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 1277 times

Subject: Re: [railML3|alpha] Suggestion for an enhanced topology model

Posted by Alain Jeanmaire on Thu, 27 Oct 2016 15:06:38 GMT
View Forum Message <> Reply to Message

Dear Martin,
Thank you for your interest to this major evolution, and for those relevant questions. I will try below to give you a clear explanation; do not hesitate to come back if any need for further details.

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.
--> if you refer to IRS 30100, this major (and disturbing) feature is clearly mentioned for attention (chapter 1.5 Modeling principles)
The tracks are nodes in the graph, not edges! OK The edges are instead the passages through switches and crossings, which connect the tracks.
--> at this level you should stick to pure TOPOLOGY, that is nodes and edges, and not speak about pieces of equipment as iron tracks, switches or crossings, which will appear in the functional layer.
--> up to now, at toplogy level, the network is described independatly of the mean of transportation; i.e. can be road, rail or pedestrian path.
--> so, in a conexity graph, all topology objects are nodes, and edges only relate relations between topology objects.

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.
--> at this toplogy level, we definitely do not YET include functional objects as switches and crossings
--> those functional objects will be modeled furher as "NetEntity"

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.
--> those capacities or restrictions are perfectly modeled in the topology layer, using the notion of "navigability", within the "PositionedRelation", independantly of the function/equipement which will ensure this capacity or restriction.

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).
--> as mentioned above, a main structuring principle of RTM is not to mix topology and functional objects, then split the system into layers: topology, functional objects, physical assets,...
--> the functional objects (switch, track,... are covered by "NetEntity".

--> Your proposal goes with different principles... as mentioned in IRS and in the earlier feasibility study, this choice is related to the limitation encountered in the multiple previous experiences, with the clear will to be agnostic of any usages and therefore open to any further development, with minimum risk of refactoring.

I hope those explantations will clarify the choice we have done for long term with RTM and railML3.

Alain jeanmaire / Gilles Dessagne
SNCF Réseau

Subject: Re: [railML3|alpha] Suggestion for an enhanced topology model
Posted by Martin Karlsson on Mon, 31 Oct 2016 12:21:09 GMT

View Forum Message <> Reply to Message

Dear Alain,

Thank you for your clarifications.

I appreciate the philosophy of separating pure topology from physical objects, and I do not suggest to change this fundamental principle, or at all to change anything in RTM. As I wrote, I was sure there were good reasons behind chosing this model, and you have clarified that further.

My suggestion is rather to add - in railML, not RTM - a second layer of information, to reflect the topology related properties of the physical objects, in particular
- dependence of relation navigability on the run time state of a physical object (i.e. the course of a switch)
- mutual exclusivity of navigable relations (in the case of diamond crossings)
- length of tracks

For sure, my suggestion is not the only way to address these issues, but they need to be adressed one way or another.

Subject: Re: [railML3|alpha] Suggestion for an enhanced topology model
Posted by Alain Jeanmaire on Mon, 14 Nov 2016 18:24:02 GMT
View Forum Message <> Reply to Message

Dear Martin,

thank you for your feed back.
we will work on your suggestion and come back to you.

Best regards
Alain Jeanmaire

Subject: Re: [railML3|alpha] Suggestion for an enhanced topology model
Posted by Matthieu Perin on Mon, 13 Feb 2017 14:32:32 GMT
View Forum Message <> Reply to Message

Hi,

If I may give my opinion about the philosophy behind a Topological "pure" part for the RTM meta model is good, but I see two major issues with the objects then :

If we need only a Topological view, why there have to be difference between linear and non-linear elements ?. Such differences does not exist in the topological level, as these elements are all nodes and can be navigate whatsoever. If we keep separated Linear and Non Linear element for clarity purpose (which is good in my opinion) then the two object MUST have somme differences : e.g. a linear element should have a "length" attribute that is not existing for the non-linear one.

I hope It can help thinking about the meta model !

Subject: Re: [railML3|alpha] Suggestion for an enhanced topology model
Posted by christian.rahmig on Fri, 17 Feb 2017 13:52:09 GMT
View Forum Message <> Reply to Message

Dear Matthieu,

thank you very much for your feedback. Together with other points, this issue will be discussed within the RailTopoModel Expert Group at their next phone conference on 01.03.2017. So, I hope to provide you an exhaustive answer shortly afterwards.

Best regards
Christian

Am 13.02.2017 um 15:32 schrieb Matthieu Perin:
> Hi,
> If I may give my opinion about the philosophy behind a
> Topological "pure" part for the RTM meta model is good, but
> I see two major issues with the objects then :
>
> If we need only a Topological view, why there have to be
> difference between linear and non-linear elements ?. Such
> differences does not exist in the topological level, as
> these elements are all nodes and can be navigate whatsoever.
> If we keep separated Linear and Non Linear element for
> clarity purpose (which is good in my opinion) then the two
> object MUST have somme differences : e.g. a linear element
> should have a "length" attribute that is not existing for
> the non-linear one.
> I hope It can help thinking about the meta model !

--
Christian Rahmig - Infrastructure scheme coordinator
railML.org (Registry of Associations: VR 5750)
Phone Coordinator: +49 173 2714509; railML.org: +49 351 47582911
Altplauen 19h; 01187 Dresden; Germany   www.railml.org

Subject: Re: [railML3|alpha] Suggestion for an enhanced topology model
Posted by christian.rahmig on Mon, 20 Nov 2017 16:24:56 GMT
View Forum Message <> Reply to Message

Dear all,

now that we are approaching towards railML 3.1 release, I want to answer on your questions/requirements and how they have been implemented in the new railML version:

Am 31.10.2016 um 13:21 schrieb Martin Karlsson:
> [...]
>
> My suggestion is rather to add - in railML, not RTM - a
> second layer of information, to reflect the topology related
> properties of the physical objects, in particular
> - dependence of relation navigability on the run time state
> of a physical object (i.e. the course of a switch)

By introducing the <switch> child element <switchBranch>, which may reference a <netRelation> element, it is now possible to have a clear linking between topology and functional infrastructure element. This solution keeps the strict separation between both "layers", but ensures that switches are correctly located within the railway topology network.

> - mutual exclusivity of navigable relations (in the case of
> diamond crossings)

The mutual exclusivity of navigable relations at diamond crossings is a task to be realized on operational or interlocking level, because from topology point of view it is possible to travel all paths like it is possible to travel both branches at a switch. There is no need to define an exclusivity parameter at the topology layer. In order to ensure that the paths are not used at the same time by different vehicles, the switch/crossing itself has the different child elements <*branch>, which can be used for that purpose: In general, only one <*branch> can be travelled at the same time.

> - length of tracks

The functional infrastructure element <track> has also a parameter @length (in meters).

Best regards
Christian

--
Christian Rahmig - Infrastructure scheme coordinator
railML.org (Registry of Associations: VR 5750)
Phone Coordinator: +49 173 2714509; railML.org: +49 351 47582911
Altplauen 19h; 01187 Dresden; Germany   www.railml.org