

---

Subject: Link a "doubleSwitchCrossing" to two switchesIL  
Posted by [Martin Zien](#) on Mon, 06 Dec 2021 18:55:30 GMT

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

---

My colleagues and me are facing a scenario, where a so called "doubleSwitchCrossing" is in a later step being linked two single switches inside the "interlocking" (in railML as well as inside the real-live-system).

The difficulty is, that certain individual "designator"-Elements and also the name would also have to be assigned in Infrastructure-node individually.

In the work group for the ETCS-Use case, we elaborated that this special scenario is not covered yet by the current possibilities of modeling inside railML 3.x.

To comply as much as possible with the already defined approaches in railML 3.2, we think it might help to introduce further optional sub-elements for the case of a "doubleSwitchCrossing" under such conditions.

These would appear only if needed - similar to what is being done with the specific elements "straightBranch" and "turningBranch".

As working title these specific extra branches should be called "switchPartition". Any referencing - e.g. inside "switchesIL" would occur to these sub-nodes.

In this regards it should be discussed, if this approach should also be considered for a singleSwitchCrossing as well - but only with one occurrence.

You can find a complete double switch crossing in the attachment.

```
<switchPartition id="swip_13cd" applicationDirection="normal">
  <name name="11" language="NO"/>
  <designator register="infrastructureRegister" entry="switch11entry"/>
  <leftBranch netRelationRef="nr_10388E_0_103C17_1"/>
  <rightBranch netRelationRef="nr_103C17_1_B543_0"/>
</switchPartition>
```

### File Attachments

1) [2021-12-06\\_DoubleSwitchCrossing\\_vs\\_2switchesIL\\_Forum.xml](#),  
downloaded 59 times

---

---

Subject: Re: Link a "doubleSwitchCrossing" to two switchesIL  
Posted by [Joerg von Lingen](#) on Tue, 07 Dec 2021 04:16:10 GMT

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

---

Dear Martin,

thanks for your proposal. I have two thoughts for it:

1) When referencing from switchIL it might be confusing if the target can be a switchIS or a switchPartition.

2) The speciality of a doubleSwitchCrossing over a singleSwitchCrossing is not clear. In the interlocking domain they will be modelled both as two linked switches.

P.S.: For everyone, who has no idea of what is already in railML3.x, please, have a look at <https://wiki3.railml.org/wiki/IL:switchIL>.

Dr.-Ing. Jörg von Lingen - Rollingstock scheme coordinator  
railML.org (Registry of Associations: VR 5750)  
Phone Coordinator: +49 351 87759 40; railML.org: +49 351 47582911  
Altplauen 19h; 01187 Dresden; Germany [www.railml.org](http://www.railml.org)

---

---

Subject: Re: Link a "doubleSwitchCrossing" to two switchesIL

Posted by [Martin Zien](#) on Fri, 14 Jan 2022 16:21:38 GMT

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

---

After a discussion with other members in December 2022, I would like to propose a new approach:

It works with two new switchIS-Types ("doubleSwitchCrossingPartitioned" - with one occurrence, "partitionSwitch" - with two occurrences) and the application of a parent-child-relation between both types.

Please refer to the attached example for details.

### File Attachments

1) [2022-01-11\\_Alignment\\_On\\_DoubleSwitchCrossing.xml](#) ,  
downloaded 46 times

---

---

Subject: Re: Link a "doubleSwitchCrossing" to two switchesIL

Posted by [christian.rahmig](#) on Mon, 17 Jan 2022 10:53:49 GMT

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

---

Dear Martin,

thank you very much for your new proposal. From infrastructure point of view, it looks good, as it does not introduce new elements, but only new types for the <switch> element. The new types "doubleSwitchCrossingPartitioned" and "partitionSwitch" can be used to define semantic rules on the appearance and multiplicity of <switch> child elements, e.g. branches.

In order to have a synchronized approach: is it also necessary to partition simple switch crossings? If yes: can we use your approach also for simple switch crossings (partitioned)? If no:

is the approach described in [1] with only one <switch> element for a simple switch crossing sufficient for all your needs?

[1] <https://cloud.railml.org/index.php/f/77969>

Thank you very much and best regards  
Christian

---

---

Subject: Re: Link a "doubleSwitchCrossing" to two switchesIL  
Posted by [christian.rahmig](#) on Mon, 24 Jan 2022 16:03:30 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Dear all,

following the latest discussions of the topic here in the forum and in the ETCS use case working group, I concluded the topic in a railML development issue [1].

[1] <https://development.railml.org/railml/version3/-/issues/484>

Best regards  
Christian

---

---

Subject: Re: Link a "doubleSwitchCrossing" to two switchesIL  
Posted by [Martin Zien](#) on Thu, 03 Feb 2022 14:07:07 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Dear all,

I added a new example with additional information on "spotLocation" on the single switches.

