

Hello,

later than promised, here some more open discussion points of V1.00 RC1 with some comments and questions.

*** Train protection / train detection *** (posting from Georg Theeg, 01.10.2004, 10:16)

First of all: Thanks to Gregor Theeg for these detailed informations.

His suggestions:

1) distribute the train protection / train detection elements into 2 containers (<detectionElements> and <trainProtectionElements>)

[I guess, these containers are meant to be implemented as children of <ocsElements>, right?]

hmm. so far, we (normally) have containers for exactly one type of trackElement (e.g. <radiusChanges> with [only] <radiusChange>-elements, etc.). Exception: <connections> with <switch> and <crossing>.

If we introduce (as proposed) a container <detectionElements> with elements <detector>, <trackCircuitChange> and later others, this ("unwritten") convention is broken.

If we do this, we could introduce new container-elements for other groups as well, e.g. <geometryChanges> (or similar) for radiusChanges and gradientChanges.

This comment is only meant to be hint to think about the hierarchy of the trackElements. In my opinion, we should keep it consistent (not that we have a deep structure in one and a flat structure in the other group of elements.

2) introduce new element <trackCircuit> (later)

3) introduce new element <detector>
--> maybe also later?

4) discard attributes "length" and "frequency" of <trackCircuitChange>

--> question about 2) and 4): do we need <trackCircuitChange> _AND_ <trackCircuit>? Or wouldn't it suffice to have only <trackCircuit>?

5) additional attributes "insulatedRail" and "side" (or sim.) for <trackCircuitChange>

6) last paragraph of Mr. Theeg's posting:

<<<

"trainProtectionElements" in most cases are a "tracksideMagnet" or a "balise". A trackside magnet can also be a combination of 2 magnets, e.g. Signum. The basic difference is that a trackside magnet submits only its condition (1 bit), e.g. "I'm under alternate current 1000 Hz" or "I'm under direct current with polarity ...", whereas a balise submits a data telegram. Additional attributes should be the system, e.g. "PZB90", "Signum", "ZUB123", "ETCS Level1", "Crocodile",...; and the type of the element (Bauform).

>>>

Question to these lines:

- in the current scheme, we have the containers <trainProtectionElements> _AND_ <balises>. Do you think balises should be reordered as a child of trainProtectionElements?

*** crossings ***

suggestion from Gregor Theeg (01.10.2004, 15:13):

- additional attributes "ID1" and "ID2" for <crossing>
--> since a <crossing> has (normally) two <connection>-elements, and each of them has a "connectionID", we can use these for identifying the two partial switches. Do you agree with me?

Best regards
Matthias Hengartner

Tel G (NEU!!): ++41 1 633 68 16
hengartner@ivt.baug.ethz.ch

Hello,

Thanks to Matthias Hengartner for his ideas. My answer to his suggestions:

> [I guess, these containers (<detectionElements> and
> <trainProtectionElements>) are meant to be implemented as children of
> <ocsElements>, right?]

Yes.

> hmm. so far, we (normally) have containers for exactly one type of
> trackElement (e.g. <radiusChanges> with [only] <radiusChange>-elements,
> etc.). Exception: <connections> with <switch> and <crossing>.
> If we introduce (as proposed) a container <detectionElements> with elements
> <detector>, <trackCircuitChange> and later others, this ("unwritten")
> convention is broken.

I would have no problem with introducing 2 containers "detectors" and
"trackCircuitBorders" instead of one container "detectionElements" resp.
"tracksideMagnets" and "balises" instead of "trainProtectionElements". I
think, each of these solutions would work as well as the other.

> 3) introduce new element <detector>
> --> maybe also later?

If nobody urgently needs it, o.k.

> 4) discard attributes "length" and "frequency" of <trackCircuitChange>
> --> question about 2) and 4): do we need <trackCircuitChange> _AND_
> <trackCircuit>? Or wouldn't it suffice to have only <trackCircuit>?

If we define the borders of track circuits within the element
trackCircuit, we usually have to define them twice because a track circuit
border usually limits two track circuits. Another solution would be to
define it once in one of the two adjoining track circuits (Which one?) and
refer to it in the other track circuit. Or we define them outside the
track circuit itself and have 2 references.

A border between track circuits is not always an insulated rail joint.
There are also solutions like S-shaped loops or just limiting the track
circuit by the length of the rail. Thus the track circuit borders will get
several attributes. Thus, I prefer to define them as an extra element.

> - in the current scheme, we have the containers <trainProtectionElements>
> _AND_ <balises>. Do you think balises should be reordered as a child of
> trainProtectionElements?

I intended to leave out the container "balises" and to make one container "trainProtectionElements" which contains balises and tracksideMagnets. But as mentioned above, we can also have 2 containers instead.

> *** crossings ***

I'm afraid I don't understand well the idea of 0...3 connections. 0 connections are for crossings, 1 for single crossing switches (EKW) and 2 for double crossing switches (DKW), am I right? But 3 connections? This means that a EKW has 1 connection ID (Right or wrong?), but for interlocking it needs 2 because it has 2 independent point machines with 2 independent pairs of blades.

