
Subject: [railML 3.1] border types

Posted by [christian.rahmig](#) on Thu, 24 May 2018 16:52:20 GMT

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

Dear all,

the functional infrastructure element <border> shall be used to describe a border point between different "zones", e.g. between railway networks operated by different infrastructure managers.

Further border types are:

- * area (quite general)
- * country (border between different countries)
- * state (border between different states within one country)
- * station (border between railway line and station)
- * tariff (border between different tariff zones)

My questions to all of you:

- * Is this list of border types complete?
- * Which border types do you need from your application perspective?
- * Which other parameters are essential for you to describe the border point?

Thank you for your contributions!

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 3.1] border types

Posted by [Thomas Nygreen JBD](#) on Tue, 29 May 2018 16:45:13 GMT

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

In Norway we discussed just a week or two ago if <border>s were suitable for specifying shunting areas etc. in stations. Would this kind of use be in line with what the element is intended for? Two questions we had was how to group borders together to actually form an area, and how to specify what kind of area it is. The former can be solved by using a common name for all borders of the same area, and the latter by using type="other:...", but creating a way to group borders together by IDREF seems preferable.

This is also one of the elements where it is difficult to understand the semantics of the dir attribute. Can a border exist in only one direction?

Subject: Re: [railML 3.1] border types
Posted by [Joerg von Lingen](#) on Sat, 02 Jun 2018 15:21:45 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

if you have something in mind which is temporarily in use and activated/deactivated by an interlocking then it shall be defined in interlocking schema. There we have thought about "RestrictedAreas" especially for:

- Working zones
- local shunting areas
- ETCS shunting areas
- permission zones

Dr.-Ing. Jörg von Lingen - Interlocking scheme coordinator
railML.org (Registry of Associations: VR 5750)
Phone Coordinator: +49 351 87759 40; railML.org: +49 351 47582911
Altplauen 19h; 01187 Dresden; Germany www.railml.org
On 29.05.2018 18:45, Thomas Nygreen wrote:

- > In Norway we discussed just a week or two ago if <border>s
 - > were suitable for specifying shunting areas etc. in
 - > stations. Would this kind of use be in line with what the
 - > element is intended for? Two questions we had was how to
 - > group borders together to actually form an area, and how to
 - > specify what kind of area it is. The former can be solved by
 - > using a common name for all borders of the same area, and
 - > the latter by using type="other:...", but creating a way to
 - > group borders together by IDREF seems preferable.
 - >
 - > This is also one of the elements where it is difficult to
 - > understand the semantics of the dir attribute. Can a border
 - > exist in only one direction?
-

Subject: Re: [railML 3.1] border types
Posted by [christian.rahmig](#) on Mon, 04 Jun 2018 13:25:53 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear Thomas,

- Am 29.05.2018 um 18:45 schrieb Thomas Nygreen:
- > In Norway we discussed just a week or two ago if <border>s
 - > were suitable for specifying shunting areas etc. in
 - > stations. Would this kind of use be in line with what the
 - > element is intended for? Two questions we had was how to
 - > group borders together to actually form an area, and how to
 - > specify what kind of area it is. The former can be solved by

- > using a common name for all borders of the same area, and
- > the latter by using type="other:...", but creating a way to
- > group borders together by IDREF seems preferable.

the situation that you describe here, is better being solved with a different implementation: Instead of using border elements, I suggest to define an <OperationalPoint> and to locate this operational point as an area covering all the affected tracks. Further, this <OperationalPoint> can be specified with an attribute <propOperational>@operationalType="shuntingYard". Finally, the interlocking element may reference this operational point.

So, to conclude: I think that grouping of borders is not the best solution here. Borders shall be used where there is an explicit point (e.g. on the track) where e.g. the ownership changes (without knowing where else it will change too).

- > This is also one of the elements where it is difficult to
- > understand the semantics of the dir attribute. Can a border
- > exist in only one direction?

Interesting question. Usually, a border point should be always valid in both directions. But let's think of a station: the border between station and "free line" may differ depending on the driving direction: when entering the station, the entry signal mark a border point. When leaving the station, the last switch may be seen as border point. However, I am not the expert in these details and therefore I am happy about any further comment from the community w.r.t. this issue.

