Subject: Haltetafel / stop post
Posted by Susanne Wunsch railML on Fri, 04 Nov 2011 21:42:09 GMT
View Forum Message <> Reply to Message

Hello,

prior to our meeting next monday, let me toss some litte ideas around.

Firstly, there is a wish for a "stop post" element/attribute in the infrastructure branch:

> A platform at a station may have more than one stop posts for different kinds of trains, mostly depending on its length, sometimes depending on other properties of the rollingstock, e.g. axle count, verbal "rollingstock kind" or wagon count.

It should be possible to refer to such "stop posts" from within a trainParts ocpTT to mark a special stop position.

That's quite a bit confusing. Different train parts can't stop at the same "stop post" - they run one behind the other. Certainly it marks the stop position of the head of a "whole train" (railML semantic: "operational train"). But the "train" doesn't provide stops (railML semantic: "ocpTT").

> I would prefer creating an element for enabling coordinates and above mentioned stopping constraints.
>
> Where to put the element in the infrastructure branch? Could it be linked to a "crossSection" (for easier referencing from within the timetable branch)?
>
> Could we reuse the "ocpRef" attribute in "ocpTT" for this more detailed ocp stop position?

I am very interested in any hints, questions and comments.

Maybe seeing you soon in Karlsruhe...
Susanne

--
Susanne Wunsch
Schema Coordinator: railML.common

Subject: Re: Haltetafel / stop post
Posted by Christian Rahmig on Mon, 02 Apr 2012 20:41:37 GMT

View Forum Message <> Reply to Message

Hello Susanne,

although any railML user didn't reply to your questions, it's obvious
that we need a stop post element for specifying the positions on the
platforms where the train may stop.

> A platform at a station may have more than one stop posts for
> different kinds of trains, mostly depending on its length, sometimes
> depending on other properties of the rollingstock, e.g. axle count,
> verbal "rollingstock kind" or wagon count.
>
> It should be possible to refer to such "stop posts" from within a
> trainParts ocpTT to mark a special stop position.
>
> That's quite a bit confusing. Different train parts can't stop at the
> same "stop post" - they run one behind the other. Certainly it marks the
> stop position of the head of a "whole train" (railML semantic:
> "operational train"). But the "train" doesn't provide stops (railML
> semantic: "ocpTT").

These details need to be discussed within the timetable schema forum.

> I would prefer creating an element for enabling coordinates and
> above mentioned stopping constraints.
>
> Where to put the element in the infrastructure branch? Could it be
> linked to a "crossSection" (for easier referencing from within the
> timetable branch)?
>
> Could we reuse the "ocpRef" attribute in "ocpTT" for this more
> detailed ocp stop position?

The stop post itself is a physical element, which is a sign right next to the track. Therefore I would not use the crossSection element for specifying stop posts. Instead I would put the stop post element inside the ocsElements container. The even more difficult question is how the stop post can be referenced with a certain platform (=serviceSection). Like with an ocpRef, the sign post may directly refer to the ID of a serviceSection via an attribute serviceSectionRef. What do you think?

Kind regards

---
Christian Rahmig
railML.infrastructure coordinator

## Subject: Re: Haltetafel / stop post
## Posted by on Tue, 03 Apr 2012 17:20:58 GMT

View Forum Message <> Reply to Message

Dear Christian,

sorry, but with

> although any railML user didn't reply to your questions...

I am a little bit annoyed because with my message form Monday March 26th
where I indeed thought to reply to Susanne:

> Additionally, I want to remember to Susanne's post from 2011-11-04 where
> she pleads for a stop markers (Haltetafeln) in RailML.

I also wrote a suggestion to handle stop posts both from the view of
infrastructure as well as from a train (timetable).

---

> The stop post itself is a physical element, which is a sign right next
> to the track. Therefore I would not use the crossSection element for
> specifying stop posts.

I agree with you. The crossSection is intended to specify a virtual place
not marked by any physical sign.

> Instead I would put the stop post element inside the ocsElements
> container.

I agree.

