
Subject: [railML3] How to assign a mileage change to a netElement
Posted by [Karl-Friedemann Jerosch](#) on Thu, 16 Sep 2021 14:58:06 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear railML users,

equipping a railway line with ETCS, a high-precise measurement of the track layout is required, because the ETCS system can calculate distances with accuracy of decimeter (1/10 of meter). As result of the exact measurement, the lengths of parallel netElements will be different and therefore mileage changes have to be considered.

For example, a siding track can be longer then the parallel continuous main track, as shown in file https://forum.railml.org/userfiles/2021-08-24_siemens_railml_3-modellingmileagechanges_v01.pdf.

Using the anchor-elements introduced with railML3.2-beta1 (see detailed explanation in file https://forum.railml.org/userfiles/2020-09-07_railml_railml3-modellingexamples-mileagechanges.pdf), it is not possible to assign an anchor-element to exact one netElement.

Best regards
KARL

Subject: Re: [railML3] How to assign a mileage change to a netElement
Posted by [christian.rahmig](#) on Thu, 16 Sep 2021 19:25:03 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear Karl,

thank you for the difficult challenge that you are bringing up with this... Several solutions may be applicable. Let me focus on some of them:

1) Decoupling of mileage and physical length

We use `<linearPositioningSystem>` for defining a linear reference system traditionally used for railway location. In history, such linear reference systems were valid for a whole railway line. This means that e.g. for a double track line both tracks got the same positioning reference even if one of the tracks was longer due to its "outer role" in a curve. Anyway, there is only one location reference system for one railway line. This is the situation that suits to the modeling approach of railML 3.2.

In order to fulfill the accuracy requirements from ETCS (every track has an individual length), it could be one idea not to use the linear reference system for that, but a separate length attribute. Every `<netElement>` has a length attribute and we could use it to specify the exact physical length of the track segment.

2) Using of different <linearPositioningSystem> elements

Another idea may be the definition of track individual linear positioning systems. In consequence, every <netElement> would reference its own <linearPositioningSystem>, where the anchor points define the mileage changes. The existing model of railML 3.2/3.1 can cope with this.

However, the remaining challenge: how to connect the different <linearPositioningSystem> instances that together form the linear positioning system for the railway line. We could probably think of some kind of hierarchy as we have it defined for other elements, too. For example, a (track individual) <linearPositioningSystem> can refer to the (line) <linearPositioningSystem> using a @belongsToParent reference. Common attributes like begin mileage and end mileage will be defined in the parent element, while the mileage changes would be defined in the child elements.

3) Defining mileage changes as children elements of <netElement>

This solution sounds straightforward: remove the <anchor> element from the <linearPositioningSystem> and place it somewhere under the <netElement>, for which it is valid.

The drawback of this solution is that it breaks the clearness of the model: So far, a <netElement> can be seen as dimension-less atomic element of a topology. The dimension information (in terms of coordinates) comes with the <linearPositioningSystem> referenced by the <netElement>. Both "worlds" are separated. When the <anchor> becomes a child element of <netElement> it will refer to (positioning) values that are only understandable in combination with the referenced <linearPositioningSystem>. Without the reference to the <linearPositioningSystem> the information of the <anchor> element is lost/useless.

These are three possible solutions, and there are more. From my personal perspective, I like the solution number 2 most. But let's forward the question towards the community: What is your favourite?

Any feedback is highly appreciated...

Best regards
Christian

Subject: Re: [railML3] How to assign a mileage change to a netElement
Posted by [Karl-Friedemann Jerosch](#) on Fri, 17 Sep 2021 08:28:01 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear Christian,

thank you very much for providing solutions for the "mileage changes" topic.

General:

In our way of ETCS trackside engineering, we assign a measured kilometer value to each track element.

Track elements with a linear expansion (like a level crossing or a platform) get an start kilometer value and an end kilometer value.

Therefore, currently we plan to use the netElements as shown below (Example corresponds to the figure of Example 01 in

[https:// forum.railml.org/userfiles/2021-08-24_siemens_railml3-modellingmileagechanges_v01.pdf](https://forum.railml.org/userfiles/2021-08-24_siemens_railml3-modellingmileagechanges_v01.pdf)):

```
<netElement id="ne_13" length="2001.1">
  <relation ref="nr_11_1_13_0"/>
  <relation ref="nr_12_0_13_0"/>
  <relation ref="nr_13_1_14_0"/>
  <relation ref="nr_12_1_13_1"/>
  <associatedPositioningSystem id="ne_13_aps02">
    <intrinsicCoordinate id="ne_13_aps02_ic1" intrinsicCoord="0.0">
      <linearCoordinate positioningSystemRef="lps02" measure="11000.0"/>
    </intrinsicCoordinate>
    <intrinsicCoordinate id="ne_13_aps02_ic2" intrinsicCoord="1.0">
      <linearCoordinate positioningSystemRef="lps02" measure="13001.1"/>
    </intrinsicCoordinate>
  </associatedPositioningSystem>
</netElement>
```

As shown, we provide the measured kilometer value for the start of an edge and the end of an edge, the attribute length is exactly the difference between start kilometer value and end kilometer value of this edge.

