
Subject: Switch: usage of attribute @course

Posted by [christian.rahmig](#) on Wed, 05 Apr 2017 10:51:33 GMT

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

Dear all,

a standard question for railML newcomers is about the connection of tracks via switches and crossings in order to form a railway network. Some years ago, we created a Wiki page [1] for this topic. It became one of the most called railML wiki pages. However, some questions remained and I would like to bring the discussion here to the forum in order to find a final solution for upcoming version 2.4.

The situation:

A switch is situated in the beginning or the end of a track and may be connected to other tracks. See the following example:

```
<track id="tr01">
  <trackTopology>
    <trackBegin id="tr01_tb" pos="0">
      <connection id="tr01_c01" ref="tr02_c01"/>
    </trackBegin>
    ...
    <switch id="sw01" pos="0" type="ordinarySwitch">
      <connection id="sw01_c01" ref="tr03_c01" orientation="incoming"
course="left"/>
    </switch>
  </trackTopology>
</track>
```

The switch begin is located in the beginning of track "tr01". The main course of the switch is defined by the <connection> in line 4. The branching course of the switch is defined by the <connection> in line 8.

The problem:

The attribute @course may have the values "left", "right" and "straight". However, the choice of this value currently depends on the orientation of the track where the switch is located. The wiki page [1] shows this in four small figures (examples 1-4). Consequently, the same type of switch (with respect to its construction layout) may define its branch one time with course="left" and the other time with course="right" depending on the different orientation of the track where the switch is located.

The question:

I want to ask you if you understand the current implementation / understanding of railML track connection modelling or whether you support to change it in the future? Shall the choice of value for

@course depend on the orientation of the track or shall it be independent and just linked with the construction layout of the switch?

I am looking forward to receiving your comments. The main aspects of the discussion and the final solution will be tracked with railML Trac ticket #39 [2].

[1] http://wiki.railml.org/index.php?title=Connection_between_tracks

[2] <http://trac.railml.org/ticket/39>

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

Subject: Re: Switch: usage of attribute @course
Posted by [christian.rahmig](#) on Tue, 13 Feb 2018 10:46:35 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear all,

please let me remind you on the following forum post about the use of <switch> attributes @course and @orientation. Please also consider the forum post by Claus Feyling on this matter (see [3]).

If there is no feedback from your side, we anticipate that for upcoming railML 2.4 you are happy with the current solution described in [4]. In any case, railML 3.x is going to implement a modified approach.

[3] <https://www.railml.org/forum/index.php?t=msg&th=516&start=0&>

[4] http://wiki.railml.org/index.php?title=Dev:Connection_between_tracks

Best regards
Christian

Am 05.04.2017 um 12:51 schrieb Christian Rahmig:

> Dear all,

>

> a standard question for railML newcomers is about the connection of
> tracks via switches and crossings in order to form a railway network.
> Some years ago, we created a Wiki page [1] for this topic. It became one
> of the most called railML wiki pages. However, some questions remained
> and I would like to bring the discussion here to the forum in order to

> find a final solution for upcoming version 2.4.

>

> The situation:

> A switch is situated in the beginning or the end of a track and may be connected to other tracks. See the following example:

>

```
> <track id="tr01">
>   <trackTopology>
>     <trackBegin id="tr01_tb" pos="0">
>       <connection id="tr01_c01" ref="tr02_c01"/>
>     </trackBegin>
>     ...
>     <switch id="sw01" pos="0" type="ordinarySwitch">
>       <connection id="sw01_c01" ref="tr03_c01" orientation="incoming"
> course="left"/>
>     </switch>
>   </trackTopology>
> </track>
```

>

> The switch begin is located in the beginning of track "tr01". The main course of the switch is defined by the <connection> in line 4. The branching course of the switch is defined by the <connection> in line 8.

>

> The problem:

> The attribute @course may have the values "left", "right" and "straight". However, the choice of this value currently depends on the orientation of the track where the switch is located. The wiki page [1] shows this in four small figures (examples 1-4). Consequently, the same type of switch (with respect to its construction layout) may define its branch one time with course="left" and the other time with course="right" depending on the different orientation of the track where the switch is located.

>

> The question:

> I want to ask you if you understand the current implementation / understanding of railML track connection modelling or whether you support to change it in the future? Shall the choice of value for @course depend on the orientation of the track or shall it be independent and just linked with the construction layout of the switch?

>

> I am looking forward to receiving your comments. The main aspects of the discussion and the final solution will be tracked with railML Trac ticket #39 [2].