Yours
Gregor Theeg

--

Dipl.-Ing. Gregor Theeg
wissenschaftlicher Mitarbeiter
Professur für Verkehrssicherungstechnik
der TU Dresden

Tel.: 0351 / 463 36542
Fax: 0351 / 463 36644

Subject: Re: V1.00 RC1: open discussion points, part 2
Posted by [Matthias Hengartner](#) on Tue, 02 Nov 2004 11:25:57 GMT
[View Forum Message](#) <> [Reply to Message](#)

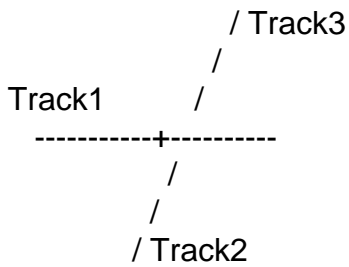
Hello,

> I'm afraid I don't understand well the idea of 0...3 connections. 0
> connections are for crossings, 1 for single crossing switches (EKW) and 2
> for double crossing switches (DKW), am I right? But 3 connections?
> This means that a EKW has 1 connection ID (Right or wrong?), but for
> interlocking it needs 2 because it has 2 independent point machines with 2
> independent pairs of blades.

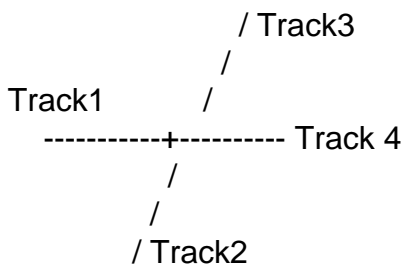
*** <answer> ***

If we have a <track> (Track1, see below) with any <crossing> (+), there are 2 <connection>-elements, one "incoming" and connected with Track2, the other "outgoing" and connected with Track3. So it doesn't matter if it's a

single/double crossing switch or a simple crossing.



The possibility to have 3 <connection>-elements is (as far as I know) only needed if we have a crossing on a <trackBegin>/<trackEnd>. In the case which is illustrated below, there is a crossing at the end of Track1, and we have 2 "outgoing" connections to Track3 and Track4 and 1 "incoming" connection to Track2.

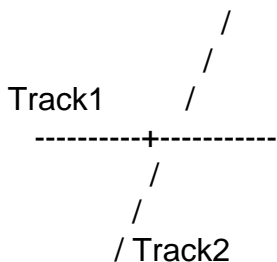


*** </answer> ***

*** <idea> ***

Some days ago, I had an idea of another way of modelling a crossing, which should also be possible:

If two tracks (see below) cross each other, it should be possible, that BOTH of them are continuous.



At least in the case of a "simpleCrossing" it should be possible to do it this way.

How to implement this?

One possible way is the following:

Given the picture above and assuming that Track1 goes from the left to the right and Track2 from bottom (left) up (right)

So we could have the following implementation in railML:

in Track1:

```
<crossing>
  <connection connectionID="C1a" branchIDRef="C2b"
branchTrackIDRef="Track2" orientation="outgoing"/>
  <connection connectionID="C1b" branchIDRef="C2a"
branchTrackIDRef="Track2" orientation="incoming"/>
</crossing>
```

in Track2:

```
<crossing>
  <connection connectionID="C2a" branchIDRef="C1b"
branchTrackIDRef="Track1" orientation="outgoing"/>
  <connection connectionID="C2b" branchIDRef="C1a"
branchTrackIDRef="Track1" orientation="incoming"/>
</crossing>
```

When the crossing is a right-angled "simpleCrossing", the orientation would be "rightAngled", of course.

Advantage of this solution:

There is a <crossing>-element in both tracks, so it's "symmetric" and both track are treated equally.

Disadvantages:

- 1) There are 2 <crossing>-elements for 1 crossing
- 2) Due to 1), there is some redundancy (which could also result in inconsistency)

I'll give you another possible way of implementing this later (in another

context).

*** </idea> ***

Best regards
Matthias Hengartner

Tel G: +41 1 633 68 16
hengartner@ivt.baug.ethz.ch

Subject: Re: V1.00 RC1: open discussion points, part 2
Posted by [Gregor.Theeg](#) on Wed, 03 Nov 2004 16:54:15 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,

> If we have a <track> (Track1, see below) with any <crossing> (+), there are
> 2 <connection>-elements, one "incoming" and connected with Track2, the
> other "outgoing" and connected with Track3. So it doesn't matter if it's a
> single/double crossing switch or a simple crossing.

For interlocking, switches, derailleurs (A derailer is a switch with the diverging track ending in the ballast.) and crossings are regarded as one element, whereas single crossing switches (German: EKW), double crossing switches (German: DKW) and three way switches consist of 2 two elements. Thus, when modelling route interlocking, we have to refer to:

- connectionID for single and double crossing switches and three way switches
- elementID for simple crossings and derailleurs
- connectionID or elementID for single switches

This is a bit confusing, especially as most elements have both, an elementID and one or more connectionID's. I think, there should be a clearer reference attribute.

Another problem ist that in the special case, where a <trackBegin>/<trackEnd> is in a crossing switch, there would be no use for the third connectionID.

Therefore, I suggest to give single and double crossing switches and three way switches no "elementID", but an "elementGroupID", an "elementID1" (for the "lower" part in direction of internal milage of the main track) and an "elementID2" (for the "upper" part). What do you think about this?

Regards.

--

Dipl.-Ing. Gregor Theeg
wissenschaftlicher Mitarbeiter
Professur für Verkehrssicherungstechnik
der TU Dresden

Tel.: +49 / 351 / 463 36542
Fax: +49 / 351 / 463 36644
<http://vsite.tu-dresden.de>