Thank you very much and 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 3.1] border types
Posted by [Thomas Nygreen JBD](#) on Thu, 07 Jun 2018 14:56:25 GMT
[View Forum Message](#) <> [Reply to Message](#)

christian.rahmig wrote on Mon, 04 June 2018 15:25
Am 29.05.2018 um 18:45 schrieb Thomas Nygreen:
> In Norway we discussed just a week or two ago if <border>s
> were suitable for specifying shunting areas etc. in
> stations. Would this kind of use be in line with what the

- > element is intended for? Two questions we had was how to
- > group borders together to actually form an area, and how to
- > specify what kind of area it is. The former can be solved by
- > using a common name for all borders of the same area, and
- > the latter by using type="other:...", but creating a way to
- > group borders together by IDREF seems preferable.

the situation that you describe here, is better being solved with a different implementation: Instead of using border elements, I suggest to define an <OperationalPoint> and to locate this operational point as an area covering all the affected tracks. Further, this <OperationalPoint> can be specified with an attribute <propOperational>@operationalType="shuntingYard". Finally, the interlocking element may reference this operational point.

A shunting yard is something else than what I am trying to describe. What we would like to do is to define areas within stations for different interlocking purposes. So Jörg is correct. Two common uses would be for defining areas that can be released from the interlocking for manual operation (probably fits locallyControlledArea in railML 2.x, except that it requires tracks to be split at the borders) or areas that are electrically separated in the signalling system, such that one can be shut down for maintenance without shutting down the whole station. It is too long since I looked at the railML 3 specs to remember if there are other groupings that work better.

christian.rahmig wrote on Mon, 04 June 2018 15:25

So, to conclude: I think that grouping of borders is not the best solution here. Borders shall be used where there is an explicit point (e.g. on the track) where e.g. the ownership changes (without knowing where else it will change too).

I agree that grouping borders is not the best solution. It might be that my mind is too occupied at the moment with solving our needs using the elements that are already in railML 2.x.

Best regards,
Thomas Nygreen

Subject: Re: [railML 3.1] border types
Posted by on Tue, 21 Aug 2018 16:10:43 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear Christian,

currently, we only use <ocp>.<propService>.tariffpoint in an unspecified way but no <border> element.

However, there may be some demand to model areas with borders for statistic purposes, to using

this in railML is planned for the future with a lower priority from our side.

But if we would do so, we would use it in a much more generic way. We would not be able to link areas and borders with countries, states or tariff zones. There is too much arbitrariness concerning this in practice. So we would only need "areas" of different "levels" which meet each other at "borders". In each level, areas cannot overlap but we could have unlimited levels. Whether a level is a country, state or tariff zone we do not know. There are much more levels imaginable but I think we cannot describe them in railML.

Some instances:

There are more administrative border types (levels) than "countries" and "states". In many countries, there are districts or counties. In Saxony (and some other German "free states"), there are "[Regierungsamts]Bezirke". There may be sub-authorities (Bestellerorganisationen) of federal authorities (Aufgabenträger) whose borders are not identical to the districts in spite of, by law in Germany, the districts (Kreise und kreisfreie Städte) are responsible for the local public transit... This all leads to a nearly inscrutable nightmare of borders.

There is a border between Germany and Czech Republic. This border crosses the railway line Plauen - Cheb _several_ times. But depending on the level of administration, not all of these border-crossings are the same: For operational purposes like Zuglaufmeldungen, there is (a.f.a.i.k.) only one border-crossing. For legal purposes in a strict sense (police, prosecution), surely all border-crossings count. There may be many graduations between them: Who maintains tracks and ballast in the short Czech sections north-west of Bad Brambach? Who pays property taxes for that (if anyone)? ;-)

So, to come back to your questions:

- > * Which border types do you need from your application perspective?
- > * Which other parameters are essential for you to describe the border point?
- > * Is this list of border types complete?

More than. In my opinion, for railML, it's already too many because as my examples should show, you can possibly not capture all aspects of borders. So, I would prefer a much more simple and generic model.

With best regards,
Dirk.

Subject: Re: [railML 3.1] border types
Posted by [Joerg von Lingen](#) on Sun, 26 Aug 2018 06:57:01 GMT
[View Forum Message](#) <> [Reply to Message](#)

In interlocking schema of railML3 we will have two different ways to reflect these needs:

1) RestrictedArea: It can be defined as LocalOperationArea with elements inside, elements as border and elements released for local operation.

2) ElementGroup: For an interlocking (SignalBox) a group of element can be defined which are operated together like setting a group of signals to stop aspect. Although the group type "catenary" just not yet exists it can be easily added.