>

> [1] http://wiki.railml.org/index.php?title=Connection_between_tracks

> [2] <http://trac.railml.org/ticket/39>

>

> 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

Subject: Re: Switch: usage of attribute @course
Posted by [Thomas Nygreen JBD](#) on Thu, 05 Apr 2018 15:38:20 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear all,

christian.rahmig wrote on Wed, 05 April 2017 12:51 The question:
I want to ask you if you understand the current implementation /
understanding of railML track connection modelling or whether you
support to change it in the future? Shall the choice of value for
@course depend on the orientation of the track or shall it be
independent and just linked with the construction layout of the switch?

I find the current implementation, described in [1] to be good, and I strongly suggest keeping it.

On the other hand, I find the wiki page for the switch/connection element to be lacking, with no
description of the reference frame for what is left and right. I suggest adding something like "as
seen in the up direction", as well as a link to [1]. It would also be interesting to know how many
software implementations follow the current interpretation and how many that don't. Notably, the
current "railML Simple Example by railML.org" [2] does not.

christian.rahmig wrote on Wed, 05 April 2017 12:51 The problem:
The attribute @course may have the values "left", "right" and
"straight". However, the choice of this value currently depends on the
orientation of the track where the switch is located. The wiki page [1]
shows this in four small figures (examples 1-4). Consequently, the same
type of switch (with respect to its construction layout) may define its
branch one time with course="left" and the other time with
course="right" depending on the different orientation of the track where
the switch is located.

It is a bit odd to argue that it is a disadvantage that the @course changes when the orientation of
the track does, as that is the case for important attributes of most elements, most notably @dir. As
the "up" direction is the main reference for everything else, it seems most reasonable to use the
same reference for @course. Changing "left" to "right" for @course is no more difficult than it is for
platformEdge/@side. It is also an advantage to have connection/@course = "right" and
platformEdge/@side = "right" actually being the same side of the track.

Regarding the fact that the same geometry can result in different topological descriptions: that is the case even if we flip the reference frame for @orientation="incoming".

christian.rahmig wrote on Wed, 05 April 2017 12:51 Please also consider the forum post by Claus Feyling on this matter (see [3]).

I have read Claus Feyling's post, and, as I have argued above, I disagree with the proposal to change the interpretation of @course. Also, I am not convinced about the need for @geometry, as it in most cases follows from the topology, and in remaining cases, such as the one he gives, can be described by switch/@trackContinueRadius and connection/@radius. Although, as the attachments are missing, there could be something I am missing.

[1] https://wiki.railml.org/index.php?title=Dev:Connection_between_tracks

[2] <https://www.railml.org/en/user/exampledata.html>

[3] <https://www.railml.org/forum/index.php?t=msg&th=516>

Best regards

Thomas

Subject: Re: Switch: usage of attribute @course

Posted by [Thomas Nygreen JBD](#) on Mon, 09 Apr 2018 12:57:16 GMT

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

Thomas Nygreen wrote on Thu, 05 April 2018 17:38 I have read Claus Feyling's post, and, as I have argued above, I disagree with the proposal to change the interpretation of @course. Also, I am not convinced about the need for @geometry, as it in most cases follows from the topology, and in remaining cases, such as the one he gives, can be described by switch/@trackContinueRadius and connection/@radius. Although, as the attachments are missing, there could be something I am missing.

I have now read the Q&A attachment (thanks to Christian Rahmig for sending it to me), and given some more thought to the suggestion of a new geometry attribute. As I will show below, @geometry is not strictly necessary. However, I now find @geometry to be a good addition, since it does not depend on using topology to deduce geometry, but separates the two. It also allows a more strict interpretation of the enumerated values for @course.

Before I continue, I would like to state that, regarding @course, I still prefer the current interpretation of "right" and "left" as seen in the "up" direction. This question is, however, completely separable from the other suggestions by Claus Feyling.

First, regarding stricter use, I have found no documentation of the meaning of the values "right", "left" and "straight" for @trackContinueCourse. Claus Feyling only uses "straight" for three-way switches, while I have always assumed that it can also be used for two-way switches. If all three values can be used also for two-way switches, the following combinations are all topologically

equivalent, as they all have a branch further left than the continuing track:

```
@trackContinueCourse = "straight" and connection[1]/@course = "left" @trackContinueCourse =  
"right" and connection[1]/@course = "left" @trackContinueCourse = "right" and  
connection[1]/@course = "straight"
```

All of these are valid under the railML XSD, but they might not all be desirable. They can be used to reflect the physical geometry of the switch, but again, that might not be desirable. Rather, I prefer to separate the geometry from the topology, and include the suggested @geometry. This also allows reducing the three possible alternatives above to just one.

Furthermore, I suggest:

making connection/@course mandatory, as it is not possible to determine topology without it deprecating @trackContinueCourse, as it is given by the remaining direction not used by any connection

Best regards
Thomas