In case, the partitionSwitch-Elements are assumed at the same location ("inherited"), I would NOT include the optional details like the entries for x-y-z-Corodimates or "measure"-Value.

In case the "spotLocation" of the children is different from the parent - e.g. due to the rules of the Infrastructure Manager - the details could of course be applied accordingly.

Best regards  
Martin

### File Attachments

1) [2022-02-03\\_Alignment\\_On\\_DoubleSwitchCrossing.xml](#) ,  
downloaded 34 times

---

Subject: Re: Link a "doubleSwitchCrossing" to two switchesIL  
Posted by [Martin Zien](#) on Thu, 03 Feb 2022 14:10:29 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

To facilitate the "orientation" in regards to netElements inside the example, I would also add a sketch of the situation that is described inside the railML-snippet.

#### File Attachments

---

1) [2021-12-01\\_Sketch\\_doubleSlipCrossing.JPG](#), downloaded 46 times

---

---

Subject: Re: Link a "doubleSwitchCrossing" to two switchesIL  
Posted by [Martin Zien](#) on Thu, 10 Feb 2022 13:29:41 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

In ETCS-Workgroup meeting On 2022-02-04, we decided to extend this special approach of additional /switchIS@type "doubleSwitchCrossingPartitioned" also to "singleSwitchCrossingPartitioned".

So, I extended the example file. At the top of the file, there is "singleSwitchCrossingPartitioned" described. Below you can find the already known "doubleSwitchCrossingPartitioned"

I am looking forward to your responses.

#### File Attachments

---

1)  
[2022-02-10\\_example\\_singleSwitchCrossing\\_and\\_DoubleSwitchCrossing.xml](#), downloaded 37 times

---

---

Subject: Re: Link a "doubleSwitchCrossing" to two switchesIL  
Posted by [Martin Zien](#) on Mon, 28 Feb 2022 10:03:15 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Another correction after the most recent ETCS-Workgroup meeting On 2022-02-18:  
- a "singleSwitchCrossing" would never have an 'applicationDirection="both" ', I corrected it accordingly.

#### File Attachments

---

1)  
[2022-02-21\\_example\\_singleSwitchCrossing\\_and\\_DoubleSwitchCrossing.xml](#), downloaded 27 times

---

Subject: Re: Link a "doubleSwitchCrossing" to two switchesIL  
Posted by [Thomas Nygreen](#) on Tue, 01 Mar 2022 08:53:03 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Dear all,

Thank you, Martin, for these proposals and examples. I suggest to use the existing type="doubleSwitchCrossing" and type="singleSwitchCrossing", also when the individual halves are included. The belongsToParent attribute is a general feature in the infrastructure schema, and I believe that we should not start splitting up types to signal that this feature is used. A simple Xpath query (from the context of the first switch in the given example: `../switchIS[@belongsToParent = "swi_10"]`) will provide the "child" switches, if there are any.

I also hope that we can identify a more intuitive term than "partitionSwitch". When used in compound nouns, the word partition is normally placed last (e.g. hard disk partition), but this use is also rare. Some alternative suggestions: "switchPart", "switchCrossingPart", "partialSwitch", "componentSwitch", "virtualSwitch".

Regarding "virtualSwitch", the point is that these two halves are a virtual model of the physical switches placed in the opposite direction at each "corner" (i.e. at a, b, c and d for double switch crossings, and at either a and b or c and d for single switch crossings). Should our modelling reflect that, and also include the more physical (and nanoscopic) perspective as an option?

In the case of the single switch crossing, I also assume that the two <switchIL>s should be linked using <hasPositionRestriction> as described in [1]. It is not completely clear to me what the missing <branchTip> should point to in these cases. Perhaps that topic is worth a new thread in the IL part of the forum?

[1] [https://wiki3.railml.org/wiki/IL:switchIL#Single\\_Slip\\_Switch](https://wiki3.railml.org/wiki/IL:switchIL#Single_Slip_Switch)

---

Subject: Re: Link a "doubleSwitchCrossing" to two switchesIL  
Posted by [christian.rahmig](#) on Mon, 07 Mar 2022 08:08:01 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Dear all,

first of all thank you Thomas for summarizing your feedback from last workshop with Bane NOR on this topic of switch crossings. From your comments I concluded the following proposal:

- \* double switch crossings are modelled with a <switchIS>@type="doubleSwitchCrossing"
- \* single switch crossings are modelled with a <switchIS>@type="singleSwitchCrossing"
- \* switch crossings can be modelled as composition of switch parts or as single elements
- \* switch crossing parts are modelled with a <switchIS>@type="switchCrossingPart"
- \* switch crossing parts shall reference the switch crossing they belong to via @belongsToParent

Dear community, please let us know if you have any doubts with this adapted proposal. Its implementation is foreseen with upcoming railML 3.2. Any feedback is highly appreciated...

Best regards  
Christian

---