

Subject: Re: Double switch crossing: 'crossingRef' attribute for the fictive switches
Posted by christian.rahmig on Mon, 28 Jan 2019 12:56:28 GMT
View Forum Message <> Reply to Message

Dear all,

Am 24.11.2012 um 13:33 schrieb Christian Rahmig:
> [...]
> since the concept of macroscopic infrastructure elements includes many
> unanswered questions and apparently a lot of further detail work, we
> (Susanne and me) agreed on moving this proposed enhancement into the
> future, e.g. railML 3.0. Therefore, macroscopic infrastructure elements
> as described in Trac ticket [1] won't be available in railML 2.2. [...]

Upcoming railML 3.1 does not consider macroscopic infrastructure elements on a generic basis. Instead, every infrastructure component uses its own dedicated data type. However, few aspects of macroscopic infrastructure elements have been implemented:

* Location
Every infrastructure element can be placed very flexible within the topology network: Spot locations, linear locations and area locations may even exist in parallel and define different spatial aspects on different modeling levels of detail.

* Hierarchy
Elements of same type can be put in a hierarchical order. In particular, an infrastructure element may reference its one and only parent infrastructure element using the attribute @belongsToParent. Such a reference attribute is available for example for platforms, operational points, signals, level crossings and lines.

The complex infrastructure elements adressed in Trac ticket [1] will be analysed in detail with the "Advanced Example" together with a future railML 3.x version.

Of course, any comments on this topic are highly appreciated...

[1] https://trac.railml.org/ticket/168

Best regards
Christian

--
Christian Rahmig - Infrastructure scheme coordinator
railML.org (Registry of Associations: VR 5750)
Phone Coordinator: +49 173 2714509; railML.org: +49 351 47582911
Altplauen 19h; 01187 Dresden; Germany   www.railml.org