Subject: Station borders
Posted by christian.rahmig on Fri, 24 Jan 2020 16:31:59 GMT
View Forum Message <> Reply to Message

Dear all,

in the SCTP use case working group we recently discussed about borders of stations and their representation in railML 3.x.

There are several candidates for implementation:
* a dedicated <border> element
* evaluate (linear or area) location of <operationalPoint>
* topology-based aggregation of <netElement>

But before we think about the specific implementation, let me first ask you, dear community, whether you need to distinguish between "in-station" and "outside-of-station" and - if needed - which attributes do you expect the border to have?

As usual, any feedback is highly appreciated...

Best regards
Christian

Subject: Re: Station borders
Posted by Thomas Nygreen on Fri, 21 Feb 2020 15:16:18 GMT
View Forum Message <> Reply to Message

Dear Christian,
Dear all,

An <operationalPoint> can already be placed using either <spotLocation> (on a macroscopic level), <linearLocation> (if only one track) or <areaLocation> (on one or more tracks). So I'm not sure that I see the need for more ways to do it.

Best,
Thomas

Subject: Re: Station borders
Posted by Thomas Langkamm on Mon, 24 Feb 2020 09:39:43 GMT
View Forum Message <> Reply to Message

christian.rahmig wrote on Fri, 24 January 2020 17:31
There are several candidates for implementation:
* a dedicated <border> element
* evaluate (linear or area) location of <operationalPoint>

* topology-based aggregation of <netElement>

I would suggest to implement option (3), topology-based aggregation. A "station" is at its core a topological element, and I would assume that it is a typical netElement in a macroscopic topology. As elements from more detailled topologies must be consistent with the top-level topology, that is, a microscopic netElement (representing a track, or part of it) must be either part of the station or not, but it's not allowed that only a part of that netElement is in the station and the remaining part is not. So we'll have to make sure that the netElement ends at the station boundary anyway.

For (1), I don't like borders. It has the potential of contradicting definitions, if you add a single track and forget to define the border then suddenly the whole track plan is the station (or at least a lot more than intended). Even worse things can happen if we mix up inside or outside. My favored solution (3) leaves no possiblity for a snafu, either a netElement is assigned to the station netElement or not.

(2) is just (1) in disguise, I think. In the end we define a border based on a positioning system.

An editor could of course still use a border element to visualize the boundaries of the station. But I'd prefer it to be a derived object, not a persistant one.

Subject: Re: Station borders
Posted by christian.rahmig on Wed, 26 Feb 2020 10:39:45 GMT
View Forum Message <> Reply to Message

Thomas Nygreen wrote on Fri, 21 February 2020 16:16Dear Christian,
Dear all,

An <operationalPoint> can already be placed using either <spotLocation> (on a macroscopic level), <linearLocation> (if only one track) or <areaLocation> (on one or more tracks). So I'm not sure that I see the need for more ways to do it.

Best,
Thomas
Dear Thomas,

you are right, but in fact, all three options for modelling station borders are already possible. Therefore, it is important to find the best practice, in order to limit flexibility of the model and thus increase compatibility of railML interfaces. So, option 2 is your preferred solution?

Best regards
Christian

Subject: Re: Station borders
Posted by Thomas Nygreen on Wed, 04 Mar 2020 15:37:39 GMT
View Forum Message <> Reply to Message

Dear Thomas,
Dear Christian,

Christian is correct that all three are already possible, although I interpreted (3) as something else than the currently available topology aggregation, and more like <propEquipment> in railML2. I fully agree that we need to limit the number of ways to model the same infrastructure.

To me, topology and functional infrastructure are separate. An <operationalPoint> of type "stoppingPoint" does not change the topology, and therefore it is not natural to me that it should have its own <netElement>. We could require it, but it would have to be through a semantic constraint, and that constraint would have to be worded to cope with differing station border definitions, including networks that do not have them.

Trouble with 2 and 3: As Dirk has pointed out a couple of times, station borders may be direction-dependent.[1][2]

What is the purpose of defining station borders in the railML infrastructure? How will the information be used? Maybe the candidates serve different, (mainly) disjunct purposes?

One possibility is to narrow down the use of each of the existing possibilities, to try to make them fully disjunct, e.g.:

(1) Borders mark a point in the infrastructure that is relevant for operations, and should be used where needed by operational rules but not to model the station area.

Pros: Allows direction-dependent borders. Avoids graph traversal to determine station area. A missing border does not affect station area.

(2) AreaLocation should be used to define the station area.

Pros: Builds upon topology but does not need to be 1:1. Defined directly as a child of <operationalPoint> (as opposed to multiple borders). <operationalPoint>s inside another <operationalPoint> clearly defined.

Question: Should SpotLocation also be accepted, when there is no need to define an area? Should we rule out LinearLocation for <operationalPoint>?

(3) NetElements model the underlying topology and should not be used to model station areas (or other areas).

Pros: Purpose separated from AreaLocation. Does not require multiple meso aggregation levels for <operationalPoint>s inside another <operationalPoint> or splitting <netElement>s at stopping points.

[1] https://www.railml.org/forum/index.php?t=msg&th=482& goto=1476#msg_1476
[2] https://www.railml.org/forum/index.php?t=msg&goto=1990#m sg_1990

Best regards,
Thomas Nygreen - Common schema coordinator, railML.org

Subject: Re: Station borders
Posted by on Mon, 09 Mar 2020 21:01:16 GMT
View Forum Message <> Reply to Message

Dear Thomases and Christian,

> As Dirk has pointed out a couple of times, station borders may be direction-dependent.[1][2]

Thank you, that's exactly what I wanted to mention right now... ;-)

Not only this: There are also several different definitions of station borders at the same time. (In Germany: interlocking, operational and traffic definitions are often different. Several operational stations can form one traffic station, as Berlin Hbf oben + unten + S-Bahn. One operational stations can also consist of several traffic stations, as Dresden Hbf / Mitte / Neustadt. The interlocking borders are wider that the operational borders typically when stations are remote-controlled. But also the other way round, one large operational station can be split into several interlocking areas, as it always was the case for Leipzig Hbf (Prussian / Saxon signal boxes strictly separated) and even nowadays still is (several ESTWs).

All together: When defining station borders, always allow a definition which kind of border type is meant and allow definitions depending on direction.

Dirk.

Subject: Re: [railML3] Station borders
Posted by christian.rahmig on Fri, 13 Mar 2020 11:23:49 GMT
View Forum Message <> Reply to Message

Dear Dirk,

always allow a definition which kind of border type is meant and allow definitions depending on direction.

Thank you for your feedback. Following your requirement for modelling station borders option 1 (a dedicated <border> element) seems to be the only solution. Are there any other opinions from the community?

I also appreciate the idea of Thomas to distinguish between different purposes. In particular,
* <border> elements can be used to locate a (direction dependent and type specific) station border
* <operationalPoint><areaLocation> may be used to define the (static) station area.

What does the community think about this modelling approach for railML 3.x? Any comments are very much welcome...

Best regards
Christian

Subject: Re: [railML3] Station borders
Posted by Fabrizio Cosso on Fri, 20 Mar 2020 17:00:51 GMT
View Forum Message <> Reply to Message

Dear all,
I would like to add my point of view to the discussion:
- "whether you need to distinguish between "in-station" and "outside-of-station": yes, we did. Several applications, one of them traffic monitoring system, need to know what is in and what is out.
- regarding how to model borders, I'm not sure we need to add a specific functional infrastructure element: is a border a physical element - e.g. a signal? There are several kind of signals already available.
What about using belongsToParent in order to determine what is within an operational point and what is out?

Thanks

BR

Fabrizio