

Dear all,

in-between I had a discussion about this topic with some users and want to add the outcome here for your info.

The attached pictures show the 3 steps of evolution from simple switches to a double slip crossing if you go into interlocking domain.

step01: For a simple switch one needs to do a geometrical check in order to find out what's right and left of the deviating branches,

step02: This is more an intermediate state for illustration. The two switches are a bit superimposed (not yet a real double slip but to show the evolution).

Here the determination right/left shall be the same as in step01.

step03: The third step is the final superimposition but if you think of two simple switches making the picture then the decision for right/left shall be under the same rule.

The picture "switches01" shows the net plan for an example of double slip switch (SLIP SWITCH Dsw02):

- 1) select from straightBranch one with starting netElement
nr_ne5ne12_dsw7 -> ne5
- 2) connection straight -> ne12
- 3) connection turning -> ne6
- 4) geometrical check: ne6 is right of ne12
rightBranch=ne6, leftBranch=ne12
- 5) select the other end of the straightBranch -> ne12
- 6) connection straight -> ne5
leftBranch=ne5 (due to symmetry)
- 7) connection turning -> ne2
rightBranch=ne2

similar procedure for SLIP SWITCH Dsw04:

- 1) nr_ne9ne11_dsw10 -> ne9
- 2) straight -> ne11
- 3) turning -> ne6
- 4) geometrical check: leftBranch=ne6, rightBranch=ne11
- 5) ne11
- 6) straight -> ne9, rightBranch=ne11
- 7) turning -> ne10, leftBranch=ne10

Best regards,
Joerg v. Lingen - Interlocking Coordinator

Am 03.04.2020 um 06:05 schrieb Joerg von Lingen:

> Dear all,

>

> there seems to be a general issue when transforming a track plan into railML:

>

> 1) For an 'ordinarySwitch' we have in IS the elements 'leftBranch' and 'rightBranch'. Just from the netRelations it

> seems not really possible to decide which is one of the both branches. How would you solve the issue?

>

> 2) For a 'doubleSwitchCrossing' we have in IS the elements 'straightBranch' and 'turningBranch' but in IL we need to

> split into two normal switches which again have 'leftBranch' and 'rightBranch'. Could this be solved just from the

> topology information? How would you do this trick?

>

File Attachments

1) [step01.jpg](#), downloaded 316 times

2) [step02.jpg](#), downloaded 303 times

3) [step03.jpg](#), downloaded 303 times

4) [switches01.png](#), downloaded 316 times
