Subject: Re: railML 2.3 infrastructure extension proposal line sections
Posted by                    on Thu, 18 May 2017 16:30:02 GMT
View Forum Message <> Reply to Message

Dear Torben,

> We should make the value optional so you do not need to use
> the description if you do not distinguish between path and
> station, or you do not have an exact border. ...

Yes, I know. But the point is: Why defining in railML at all? That's why
on 19.01.2017 I added the question: "I would prefer to describe exactly
what is the functional (operational?) background behind <lineSection>.
So my question would be: What is the operational background behind it?"

> Could you refer me to the earlier discussion about not
> having a station defined?

Sorry I tried but it's difficult because it seems to be spread over
years and not everything is in forum posts.

Anyway, I do not want to convince you from not-getting a reference to an
<ocp>. I myself would prefer it. Only, it is very difficult in general
and so I do only say: If we do it now, we should also define the
operational background. For instance, in railML wiki, provide a
definition of <station> or <lineSection>. To avoid that every country
uses these elements in different semantics.

> I think it's a great concept to
> optionally refer to an ocp with an @ocpRef, either on the
> track or in <NO:lineSection>@type"station"

I agree.

> The ocpRef should only go to ocp's of @operationalType"station".

I do not agree. Blocking signals should be allowed to refer to an <ocp>
of @operationalType='blockPost', for instance.

As you wrote, line-side sidings should be allowed to refer to an <ocp>
of @operationalType<>"station". They must refer to an <ocp> at all
because there may be trains entering, stopping or starting there.

> This for better orientation in the railML structure.
> Today the user needs to deduct a tracks ocp reference by other means, like
> absPos values of the crossSection of the ocp.

Yes. Why should the user need to deduct a tracks <ocp>? This again leads

to the operational background. (To make it clear again: I agree that it would be helpful to find a solution in railML. But we clarify the usage. We should avoid misunderstandings and uncontrolled usage.)

> Furthermore I refer to Christian Rahmigs comment on my forum
> posting for "ocp". Here he mentiones that we do not need to
> define which tracks are on a path and which are on a station
> as the <track>@type defines this. The values
> "connectingTrack" and "sidingTrack" are paths and the values
> "secondaryTrack",and "stationtrack" are stations. The
> problem is that, as I read the railML wiki, a main track can
> be both in a path and in a station.

I do not agree with Christian's comment in general. Yes, a main track can be both in a station and between stations. Also, <track>@type="connectingTrack" can be between two line-side switches of a crossover (German "Überleitstelle"). So "connectingTrack" may be inside and outside stations, same as with "sidings" and others.

May be Christian means that "connectingTrack" is always inside an <ocp>. If so, I probably would agree. That's why there is the term "ocp" in railML which is not the same as "station". A crossover (German "Überleitstelle") is an <ocp> in railML but not a station in Germany.

Do you see the problem? That's why we have to define what we mean with "station" if we introduce station limits in railML.

Conclusion from my side: I agree with most of your suggestions.

I would agree to assign tracks and track elements optionally to <ocp>s. If an <ocp> is a station, then the track or track element belongs to that station. If the <ocp> is no station, the track or track element is line-side.

I would agree to define station limits if we define what is a station (at least, in Wiki). Is a German Überleitstelle a station in the sense of railML or not?

With best regards,
Dirk.