
Subject: Platform @belongsToParent and <ownsPlatformEdge>
Posted by [Thomas Nygreen JBD](#) on Fri, 08 Feb 2019 19:46:42 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear all,

Two or more <platform> elements can be grouped either by each part referencing the "main" platform using the belongsToParent attribute, or by the "main" platform listing all the parts using the ownsPlatformEdge element. Or both. Are they both needed? To me they seem to have the same function, just differing in whether the reference is upwards or downwards. It will be easier for consuming systems if there is just one.

Also, if both reference methods are used, but not identically, how should the reading system interpret this? E.g. <platform id="A"><ownsPlatformEdge @ref="B"/></platform> and <platform id="B" belongsToParent="C"/> and <platform id="C"/>. Are all three platforms then grouped as one, and is there any semantic difference between the two references?

Several other functional infrastructure types have the belongsToParent attribute, but I cannot find one that has a similar owns* construct for referencing other elements of the same type. So I suggest removing the element ownsPlatformEdge.

Subject: Re: Platform @belongsToParent and <ownsPlatformEdge>
Posted by [christian.rahmig](#) on Mon, 11 Feb 2019 14:29:17 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear Thomas, dear all,

Am 08.02.2019 um 20:46 schrieb Thomas Nygreen:

- > [...] Two or more <platform> elements can be grouped either by
- > each part referencing the "main" platform using the
- > belongsToParent attribute, or by the "main" platform listing
- > all the parts using the ownsPlatformEdge element. Or both.
- > Are they both needed? [...]

This "redundancy" issue results from the past where we had two separate elements: <platform> and <platformEdge>. A <platform> could refer to all <platformEdge> objects that represent the interface between a train (on the track) and the platform. This is a reference from an object to an object part.

The hierarchical reference between objects of the same type (platform to platform or platform edge to platform edge) shall be done using the attribute @belongsToParent. One example for this type of referencing: grouping together platform edges with different height to a long platform edge.

- > [...] So I suggest removing the element

> ownsPlatformEdge.

Well, regarding the redundancy, I agree with you. But then we need to think about alternatives of how to distinguish between platforms and platform edges.

Alternative 1:

Introduce a boolean flag in <platform>: @isPlatformEdge

Alternative 2:

Re-Introduce the functional infrastructure element <platformEdge>

So, what do you prefer or do you have a third solution proposal?

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: Platform @belongsToParent and <ownsPlatformEdge>
Posted by [Thomas Nygreen JBD](#) on Mon, 11 Feb 2019 17:11:56 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear Christian,

christian.rahmig wrote on Mon, 11 February 2019 15:29

The hierarchical reference between objects of the same type (platform to platform or platform edge to platform edge) shall be done using the attribute @belongsToParent. One example for this type of referencing: grouping together platform edges with different height to a long platform edge.

This assumes that the hierarchy is given, but the current model allows doing this in multiple ways. Let's say we have a symmetrical platform between tracks 1 and 2; 200 meters of it is 5 meters wide, 0.76 meters high and made of concrete, while the remaining 150 meters is older with gravel surface, tapering off to 1 meters wide, 0.55 meters high. The hierarchy can be:

Either:

Platform 1/2, length=350m

|- PlatformEdge 1, length=350m

| |- PlatformEdge 1a, length=200m, width=? (5m or 2.5m?), height=0.76m

| \- PlatformEdge 1b, length=150m, width=? (somewhere between 0.5m and 5m), height=0.55m

\- PlatformEdge 2, length=350m

|- PlatformEdge 2a, length=200m, width=? (5m or 2.5m?), height=0.76m

\- PlatformEdge 2b, length=150m, width=? (somewhere between 0.5m and 5m), height=0.55m

Or:

Platform 1/2, length=350m

| - Platform 1a/2a, length=200m

| | - PlatformEdge 1a, length=200m, width=? (5m or 2.5m?), height=0.76m

| | - PlatformEdge 2a, length=200m, width=? (5m or 2.5m?), height=0.76m

\- Platform 1b/2b, length=150m

\- PlatformEdge 1b, length=150m, width=? (somewhere between 0.5m and 5m), height=0.55m

