Subject: Extension suggestion for alternativeSectionTT
Posted by Torben Brand on Thu, 20 Feb 2020 12:03:12 GMT
View Forum Message <> Reply to Message

Dear TT community,

For the use case TT:LongTermStrategicTimetabling, TT:ATimetableForACompetition, TT:SlotOrdering we would like to indicate the alternative paths to be available for a train or pattern/template train of a train group to be used if the planned path is not available. This being in a simulation, short term scheduling, dispatching decisions or data preparation for long term planning.

We suggest adding an extension in railML2.4 for Norwegian sector purposes, and if the community thinks this is beneficial to add the new element in railML2.5.

The proposed solution is to copy the element <sectionTT> with all attributes and sub elements, with the new name <nor:alternativeSectionTT>. Placement would be parallel to <sectionTT under <ocpTT>.
We will add one extra attribute @rank to indicate the rank between the different elemnts, if more than one <nor:alternativeSectionTT>.
For info on @rank see separate forum posting:
https://www.railml.org/forum/index.php?t=msg&th=691& goto=2332&#msg_2332
nor:alternativeSectionTT@rank has a semantic constraint of the value 2 or higher. This as <sectionTT> is always @rank="1".

What does the community think?

Subject: Re: Extension suggestion for alternativeSectionTT
Posted by Milan Wölke   on Tue, 10 Mar 2020 14:14:59 GMT
View Forum Message <> Reply to Message

Hi Torben,

thanks for your input. I have put this topic on the agenda for the next TT developer group telco. One question in advance though, why create a new element for this at all. Shouldnt it already be enough to change the cardinalities of sectionTT and extend it with the priority. Like this we dont need the semantic constraint and it should still be semantically clear.

Best regards, Milan

Subject: Re: Extension suggestion for alternativeSectionTT
Posted by   on Tue, 10 Mar 2020 16:10:38 GMT
View Forum Message <> Reply to Message

Dear Torben,

from a general railML view, I would not be happy with your suggestion to encode alternative paths at a <trainPart>. The original idea of a <trainPart> was that it should be the elementary, most little and impartible part of a train. With that reason, we already denied requests of making other attributes of <trainPart> divisible like formation or load. We should write that we should be consequent here...

So, I would expect hat for an alternative path, the <train> must be copied and the options connected with @trainNumber, @additionalTrainNumber and @scope should be used to encode two <train> elements as alternatives (same train number, different scope and/or different additionalTrainNumber).

Everything else leads to (even more) "uncontrolled growth".
I know, live is full of compromises and for the certain Norwegian use case, your suggestion may be easier and more direct. So please understand my post rather as a remark in general, a warning, but no decision on your case - this would be up to the community and the coordinators.

Only to avoid misunderstandings:
With "alternative paths" you mean just the route of the train (way along the network) but no different arrival/departure/run times? If you only copy the element <sectionTT>, you would not be able to encode different arrival/departure/run times. This is also a general problem I see, because the background you mention ("if the planned path is not available") may concern the route as well as the times! So, to encode alternative paths in general railML, I would expect being able for alternative routes as well as alternative times.

Best regards,
Dirk.

Subject: Re: Extension suggestion for alternativeSectionTT
Posted by Milan Wölke   on Fri, 26 Jun 2020 10:27:11 GMT
View Forum Message <> Reply to Message

A ticket for modelling the requirement described here as part of railML 2.5 has been created at https://trac.railml.org/ticket/381.
The ticket is a reminder in the tracking system so this requirement is not forgotten wen finalizing railML 2.5. The actual modelling however still needs to be discussed.

Best regards, Milan

Subject: Re: Extension suggestion for alternativeSectionTT
Posted by Torben Brand on Wed, 01 Jul 2020 13:14:20 GMT
View Forum Message <> Reply to Message

Dear Dirk and TT development group,

I will explain our UC in more detail for better understanding. We want to describe the n alternate path(s) of a train. With the path we mean the route of the train/ the way along the network (also as explained in this forum posting: https://www.railml.org/forum/index.php?t=msg&th=710& start=0&). The alternative path(s) is only "minor" different to the primary path in relation to run time calculation. Usually (secondary) track 2 instead of (main) track 1 on a crossing loop. So that different arrival/departure/run times can be neglected. We would also primarily use the feature in trainParts forming the extension nor:patternTrains (norwegian extension documented on www.jernbanedirektoratet.no/railML)), that do not have any times (do not use <times>).

