

Hello...

>> But this is of course a very "dirty" implementation. And I don't think that
>> the possibility to implement case (2) is really needed (It can easily be
>> avoided).
>
> Full ack. But the thing is to take EVERY case into account that CAN be
> realized in railML and therefore must be handled by the interpreting
> software.
> Of course there's no need to implement switches in such a queer way like
> case (2). But sooner or later, someone will fail to withstand the
> temptations of the dark side of the force^W^W^W eeeeeer railML... ;-)

hmm. Since the railML schema does only define the syntax of a railML file (and not the semantics/rail logics/consistency/...), there have always lots of restrictions and conventions to be made in the documentation/specification. So we could also "prohibit" such "queer ways" of implementing switches.

>> Or, final idea (for the moment ;-)) : We could combine these 2 approaches:
>> We could have a <simpleConnection> with a reference to a
>> <switch>/<crossing>.
>
> Hmm? That would be the above mentioned way: define a switch at
> pos="0.000" or pos="[length]" and refer to it from the branching
> element. The straight element is implicitly given by the parent of
> <switch>. This avoids some extra attributes and the handling of special
> cases....

No, this was not my idea (I have to admit that the description of my idea was not very comprehensible...):

Given a switch on the beginning or end of a track (track1)

```
o track2
/
/
/
```

o
o-----o track1

then we have a switch at pos="0.000" / pos="[length]" with a connection to the <trackBegin>/<trackEnd> - <simpleConnection> - <connection> of track2. So far nothing new.

Of course, we also have <trackBegin>/<trackEnd> - <simpleConnection> (***) on track1, with a connection to the previous/next <track> (which would be on the left in our ASCII-drawing).

Still nothing new.

The only additional thing would be a new attribute of the <simpleConnection>(***), something like "switchIDRef", which would refer to the switch (which is on the same track).

Is this a bit more comprehensible?

Best regards from Zurich in dark clouds
Matthias