Feedback to the suggestions given in this post:

I understand from your suggestion 1), that the linearPositioning-System can be considered as completely independent from the physical length of the edges. If so, then there is no problem for me, because it is possible to place a mileage change always at a start or an end of the edge. Software tools can check if the provided end kilometer value of the previous edge is the same as the start kilometer value of the connected following edge. If both kilometer values are not identical, then there is a mileage change.

Consequences:

Mileage changes are a little "hidden". But for me, it would be ok.

I like the idea of the anchor elements, because they provide an universal solution for all kinds of mileage changes (gaps, overlaps, compressed and stretched kilometers). But with the solution of "hidden mileage changes", the anchor elements will not often be used, I think.

If you are fine with the "hidden mileage changes" as described, then my problem is solved. Otherwise, if you want that we shall decide for exactly one of the 3 provided solutions presented in your forum post, then my favorite solution would be suggestion 3.

Best regards
KARL

Subject: Re: [railML3] How to assign a mileage change to a netElement
Posted by [Heidrun Jost](#) on Fri, 01 Oct 2021 10:28:34 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear all,

we propose to consider the mileageChange as a new infrastructure element.

The goal is to create a relation to a netElement and have one unique mileageChange element which is topology related.

PRO:

- mileageChanges can be explicitly placed in the infrastructure.
- There is one common way to engineer the mileageChange instead of two (anchor, mileage post definition) as proposed in the mileageChange document from RailML3.2.

CONTRA:

- mileageChanges cannot be engineered outside netElement related information. However, as the netElements are the topology elements that build the relation
- between the graph and measure (km, m, mileage) information and all other elements have relations to these net elements it should be ok.

Please find a code snippet of our proposal with two variants:

```
<infrastructure id="is_01">
  <functionalInfrastructure>
    <mileageChanges>
      <!-- Mileage Gap: Changes the measure value from 500 to 1500. -->
      <!-- First proposal with mileage change engineered by two
spotLocations (from:500/to:1500) -->
      <mileageChange id="mc_01">
        <change type="gap" (optional) from="mc_01_spl_01"
to="mc_01_spl_02"/>
        <spotLocation id="mc_01_spl_01" netElementRef="ne_01"
```

```
applicationDirection="normal" pos="500" intrinsicCoord="0.1">
  <linearCoordinate positioningSystemRef="lps01" measure="500"/>
</spotLocation>
  <spotLocation id="mc_01_spl_02" netElementRef="ne_01"
applicationDirection="normal" pos="500" intrinsicCoord="0.1">
  <linearCoordinate positioningSystemRef="lps01" measure="1500"/>
</spotLocation>
</mileageChange>
<!-- Alternative approach with only ONE spotLocation information
per mileage change and measureChange value-->
<mileageChange>
  <change type="gap" refLocation="mc_01_spl_01" measureChange="1500"/>
  <spotLocation id="mc_01_spl_01" netElementRef="ne_01"
applicationDirection="normal" pos="500" intrinsicCoord="0.1">
  <linearCoordinate positioningSystemRef="lps01" measure="500"/>
</spotLocation>
</mileageChange>
</mileageChanges>
</functionalInfrastructure>
</infrastructure>
```

Best regards,

--

Heidrun Jost
Data Manager
Transportation Systems
Thales Deutschland GmbH

Phone: +49 (0) 30 688306 423
Schützenstr. 25 – 10117 Berlin – Germany

Subject: Re: [railML3] How to assign a mileage change to a netElement
Posted by [Karl-Friedemann Jerosch](#) on Tue, 19 Oct 2021 15:36:36 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear Heidrun,

first thank you very much for providing additional 2 solutions for the described problem how to assign mileage changes on netElements.

In the 2nd solution using 1 spot location only, in my opinion there is a risk of misinterpretation, because it is not obviously clear, if the spot location provides the line kilometer value before or after the kilometer jump.

Conclusion:

Comparing all 3 solutions provided by Christian and the 2 solutions provided by Heidrun, I prefer Heidrun's first solution, where a mileageChange is using 2 spot locations describing the identical location on the track, but each of both with different line kilometer values.

Subject: Re: [railML3] How to assign a mileage change to a netElement
Posted by [christian.rahmig](#) on Thu, 11 Nov 2021 13:47:09 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear all,

I concluded the discussion in our new railML ticketing system: you can find the problem description and the solution proposal as Gitlab issue #478 [1].

The approach proposed by Heidrun allows for explicit modelling of gaps and overlaps. But what about compressed and extended kilometers? Is there a need for modelling them with <mileageChange> elements, too, or shall we remain with the <linearPositioningSystem / anchor> solution?

Any comments are highly appreciated...

[1] <https://development.railml.org/railml/version3/-/issues/478>

Best regards
Christian

Subject: Re: [railML3] How to assign a mileage change to a netElement
Posted by [christian.rahmig](#) on Sun, 20 Feb 2022 17:27:37 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear all,

if there are no objections until February 28, 2022, I conclude that there is no interest / need for modelling compressed or stretched kilometers with the new infrastructure element <mileageChange> and we will implement the new element for gaps and overlaps as described in Git issue #478 [1].

[1] <https://development.railml.org/railml/version3/-/issues/478>

Best regards
Christian