\- PlatformEdge 2b, length=150m, width=? (somewhere between 0.5m and 5m), height=0.55m

Or:

Platform 1/2, length=350m

| - PlatformEdge 1a, length=200m, width=? (5m or 2.5m?), height=0.76m

| - PlatformEdge 1b, length=150m, width=? (somewhere between 0.5m and 5m), height=0.55m

| - PlatformEdge 2a, length=200m, width=? (5m or 2.5m?), height=0.76m

\- PlatformEdge 2b, length=150m, width=? (somewhere between 0.5m and 5m), height=0.55m

Or even just:

PlatformEdge 1a, length=200m, width=? (5m or 2.5m?), height=0.76m

+ PlatformEdge 1b, length=150m, width=? (somewhere between 0.5m and 5m), height=0.55m

+ PlatformEdge 2a, length=200m, width=? (5m or 2.5m?), height=0.76m

+ PlatformEdge 2b, length=150m, width=? (somewhere between 0.5m and 5m), height=0.55m

Without an independent grouping element, but instead just references to one platform edge from the other platform edges.

Also, in all these examples, the lower levels of the hierarchy can be skipped. There are also multiple ways to assign <linearLocation>s: either all elements have them (and platforms have one for each track), or only the lower levels have them, or something in between.

christian.rahmig wrote on Mon, 11 February 2019 15:29

we need to think about alternatives of how to distinguish between platforms and platform edges.

Alternative 1:

Introduce a boolean flag in <platform>: @isPlatformEdge

Alternative 2:

Re-Introduce the functional infrastructure element <platformEdge>

If we need or want a strict hierarchy, considering platforms and platform edges as separate entities, then I think it would make the most sense to have them as separate elements. But do we really need to? I would rather keep using one type and leave the hierarchy up to the writing system. One thing we could do to make life a bit easier for the reading systems is to introduce one semantic constraint: If attributes describing a parent <platform> such as length and height are given, they must include other <platform>s that @belongsToParent.

I see that, apart from <ownsPlatformEdge>, there is only one other occurrence of "platformEdge" in the XSD, and that is <stoppingPlace>@platformEdgeRef. Since platformEdge does no longer

exist, it is probably better to rename this to @platformRef. The documentation can still specify that the reference can be to a platform edge, in the form of a <platform> element.

Subject: Re: Platform @belongsToParent and <ownsPlatformEdge>

Posted by [christian.rahmig](#) on Mon, 18 Feb 2019 15:12:07 GMT

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

Dear Thomas,

Am 11.02.2019 um 18:11 schrieb Thomas Nygreen:

> [...]
> This assumes that the hierarchy is given, but the current
> model allows doing this in multiple ways. Let's say we have
> a symmetrical platform between tracks 1 and 2; 200 meters of
> it is 5 meters wide, 0.76 meters high and made of concrete,
> while the remaining 150 meters is older with gravel surface,
> tapering off to 1 meters wide, 0.55 meters high. The
> hierarchy can be:
>
> Either:
> Platform 1/2, length=350m
> |- PlatformEdge 1, length=350m
> | |- PlatformEdge 1a, length=200m, width=? (5m or 2.5m?),
> height=0.76m
> | \- PlatformEdge 1b, length=150m, width=? (somewhere
> between 0.5m and 5m), height=0.55m
> \- PlatformEdge 2, length=350m
> |- PlatformEdge 2a, length=200m, width=? (5m or 2.5m?),
> height=0.76m
> \- PlatformEdge 2b, length=150m, width=? (somewhere
> between 0.5m and 5m), height=0.55m
>
> Or:
> Platform 1/2, length=350m
> |- Platform 1a/2a, length=200m
> | |- PlatformEdge 1a, length=200m, width=? (5m or 2.5m?),
> height=0.76m
> | |- PlatformEdge 2a, length=200m, width=? (5m or 2.5m?),
> height=0.76m
> \- Platform 1b/2b, length=150m
> \- PlatformEdge 1b, length=150m, width=? (somewhere
> between 0.5m and 5m), height=0.55m
> \- PlatformEdge 2b, length=150m, width=? (somewhere
> between 0.5m and 5m), height=0.55m
>
> Or:
> Platform 1/2, length=350m

