Subject: [railML3|alpha] Suggestion for an enhanced topology model
Posted by Martin Karlsson on Fri, 07 Oct 2016 13:35:33 GMT
View Forum Message <> Reply to Message

The idea of modelling a railway network topology as a mathematical graph comes very naturally. After all, the network layout even looks like a graph, with the tracks as edges and the switches and crossings as nodes.

Deceived by this intuitive picture, it actually took me a long time to realise that the RailTopoModel is modeled the other way around. The tracks are nodes in the graph, not edges! The edges are instead the passages through switches and crossings, which connect the tracks.

I guess there were good reasons for choosing this model, even though I don't think I'm the only one to make this mistake. But the comprehension of the model is a secondary issue. What bothers me more is the fact that the switches and crossings are not at all included in the model.

This is a problem, because the switch or crossing as such implies a restriction on the use of the relations through it. Take the example of a diamond crossing. We will have four tracks, connected in pairs by navigable relations. The model does not tell us that there is a restriction against using both these relations for traffic at the same time. But many applications are dependant on that information. Interlockings obviously, but also CTC pretest function, TMS traffic optimisation, time table simulation tools etc. Some of these may not be railML use cases today, but they will for sure be in the future.

I believe railML needs to cover the gaps that RTM leaves open in this respect. My suggestion is to introduce topology relations on asset level.

Please look at the UML diagram below. In addition to the well known RTM classes, I have introduced the classes Track and NetworkNode. NetworkNode is an abstract base class with a number of possible descendants, shown here is Switch. The Track has a one-to-one relation with a LinearNetElement, and two relations to the nodes at each end. Depending on the subclass, the node will have relations to relevant PositionedRelations. In the case of the switch, the relations indicate which course of the switch each PositionedRelation corresponds to (this eliminates the need for the SwitchBegin and SwitchEnd assets).

This amendment would make the model of the topology complete, from a navigation and conflict perspective. By also including the length of each track as an attribute, we have all information for modeling or simulating network usage.

I believe Track and NetworkNode should be contained under Assets in the model. I saw that a Track class has been added to the model in 3.0.is4, but under Operation. In my opinion, Operation should be limited to logical designations (e.g. these objects form a station), whereas the track is a real life hardware asset. Switch is already listed as an asset. Crossing was also, but was removed in this version, so it would need to be put back.

I suggest however to not make Track a subclass of Asset. It would work, but the schema would then allow assigning the Track to an arbitrary location. I think it is better to restrict the localisation to a single LinearNetElement also in the schema. In the same logic, NetworkNode does not really need to be an Asset descendant either. It is localised through its track relations.

Martin Karlsson

Bombardier Transportation
Rail Control Solutions
EAPD/ECC
S-405 02 Göteborg, Swede

File Attachments

1) Asset level topology.PNG, downloaded 1234 times