As we are also concerned with "uncontrolled growth" we think that the solution suggested by Milan is the easiest (increase the multiplicity/cardinality to "n" and add @priority to <sectionTT> in railML 2.5. This would bloat the xml way less than having to create a new copy of the trainPart and train with only minor changes to the orriginal.

We could also not use the attribute @scope today as it only allows a "secondary" path and not "n" paths.

We welcome a discussion about the correct use of train@scope as we know it has been used somewhat differently as described in the wiki/in this forum posting. We think that the correct focus here is, as described in the wiki, is the train@scope is a separate slot allocation (DE:Trasse). For the solution with alternative sectionTTs this is within the same slot (order/allocation).

To describe it in a different (and maybe more precise way). alternativeSectionTT has the same order of OCPs (as it only has one trainpart). Different to train@scope which has different ocps, or a shorter length of ocps between the two alternative trains/trainparts.

Subject: Re: Extension suggestion for alternativeSectionTT
Posted by   on Thu, 02 Jul 2020 09:29:01 GMT
View Forum Message <> Reply to Message

Dear Torben,

I understand your point and I agree that multiple <sectionTT> would be an easy solution.

But
 - With multiple <sectionTT>s, you could encode different tracks between two stations. That's not the example you described, as far as I understand.
 - If you want to encode different tracks within one station, you would need different <ocpTT>s, I think.
 - If so (multiple <sectionTT>s and/or <ocpTT>s will be allowed), I would expect that they should have disjunctive operating days. That may not be necessary in your UC, but it is surely a much more common need for a "minor deviance" of a slot.
 - May be @priority can be introduced additionally to operating days. The @priority would clarify what should happen in case the operating days are not disjunctive.

> So that different arrival/departure/run times can be neglected.

That may be so in your UC but it is surely not so in general. I think railML should first orientate on general context, not on special cases at the first place.

> The alternative path(s) is only "minor" different to the
> primary path in relation to run time calculation.

Surely you are aware that there is no hard border between a "minor" difference and a major. It would rather be a grey zone in between. And that's the point where I am really concerned about multiple <sectionTT>s and/or <ocpTT>s.

Currently, one can encode all deviations ("minor" and major) of a train path by copying it. With your extension, there would be two option for the same purpose. That would make import harder (since both options should be understood) and leads to wild growth.

So, for the sake of consequence, I still have to say: A <trainPart> is designed to be the elementary, atomic and impartible part of a train. If we weaken this here today, we can hardly deny to weaken it again tomorrow there and so on...

> We could also not use the attribute @scope today as it only
> allows a "secondary" path and not "n" paths.

That's a misunderstanding. The official primary key is of a train in railML is @trainNumber+@scope+@additionalTrainNumber. So, there can be "n" paths with the same @trainNumber and @scope=primary, they only need to be counted by @additionalTrainNumber=1, 2, 3... and so on.

And more, you do actually _not_ need to copy/repeat all the <trainPart> if you only want to encode a "minor" deviation in one <sectionTT> or <ocpTT>: You could use @scope=secondaryInner and only repeat the "inner" (minor) part of the path which deviates from the primary. Again, you could do that "n" different times by using @additionalTrainNumber=1, 2, 3...

That's the "official" way to do it now and that's why I say there would be a redundancy if you allow one more other way. However, I am no big friend of the current model, but I think that more important than that is consequence and less effort on import.

Best regards,
Dirk.

