
Subject: railML 2.3 infrastructure extension proposal line sections

Posted by [Torben Brand](#) on Tue, 20 Dec 2016 17:26:16 GMT

[View Forum Message](#) <> [Reply to Message](#)

Dear railML infrastructure forum,

This posting contains the discussion to an extension towards the trackGroups

In Norway we segment a line into line sections. They consist of either stations (defined from home to home signal) or paths (sections between stations; DE:freie Strecke).

There is a need to define which line section a track belongs to. The idea is to define a line section as a group of tracks.

Thus I have added the new element <NO:lineSection> under <trackGroups>.

<NO:LineSection> has the attributes: @trackRef and @type.

@Type [datatype: enumeration] is preset to "station" or "path", but allows other values, too ("other:").

Subject: Re: railML 2.3 infrastructure extension proposal line sections

Posted by [christian.rahmig](#) on Mon, 02 Jan 2017 16:29:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

Dear Torben,

Am 20.12.2016 um 18:26 schrieb Torben Brand:

> [...]

> trackGroups

> In Norway we segment a line into line sections. They consist

> of either stations (defined from home to home signal) or

> paths (sections between stations; DE:freie Strecke).

> There is a need to define which line section a track belongs

> to. The idea is to define a line section as a group of

> tracks.

> Thus I have added the new element <NO:lineSection> under

> <trackGroups>.

> <NO:LineSection> has the attributes: @trackRef and @type.

> @Type [datatype: enumeration] is preset to "station" or

> "path", but allows other values, too ("other:").

A line section (or section of line) is a structural unit that is used in other data models, e.g. RINF (see [1]), too. It can be seen as a component of a railway line and therefore represents some "meso level of detail" in the model of the railway network. If there is a need by several railML use cases (see [2]), I appreciate to integrate the line section into the railML data model, latest for version 3.

Your proposed structure

<trackGroups>

```
<NO:lineSection>
</NO:lineSection>
</trackGroups>
```

looks fine and reasonable. If the community agrees with me, I will hurry to open a Trac ticket for implementation with railML v3.

The attribute @type (enumeration with "station" and "path") makes sense as well in those countries that distinguish between station tracks and path tracks (de: "freie Strecke"). However, it should not be forgotten, that some countries like the United States of America do not know this differentiation. Therefore, the attribute @type must remain optional and allow for other values, too.

The reference to the contained tracks should not be done by an attribute, but by a sequence of child elements, similar to the current implementation of track references within the <line> element (see [3]). This will allow to reference an arbitrary number of tracks. An example may look like this:

```
<trackGroups>
  <NO:lineSection type="path">
    <trackRef ref="tr01" />
    <trackRef ref="tr02" />
  </NO:lineSection>
</trackGroups>
```

Are there any further ideas or remarks regarding the usage of the line section entity? Any comments appreciated...

- [1] <http://www.era.europa.eu/Core-Activities/Interoperability/Pages/RINF.aspx>
- [2] http://wiki.railml.org/index.php?title=Dev:Use_cases
- [3] <https://www.railml.org/files/download/schemas/2016/railML-2.3/documentation/railML.html#Link2D3>

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

Subject: Re: railML 2.3 infrastructure extension proposal line sections

Posted by on Thu, 19 Jan 2017 18:28:18 GMT

[View Forum Message](#) <> [Reply to Message](#)

Dear Torben and Christian,

- > If the community agrees with me, I will hurry
- > to open a Trac ticket for implementation with railML v3.

We also distinguish between station tracks and line tracks in other countries, and we have many railML (IS 2.x) files where one railML<track> representing a "trough line track" goes through several stations and such "switches" between line and station track from home signal to end of station and so on.

Therefore, the requirement of Torben is comprehensible.

But, I am not convinced of (nor happy with) the solution

- > <trackGroups>
- > <NO:lineSection>
- > </NO:lineSection>
- > </trackGroups>

First, we should avoid that there is a need to make such distinction (as Christian wrote: North America...).

Secondly, there may be not "hard" or "exact" definition where a station begins or ends concerning a <line>. For instance, it may start at the home signal and end at the shunting limit marker board. On a double-track line, this may be different locations depending driving direction or which track is used. So, shouldn't this distinction (if at all) be made at <track>s rather than <trackGroup>s?

Last, so far, we have avoided creating a construct like a "station" in railML, by intention! (It is no mistake that you do not find a <station> in railML so far!) So, the <lineSection> should not lead to the "accidental definition" of a <station>... which then would many un-wanted discussions (such as: Why no reference from <lineSection> to <ocp>? Is it allowed that there is more than one <ocp> in one <lineSection>?).

Rather, 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?

With best regards,
Dirk.

Am 02.01.2017 um 17:29 schrieb Christian Rahmig:

> Dear Torben,

>

> Am 20.12.2016 um 18:26 schrieb Torben Brand:

>> [...]

>> trackGroups

>> In Norway we segment a line into line sections. They consist

>> of either stations (defined from home to home signal) or

>> paths (sections between stations; DE:freie Strecke).

>> There is a need to define which line section a track belongs

>> to. The idea is to define a line section as a group of

>> tracks.

>> Thus I have added the new element <NO:lineSection> under

>> <trackGroups>.

>> <NO:LineSection> has the attributes: @trackRef and @type.

>> @Type [datatype: enumeration] is preset to "station" or

>> "path", but allows other values, too ("other:").