- > |- PlatformEdge 1a, length=200m, width=? (5m or 2.5m?),
- > height=0.76m
- > |- PlatformEdge 1b, length=150m, width=? (somewhere between
- > 0.5m and 5m), height=0.55m
- > |- PlatformEdge 2a, length=200m, width=? (5m or 2.5m?),
- > height=0.76m
- > \- PlatformEdge 2b, length=150m, width=? (somewhere between
- > 0.5m and 5m), height=0.55m

Option 1 is my favourite.

In case we want to remove the attribute @ownsPlatformEdge, the related platform edges have to reference their parent platform via @belongsToParent.

- >
- > Also, in all these examples, the lower levels of the
- > hierarchy can be skipped. There are also multiple ways to
- > assign <linearLocation>s: either all elements have them (and
- > platforms have one for each track), or only the lower levels
- > have them, or something in between.

I would expect platform edges to be modelled / located as linear elements, while a platform is more likely seen as an areal object. For operational purposes it is necessary to have linear coordinates (mileage values) for the platform edges.

- >
- > christian.rahmig wrote on Mon, 11 February 2019 15:29
- >> we need to think about alternatives of how to
- >> distinguish between
- >> platforms and platform edges.
- >>
- >> Alternative 1:
- >> Introduce a boolean flag in <platform>: @isPlatformEdge
- >>
- >> Alternative 2:
- >> Re-Introduce the functional infrastructure element
- >> <platformEdge>

Alternative 3:

<platformEdge> as child element of <platform>

In that case, a <platformEdge> cannot exist alone (without a platform).

- >
- > If we need or want a strict hierarchy, considering platforms
- > and platform edges as separate entities, then I think it
- > would make the most sense to have them as separate elements.
- > But do we really need to? I would rather keep using one type

- > and leave the hierarchy up to the writing system. One thing
- > we could do to make life a bit easier for the reading
- > systems is to introduce one semantic constraint: If
- > attributes describing a parent <platform> such as length and
- > height are given, they must include other <platform>s that
- > @belongsToParent.

I don't understand how you mean this, sorry.

- >
- > I see that, apart from <ownsPlatformEdge>, there is only one
- > other occurrence of "platformEdge" in the XSD, and that is
- > <stoppingPlace>@platformEdgeRef. Since platformEdge does no
- > longer exist, it is probably better to rename this to
- > @platformRef. The documentation can still specify that the
- > reference can be to a platform edge, in the form of a
- > <platform> element.

Agreed. A <stoppingPlace> can refer to the platform edge it belongs to also via @platformRef.

How about the idea from above: adding a boolean flag @isPlatformEdge to distinguish between a platform and a platform edge?

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: Platform @belongsToParent and <ownsPlatformEdge>
Posted by [Thomas Nygreen JBD](#) on Mon, 18 Feb 2019 19:55:45 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear Christian,

christian.rahmig wrote on Mon, 18 February 2019 16:12
> Option 1 is my favourite.

I assumed so, but the other options are also valid (although the last one is weird).

- > In case we want to remove the attribute @ownsPlatformEdge, the related
- > platform edges have to reference their parent platform via @belongsToParent.