Am 01.07.2020 um 15:14 schrieb Torben Brand:
> Dear Dirk and TT development group,
>
> I will explain our UC in more detail for better
> understanding. We want to describe the n alternate path(s)
> of a train. With the path we mean the route of the train/

> the way along the network (also as explained in this forum
> posting:
> https://www.railml.org/forum/index.php?t=msg&th=710& start=0&).
> The alternative path(s) is only "minor" different to the
> primary path in relation to run time calculation. Usually
> (secondary) track 2 instead of (main) track 1 on a crossing
> loop. So that different arrival/departure/run times can be
> neglected. We would also primarily use the feature in
> trainParts forming the extension nor:patternTrains
> (norwegian extension documented on
> www.jernbanedirektoratet.no/railML)), that do not have any
> times (do not use <times>).
> As we are also concerned with "uncontrolled growth" we think
> that the solution suggested by Milan is the easiest
> (increase the multiplicity/cardinality to "n" and add
> @priority to <sectionTT> in railML 2.5. This would bloat the
> xml way less than having to create a new copy of the
> trainPart and train with only minor changes to the
> orriginal.
>
> We could also not use the attribute @scope today as it only
> allows a "secondary" path and not "n" paths.
>
> We welcome a discussion about the correct use of train@scope
> as we know it has been used somewhat differently as
> described in the wiki/in this forum posting. We think that
> the correct focus here is, as described in the wiki, is the
> train@scope is a separate slot allocation (DE:Trasse). For
> the solution with alternative sectionTTs this is within the
> same slot (order/allocation).

Subject: Re: Extension suggestion for alternativeSectionTT
Posted by Stefan Hubrig on Mon, 24 Aug 2020 06:32:59 GMT
View Forum Message <> Reply to Message

Hello,

I support the idea of having alternative sections/itinerary in railML 3.x. There is more than one system which uses this for planning.

Best regards,
Stefan

Subject: Re: Extension suggestion for alternativeSectionTT
Posted by Jörg von Lingen   on Tue, 25 Aug 2020 14:48:07 GMT

Dear all,

just for information we have already in interlocking the possibility to define
itineraries under controller element. Then any train in TT can refer to this
defined itineraries.

Best regards,
Joerg v. Lingen - Interlocking Coordinator
Am 24.08.2020 um 08:32 schrieb Stefan Hubrig:
> Hello,
>
> I support the idea of having alternative sections/itinerary
> in railML 3.x. There is more than one system which uses this
> for planning.
>
> Best regards,
> Stefan

File Attachments
1) EA18.png, downloaded 250 times

Subject: Re: Extension suggestion for alternativeSectionTT
Posted by Milan Wölke   on Tue, 06 Oct 2020 10:41:23 GMT

Hi Jörg,
thanks for your suggestion. I think for railML 3 this approach is something to be considered.
However since the requirement is focussed on railML 2.5 I am afraid we need to go for a different
solution.

Best regards, Milan

Subject: Re: Extension suggestion for alternativeSectionTT
Posted by Milan Wölke   on Tue, 06 Oct 2020 10:49:17 GMT

Hi all,

to keep you guys posted, this topic has been discussed in the last timetable developer telco. We
decided to introduce a new element for the <ocpTT> named <alternativeSectionsTT>. Starting
with railML 2.5 it will be possible to define alternative routes between the current and the following
ocp along with a priority to help deciding which to choose if the default one specified as before
with <sectionTT> does not seem appropriate anymore in operation. The idea here is that an

alternative route does not impact the arrival and departure times of the followup stop.

The changes have been commited to the repository and can be viewed in the railML 2.5 development branch available here (https://svn.railml.org/railML2/branches/railML2.5-dev/).

Best regards, Milan

Subject: Re: Extension suggestion for alternativeSectionTT
Posted by   on Fri, 09 Oct 2020 13:33:02 GMT
View Forum Message <> Reply to Message

Dear Milan,

we are by far not happy about this decision. We see raised effort on import and the danger for uncontrolled growth due to the overlay of this and other solutions how to encode certain situations.

If railML introduces this <alternativeSectionsTT>, I expect a clear rule when and for what to use it on export. I see many potential alternative routes which could be exported and we must avoid misunderstandings. For instance, in Germany, it is common that nearly all trains can use the opposite track (of a double-track line) without a change of timetable as well as all compatible other tracks in a station. When I export a railML file, shall I encode
 a) all technological possible alternative routes as <alternativeSectionsTT>? (Means: nearly all opposite tracks?)
 b) all alternative routes which can be used with the same schedule or run times?
 c) all possible alternative routes which are allowed without a change of timetable? (Means also some deviations in junction areas?)
 d) ... what else is the purpose of <alternativeSectionsTT>?

Best regards,
Dirk.