Best regards,
Joerg v. Lingen

Rollingstock Coordinator

On 07/06/2018 16:56, Thomas Nygreen wrote:

```
> christian.rahmig wrote on Mon, 04 June 2018 15:25
>> Am 29.05.2018 um 18:45 schrieb Thomas Nygreen:
>>> In Norway we discussed just a week or two ago if
>>> <border>s
>>> were suitable for specifying shunting areas etc.
>>> in
>>> stations. Would this kind of use be in line with
>>> what the
>>> element is intended for? Two questions we had was
>>> how to
>>> group borders together to actually form an area,
>>> and how to
>>> specify what kind of area it is. The former can be
>>> solved by
>>> using a common name for all borders of the same
>>> area, and
>>> the latter by using type="other:...", but creating
>>> a way to
>>> group borders together by IDREF seems preferable.
>>
>> the situation that you describe here, is better being
>> solved with a different implementation: Instead of using border
>> elements, I suggest to define an <OperationalPoint> and to locate this
>> operational point as an area covering all the affected tracks. Further, this
>> <OperationalPoint> can be specified with an attribute
>> <propOperational>@operationalType="shuntingYard".
>> Finally, the interlocking element may reference this operational
>> point.
>
>
> A shunting yard is something else than what I am trying to
> describe. What we would like to do is to define areas within
> stations for different interlocking purposes. So Jörg is
> correct. Two common uses would be for defining areas that
> can be released from the interlocking for manual operation
> (probably fits locallyControlledArea in railML 2.x, except
```

> that it requires tracks to be split at the borders) or areas
> that are electrically separated in the signalling system,
> such that one can be shut down for maintenance without
> shutting down the whole station. It is too long since I
> looked at the railML 3 specs to remember if there are other
> groupings that work better.
>
> christian.rahmig wrote on Mon, 04 June 2018 15:25
>> So, to conclude: I think that grouping of borders is not
>> the best solution here. Borders shall be used where there is an
>> explicit point (e.g. on the track) where e.g. the ownership changes
>> (without knowing where else it will change too).
>
>
> I agree that grouping borders is not the best solution. It
> might be that my mind is too occupied at the moment with
> solving our needs using the elements that are already in
> railML 2.x.
>
> Best regards, Thomas Nygreen

Subject: Re: [railML 3.1] border types
Posted by [Torben Brand](#) on Fri, 09 Apr 2021 07:57:33 GMT
[View Forum Message](#) <> [Reply to Message](#)

I suggest adding the border type "project" to the list.

This also for railML 2.5

A project border defines the outer limits of a project area. The project area can be defined as the area where either changes are made to the infrastructure based on a specific project or an area that is relevant for a specific project.

In its simplest form the project area can be defined only by its borders. In a more precise manner, a project border can form an area (see also posting for an "area" element extension under railML2.5 <https://www.railml.org/forum/index.php?t=msg&th=804&start=0> , already available in railML3.1).

In both these cases the project can only be described through the name and/or description attributes. With a reference by the proposed project metadata element (See posting in misc <https://www.railml.org/forum/index.php?t=msg&th=805&start=0>) the project can be defined more precise.

For more details see document "railML2.4nor Infrastructure Documentation"
(<https://www.jernbanedirektoratet.no/railML>), version 1.4, 17.12.2020, point 4.13.

Subject: Re: [railML 3.1] border types

Posted by [christian.rahmig](#) on Fri, 09 Apr 2021 13:07:21 GMT

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

Dear Torben,

for railML 2.5 it is proposed to have a (generic) area element and to define the area borders using references from this area element to <border> elements of type "area". This means, that a border of a project area is modelled in the same manner like a border of a working area.

So, the question is: Is it necessary to distinguish between different area type borders or is it sufficient to distinguish between different types of areas inside an <area> element? What is the added value of writing <border type="project"> instead of <border type="area"> when the referenced object (<area> element) is the same?

Any opinions from the community are highly appreciated.

Best regards
Christian

Subject: Re: [railML 3.1] border types

Posted by [Torben Brand](#) on Wed, 19 May 2021 05:52:06 GMT

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

When using the (in railML2.5 suggested new in railML3 existing) element <area> with type="project" this would be redundant. but as the use of border@type is mandatory, we suggest to use the existing type="area" to reference to the area, when in use.

But you can also have the need to define stand-alone project borders without an area (when an area is open or not defined clearly). So we still suggest to introduce the new value border@type="project".

Subject: Re: [railML 3.1] border types

Posted by [christian.rahmig](#) on Fri, 21 May 2021 14:14:37 GMT

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

Dear Torben, dear all,

thank you for your feedback. The modification has been implemented in railML 2.5 schema as described in Trac ticket #418 [1].

[1] <https://trac.railml.org/ticket/418>

Best regards
Christian
