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 seperated. 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
