Subject: Re: More detailed 'speed change' definitions
Posted by on Fri, 29 Jun 2012 17:50:09 GMT
View Forum Message <> Reply to Message

Dear Susanne and all others,

thank you for the good conclusion on speed changes!

> Please check the new proposal for your needs and give us appropriate
> feedback. (We are also pleased for positive feedback like "All is fine
> for us"

This would be nice… But it is unfortunately not so easy. I think, for iRFP, we can implement a programming based on the changes you introduced.. If this works well - which I would expect - we could send you an “All is fine for us” from iRFP. As you know, sometimes one only recognises the wishes when implementing the solution.

So far, I wrote some notes below. Most are minor issues, some are possibly important. I think we should consider now the question of the relations (see below). Depending on this, we could start implementing possibly in some weeks.

I did not reply to all of your issues - they are welcome and understandable. So no objection from our side. And thank you again for the good work.

<speedChange id="sc3"...>

Can we have the ‚id’ of a speed change as an optional attribute? Since we have many, many speed changes and seldom the need to refer to them I do (optionally) want to save some space and effort.

> axleLoad="40"

would be interesting to know the unit of ‘axleLoad’ ;-)

> ** The axleLoad attribute is of the railML type tWeightTons.

Oh, thank you. One would not expect it with this value, even not in Australia… ;-)

If they talk about ’40-Tonner’ on streets, I guess they mean _total_ weight - not axle weight (I hope)... ;-)

> minimumBrakePercentage="123"

There are at least two minimum brake percentages for a speed profile: One for brake switch G and another for the other brake switches (P, R). This applies to the middle-European (UIC/RIC) practice of two-valued braking tables. (The other braking switches result in different braking percentages of the vehicles but not in different braking tables.)

In General, we should consider to make the speed profile (section) depending on the triplet
”brakeType”+“minimumBrakePercentage”+”minimumBrake Switch” (see news entry
from 26.04.2012).

> minimumBrakePercentage="123"

This does not take into account the aspects written on 26.04.2012: There is a "minimumBrakePercentage" for each section of a speed profile between two places where trains can start or end (i. e. between two stations). So, I would prefer to add the above named triplet as attributes of a speed change: They are valid until the next speed change with such attributes.

> axleLoad="40"
> meterLoad="8"

For non-Boolean conditions I would prefer to clarify the relation they are meant with their name: maximumAxleLoad or minimumAxleLoad?

Is the speed profile necessary for trains with an axle load of more than ‘40’ (probably declaring speed restrictions) or for trains with an axle load of less than ‘40’ (probably permitting raised speeds)?

We have done so with ‘minimumBrakePercentage’ so we should also do it with the others.

We can also describe the relation explicitly, defining an (obligatory) attribute with the enumerations ‘lessThan’, ‘greaterThan’, ‘lessOrEqual’, and ‘GreaterOrEqual’. However, we should somehow clarify that, either implicitly in the name or explicitly. We should not leave it to the documentation alone.

There are some attributes of <speedProfile> which act as conditions: The <speedProfile> is valid if the train fulfils the condition (axleLoad, meterLoad, operatingPeriodRef, nationalSystem). At least if there are more than one such conditions, there is the question: Are they linked with logical AND or with logical OR?

Is the speed profile valid for the train if it has an axleLoad of more than xxx tons AND a meter load of more than yyy tons per meter?
Or for if the train has an axleLoad of more than xxx tons OR a meter load of more than yyy tons per meter?

This may be left to the documentation but it is not very easy. It depends at least on the attribute influence=decreasing/increasing: Typically, the conditions of an increasing speed profile are necessary (AND), and the conditions of an decreasing speed profile are optional (OR). To clarify this, I would prefer an attribute for the logical link if there is more than one condition.

> ** The trainRelation attribute provides one of the enumeration values
> headOfTrain, midOfTrain and endOfTrain. It can be made required with
> next major release only. Mostly the endOfTrain value is used for
> increasing speed aspects and the headOfTrain value for decreasing speed
> aspects. In some special cases, e.g. level crossings in Germany,
> headOfTrain would be also used for increasing speed aspects.

<DOCUMENTATION HINT>

We should note, in the documentation, that all speed changes are to be understood to act independently from each other: A speed change at the front of a train does not effect the speed changes still ‘under’ the train. The lowest of all speed changes currently valid for a train is relevant.

This explains why we do not need a trainRelation="headOfTrainIfDecreasingAndEndOfTrainIfIncreasing " enumeration value… If the speed changes come closer than the length of the train (which may always be the case), the train has to maintain several speed restrictions. If the train passes an increasing speed change (valid for head) with its front but has not jet passed a lower increasing speed change (valid for end) with its end, the lower speed still has to be maintained - _in_spite_of_ the new speed change being valid for head of train. But anyway, the train has also to remember the new speed, so it cannot ignore the new speed change.

This is only another (but less obvious) variant of the very common case where a train is in one speed restriction with it’s end and already in a next speed restriction with it’s front, so never enjoying the raised speed between both restrictions… A case which all drivers of the world must handle.

So, no reason to worry about the trainRelation="headOfTrain” aspect.

<\END OF DOCUMENTATION HINT>

** The signalised attribute indicates whether the speed aspects is shown
next to text with some light signal or a panel (true) or only in the
"drivers timetable" (false).

We would need the same Boolean value for the fact whether the speed change is pre-signalised. Normally, the question of the pre-signalisation is more important (for decreasing speed changes) than the question of the in-place signalisation. The in-place signalisation is relevant for increasing speed changes, so we do need both attributes.

<DOCUMENTATION HINT>

To anticipate possible questions: A speed change is treated as not pre-signalised if there is at least one route at which a train can pass the speed change decreasing and did not pass any pre-signalisation for the speed reduction a proper braking distance before.

<\END OF DOCUMENTATION HINT>

> ** What to do with the current trainCategory attribute?

With the discussions earlier this year, nobody seams to need it. As far as I know, with the typical usage of train categories nowadays, they have nothing to do with technical (speed) matters. It would need a trainCategory with much more technical background than the categories of RailML.

I would prefer to declare it deprecated.

> ** How to define the "blocking of a track"?** Using a time restricted
> speed profile with vMax="0" in the
> speedChange elements?

From my experience, blockings have nothing to do with speed restrictions. We should consider a new ‘list of blockings’ in future, probably rather in TimeTable subschema. If we mix blockings and speed restrictions, it becomes much more difficult here…

> ** How to define an "obligational stop" where all or only certain trains
> have to stop prior going on with the same speed aspect as before?
> ** Could this be an attribute of the speedChange element?

That is an issue I would like to see solved with RailML 2.2. I would prefer either such an attribute (obligational stop of head/end of train).. The other, possible “cleaner” but also more complex solution would be to define an infrastructure element “obligationalStop” (independently from speed changes).

I think the attribute at a speedChange will be ok.

** Do we need the midOfTrain value as trainRelation in the speedChange context?

I think so (see news entry from 26.04.2012 on ‘train relation’):
Especially in very ‘rough’ infrastructure modelling, with stations being only a deterministic point at a line and trains being only a ‘point of mass’, the default understanding is that the mid of a train arrives/departs at the mid of a station. If the train arrives from a line with a different speed than that of the line it departs, consequently the speed change lies exactly at the station place and is valid for the mid of the train.

(Additionally, but less general, in Germany we have speed changes officially valid “until scheduled stopping place” (planmäßiger Halteplatz). If a stopping place is “mid of train = mid of platform”, you consequently need a speed change valid for mid of train and placed at the mid of the platform.)

Best regards,
Dirk.