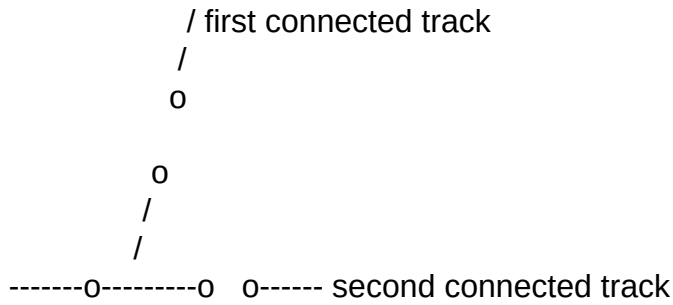
Subject: Re: V1.00 RC1: switchRef/crossingRef
Posted by Volker Knollmann on Wed, 29 Sep 2004 09:54:52 GMT
View Forum Message <> Reply to Message

On 28.09.2004 15:01, Matthias Hengartner wrote:
> In Berlin, we forgot to discuss about the implementation of
> switches/crossings which are placed on a <trackBegin>/<trackEnd>. In the
> last thread of this newsgroup, I suggested the following:
> "I'd prefer to have only a reference to a switch/crossing which is located
> in the <connections>-container."

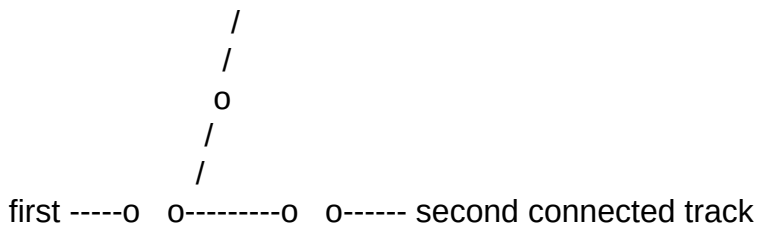
Jepp, that's the way I would prefer it, too. But as usual, things are
not as easy as in your example, although it was perfect to understand
your intention. So I will do my very best to make things complicated ;-)

If we start or end a track with a switch, we can distinguish between 2
cases:

(1) the switch element belongs to the straight track



(2) the switch element belongs to the branch track



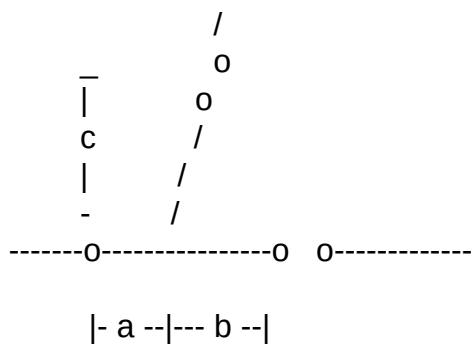
The crucial thing is the required "orientation"-attribute in the

<connection>-element of a switch. "orientation" can be either "incoming", "outgoing", "right angled" (???) or "unknown". Which value is to be chosen for the second track in case (1) and for both tracks in case (2)?

I suggest an additional value "straight" (which perfectly coincides with the possible values for "trackContinueCourse") and the __convention__ to let the <switch>-element be part of track at the switch's tip. Thus, role of every track is unambiguous.

The second point I want to make about switches is the "length"-attribute. If a switch is connected at the start or end of a track and the switch's "length"-attribute is set, where does the track effictively end or start? At the connection point of the switch or at the connection point + "length"?

And which length does "length" describe? I would interprete it as follows:



(a) is the "length"-attribute of <switch>

(b) is the "branchDist"-attribute of the <connection>-element to the straight track

(c) dito for the branching track

But this still doesn't solve the problem where the <track> ends. From my point of view this is important to generate proper information for the vacancy detection of elements.

And by the way: "branchDist" is to be defined in [m], not in [km]. Should this be changed to [km] to harmonize all distances and positions in the file?

So much for the moment... maybe I'll terrorize you with more posts as I work my way through 1.0RC1 and try to adapt our internal file format specs...

Thanks in advance for your comments!

Best regards from Braunschweig,
Volker Knollmann