Yes. Nothing strange about that.

```
>> Also, in all these examples, the lower levels of the
>> hierarchy can be skipped. There are also multiple ways to
>> assign <linearLocation>s: either all elements have them (and
>> platforms have one for each track), or only the lower levels
>> have them, or something in between.
>
> I would expect platform edges to be modelled / located as linear
> elements, while a platform is more likely seen as an areal object. For
> operational purposes it is necessary to have linear coordinates (mileage
> values) for the platform edges.
```

Or you can have no platform edges at all, but linear coordinates on two different tracks on the same platform. I.e. with the lower levels of the hierarchy skipped. <stoppingPlace>s on both tracks can reference this same platform.

```
>> christian.rahmig wrote on Mon, 11 February 2019 15:29
>>> we need to think about alternatives of how to
>>> distinguish between
>>> platforms and platform edges.
>>>
>>> Alternative 1:
>>> Introduce a boolean flag in <platform>: @isPlatformEdge
>>>
>>> Alternative 2:
>>> Re-Introduce the functional infrastructure element
>>> <platformEdge>
>
> Alternative 3:
> <platformEdge> as child element of <platform>
> In that case, a <platformEdge> cannot exist alone (without a platform).
```

Alternative 4: no explicit platform edge concept.

If the user systems use this separation between platforms and platform edges, they can be separated by, for instance, regarding <platform>s with <areaLocation>s as platforms and <platform>s with <linearLocation>s as platformEdges. But if this separation is a strong requirement, I strongly prefer the @isPlatformEdge attribute (as optional).

```
>> If we need or want a strict hierarchy, considering platforms
>> and platform edges as separate entities, then I think it
>> would make the most sense to have them as separate elements.
>> But do we really need to? I would rather keep using one type
>> and leave the hierarchy up to the writing system. One thing
>> we could do to make life a bit easier for the reading
>> systems is to introduce one semantic constraint: If
>> attributes describing a parent <platform> such as length and
```


>> height are given, they must include other <platform>s that
>> @belongsToParent.
>
> I don't understand how you mean this, sorry.

In the first half, I was just saying what I have reiterated above, that I do not see a strong need to separate the concepts platform and platform edge.

The second half was an attempt to rule out the weird option from my list of hierarchies. (The one where the platform edges just reference each other and not a parent platform.) My example is probably too weird, but a slightly more realistic one is this: Let's say we have a platform of 250 metres physically divided in two parts A (150 m) and B (100 m) (e.g. by a level crossing or change in height or surface).

```
=====|===== track
 /-----|-----\
 |      A      || B      | platform
```

Maybe we regard A as the main part of the platform, and would use @belongsToParent on B to reference A. But this would be confusing if A@length="150" and B@length="100" and B@belongsToParent="A". What is then the total length of the platform? What I suggested, in more plain text, was to require that whatever B@belongsToParent references must be something that B is a part of.

[b]Summing up:[b] I do not need a way to separate platforms from platform edges (alternative 4). But if someone else needs a way to do this explicitly, I prefer to add an optional attribute (alternative 1).

Subject: Re: Platform @belongsToParent and <ownsPlatformEdge>
Posted by [christian.rahmig](#) on Mon, 08 Apr 2019 17:56:08 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear Thomas,

thank you for your input and for summing up the topic:

Am 18.02.2019 um 20:55 schrieb Thomas Nygreen:
> [...]

> [b]Summing up:[b] I do not need a way to separate platforms
> from platform edges (alternative 4). But if someone else
> needs a way to do this explicitly, I prefer to add an
> optional attribute (alternative 1).

From my perspective there are two aspects to get feedback on:

1) Is there a need to distinguish between platforms and platform edges.
I think, there is. For example, in Czech Republic the platform

(nastupiste) has a different identifier/designator than the connected platform edges (kolej). Other examples around Europe and the world?

2) If there is a need for both, platforms and platform edges, is it enough to use the same element <platform> and just have a boolean attribute @isPlatformEdge to distinguish between them or do we need further attributes that are specific for only one of the types?

As usual, any contribution is very much appreciated...

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: Platform @belongsToParent and <ownsPlatformEdge>
Posted by [christian.rahmig](#) on Wed, 18 Nov 2020 10:31:47 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear all,

discussions in the railML use case working groups have shown that there is a need for having two separate elements modelling platforms and platform edges. Consequently, I created a new Trac ticket #438 for this issue [1].

[1] <https://trac.railml.org/ticket/438>

Best regards
Christian
