
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
