Subject: Re: Modelling changes of mileage direction
Posted by Martin Karlsson on Mon, 27 Aug 2018 14:06:44 GMT
View Forum Message <> Reply to Message

How RTM intends to map linear coordinates onto a track is not explicitly explained. I have made the assumption that the mapping in the simplest case is done by creating two IntrinsicCoordinate objects in each AssociatedPositioningSystem. They would have the intrinsicCoord attributes set to 0 and 1 respectively, and point to a LinearCoordinate indicating the mileage at the beginning and end of the track, which is represented by the owning PositioningNetElement. The linear coordinates along the track could then be obtained by interpolation.

The RTM experts would need to confirm that my assumption is correct. if so, let's examine the more complex cases.

First, mileage counting could be in the opposite direction to the track. In this case, the mileage coordinate at 0 would just have a higher value than at 1.

Next case is that there is a change of reference system somewhere in the track. Say that the track starts at 52.250 km. After 3 km (i.e. at 55.250) we change to 20,000 km. This reference is valid to the end of the track, at 22.000 km.

In this case, the PositioningNetElement would have two AssociatedPositioningSystem objects. The first one would have an intrinsicCoordinate 0, pointing to a LinearCoordinate 52,250 km, and an IntrinsicCoordinate 0.6, pointing to 55.250 km. The second AssociatedPositioningSystem would have an intrinsicCoordinate 0.6, pointing to 20.000 km, and an intrinsicCoordinate 1, pointing to 22.000 km. At the very point of change, mileage cannot be unambiguously determined, but that lies in the nature of this case. There are two correct mileage positions for this point.

This would also work even if one of the two systems was to count in the opposite direction from the other, which was the original question in this forum post.

Next case would be a mileage change within the same reference system ("missing" or "overlapping" mileage, in the terminology of railML 2). This can be solved by introducing additional coordinates. Let's say that the mileage count at one point will "jump" from 55.948 to 56.000. We would then introduce two LinearCoordinates with these values, both referred to with the same intrinsicCoordinate. Also here, we would get the effect that two mileage positions would be correct at the point of change. But on either side, we can find the correct position by interpolation.

Finally, what if the mileage change in the last example is handled not by a gap in the counting, but by "compressing" the length of km 55 (i.e. 55.500 would in effect mean 474 meters after the 55 km post)? In this case, we would just exclude the 55.948 coordinate. The position of the 56.000 coordinate would indicate that the previous km is shorter than the nominal value.

So the answer seems to be yes, the model works for all cases if used consistently in this way.

However, I suspect that I am missing something important here, since the LinearAnchorPoint object was not necessary to use in any of my use case descriptions. What is the intention with this class?