>

> A line section (or section of line) is a structural unit that is used in

> other data models, e.g. RINF (see [1]), too. It can be seen as a

> component of a railway line and therefore represents some "meso level of

> detail" in the model of the railway network. If there is a need by

> several railML use cases (see [2]), I appreciate to integrate the line

> section into the railML data model, latest for version 3.

>

> Your proposed structure

>

> <trackGroups>

> <NO:lineSection>

> </NO:lineSection>

> </trackGroups>

>

> looks fine and reasonable. If the community agrees with me, I will hurry

> to open a Trac ticket for implementation with railML v3.

>

> The attribute @type (enumeration with "station" and "path") makes sense

> as well in those countries that distinguish between station tracks and

> path tracks (de: "freie Strecke"). However, it should not be forgotten,

> that some countries like the United States of America do not know this

> differentiation. Therefore, the attribute @type must remain optional and

> allow for other values, too.

>

> The reference to the contained tracks should not be done by an

> attribute, but by a sequence of child elements, similar to the current

> implementation of track references within the <line> element (see [3]).

> This will allow to reference an arbitrary number of tracks. An example

> may look like this:

>

> <trackGroups>
> <NO:lineSection type="path">
> <trackRef ref="tr01" />
> <trackRef ref="tr02" />
> </NO:lineSection>
> </trackGroups>
>
> Are there any further ideas or remarks regarding the usage of the line
> section entity? Any comments appreciated...
>
> [1]
> <http://www.era.europa.eu/Core-Activities/Interoperability/Pages/RINF.aspx>
> [2] http://wiki.railml.org/index.php?title=Dev:Use_cases
> [3]
> <https://www.railml.org/files/download/schemas/2016/railML-2.3/documentation/railML.html#Link2D3>
>
>
> Best regards
> Christian
>

Subject: Re: railML 2.3 infrastructure extension proposal line sections
Posted by [Torben Brand](#) on Mon, 06 Mar 2017 08:56:18 GMT
[View Forum Message](#) <> [Reply to Message](#)

These are my answers to Christian and Dirks comments:

We also distinguish between station tracks and line tracks in other countries, and we have many railML (IS 2.x) files where one railML<track> representing a "trough line track" goes through several stations and such "switches" between line and station track from home signal to end of station and so on.

To use the suggested new element "line section" tracks would have to be segmented on path/station border. This is comparable to when the track name changes. As this also requires a segmentation of tracks in railML today.

But, I am not convinced of (nor happy with) the solution

```
> <trackGroups>  
> <NO:lineSection>  
> </NO:lineSection>  
> </trackGroups>
```

First, we should avoid that there is a need to make such distinction (as Christian wrote: North America...).

Secondly, there may be not "hard" or "exact" definition where a station begins or ends concerning a <line>. For instance, it may start at the home signal and end at the shunting limit marker board. On a double-track line, this may be different locations depending driving direction or which track is used. So, shouldn't this distinction (if at all) be made at <track>s rather than <trackGroup>s?

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. You can have different borders/border rules per track. You then need to segment the tracks on the border. You could distinguish per direction if you added a direction element to the description. As we do not need this in Norway we have not added it to our extension.

Last, so far, we have avoided creating a construct like a "station" in railML, by intention! (It is no mistake that you do not find a <station> in railML so far!) So, the <lineSection> should not lead to the "accidental definition" of a <station>... which then would many un-wanted discussions (such as: Why no reference from <lineSection> to <ocp>? Is it allowed that there is more than one <ocp> in one <lineSection>?).

Could you refer me to the earlier discussion about not having a station defined? 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" The ocpRef should only go to ocp's of @operationalType"station". 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.

In Norway a line section can have multiple ocp's on it.
<NO:lineSection>@type"station" can have either reference to one @operationalType"station","junction", "depot" or "crossover". For @operationalType"station" the line section can also have one or more @operationalType"stoppingpoint" on it.
<NO:lineSection>@type"path" can have no ocp's on it or one or multiple @operationalType"halt", "siding" or "blockSignal".

Furthermore I refer to Christian Rahmigs comment on my forum posting for "ocp". Here he mentions 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.

The documentation would have to be clear about the interface towards the existing sister element <locallyControlledArea>. For instance, in Norway a line section of type "station" is usually, but not always, a local controlled area. Likewise, local controlled areas can be found on track sections on the path. To summarize: it seems to me that there are two different definitions and usage to the two elements.

Rather, 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?

Relevant use cases:

Network statement, line description (DE:"Streckenbuch"), capacity planning. In Norway different operational rules apply pending if you are on the path or in the station area. Also values could change. For instance a @operationalType"stoppingPoint" changes value to "halt" if it's on the path.

It would help the general orientation in the railML structure by showing which ocp the hundreds of <track>@name"1" belongs to.

Another very useful effect of grouping tracks together and referencing them to an ocp would allow all objects placed on the tracks to be referred to the same ocp by proxy. Many ocsElements need to be ocp referenced. This would make things simpler. If an object by exemption needs to reference to a different ocp, that ocp could be described on the specific object.

Bob Jansen mentioned "Please note that the interlocking schema introduces TVD sections. These shouldn't conflict with line sections.» . If there are TVD sections present they correspond roughly to the line sections in Norway. This as the line section border is at a home signal and this usually has a TVD section border associated to it. So the line section suggestion adheres to TVD sections.

To round up the use case: The element lineSection is useful for us, but not essential.

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 mentions 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.

Maybe 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.