> The even more difficult question is how the stop post can be referenced
> with a certain platform (=serviceSection). Like with an ocpRef, the sign
> post may directly refer to the ID of a serviceSection via an attribute
> serviceSectionRef. What do you think?

I also agree in general. But as written in the above mentioned post I
would not create a <serviceSection> but a <platform> (splitting 'passenger
service sections' and other service sections into different elements). But
this is only a small detail which does not change the principle.

From my side, it would also be ok to add the properties of a platform
(orientation, length, height, a. s. o.) directly into the stop post
element and in that way to eliminate the <serviceSection> or <platform> at
all. But I understand that this is probably too much from the operational
view. So defining a <serviceSection> or <platform> and referencing it from
the stop post would be ok from my side.

Best regards,
Dirk.

---

Subject: Re: Haltetafel / stop post
Posted by Christian Rahmig on Tue, 03 Apr 2012 19:07:48 GMT
View Forum Message <> Reply to Message

Hello Dirk,

> Dear Christian,
>
> sorry, but with
>
>> although any railML user didn't reply to your questions...
>
> I am a little bit annoyed because with my message form Monday March 26th
> where I indeed thought to reply to Susanne:
>
>> Additionally, I want to remember to Susanne's post from 2011-11-04
>> where she pleads for a stop markers (Haltetafeln) in RailML.
>
> I also wrote a suggestion to handle stop posts both from the view of
> infrastructure as well as from a train (timetable).

I am sorry, but my comment was not meant to ignore your post from March, 26. On the contrary, it was exactly your post that let me focus on Susanne's post again, which has not been answered until then. So, sorry again for causing any misunderstandings.

>> The stop post itself is a physical element, which is a sign right next
>> to the track. Therefore I would not use the crossSection element for
>> specifying stop posts.
>
> I agree with you. The crossSection is intended to specify a virtual
> place not marked by any physical sign.
>
>> Instead I would put the stop post element inside the ocsElements
>> container.
>
> I agree.

So I suggest defining a new ocsElement named <stopPost>. Like the other ocsElements, it is an optional element and it will be placed in a container <stopPosts>. Required attributes for a <stopPost> element are:

- "id"
- "pos"

Further attributes for describing the stop post may be optional:
   - "serviceSectionRef" for referencing the service section, where the stop post is situated.
   - "stopPostType" for specifying the stop post element.

Connected with the last two attributes, the following two questions need to be answered:
1. Does any stop post exist, which is not referenced to a service section (or platform)?
2. Is it necessary to further specify a stop post element? If so, which types are useful?

>>  The even more difficult question is how the stop post can be
>>  referenced with a certain platform (=serviceSection). Like with an
>>  ocpRef, the sign post may directly refer to the ID of a serviceSection
>>  via an attribute serviceSectionRef. What do you think?
>
>  I also agree in general. But as written in the above mentioned post I
>  would not create a <serviceSection> but a <platform> (splitting
>  'passenger service sections' and other service sections into different
>  elements). But this is only a small detail which does not change the
>  principle.

This idea sounds reasonable to me. However, what do other users think about it (see also discussion in post "Platforms and ramps for railML 2.2")?

>   From my side, it would also be ok to add the properties of a platform
>  (orientation, length, height, a. s. o.) directly into the stop post
>  element and in that way to eliminate the <serviceSection> or <platform>
>  at all. But I understand that this is probably too much from the
>  operational view. So defining a <serviceSection> or <platform> and
>  referencing it from the stop post would be ok from my side.

Interesting idea, but I think that we should keep both elements <stopPost> and <serviceSection> or <platform> since it provides us more flexibility in extending this first approach to the platform problem.

Best regards

---
Christian Rahmig
railML.infrastructure coordinator

Subject: Re: Haltetafel / stop post
Posted by Susanne Wunsch railML on Wed, 04 Apr 2012 08:45:03 GMT
View Forum Message <> Reply to Message

Hello to all,

Christian Rahmig <coord@infrastructure.railml.org> writes:

>>> The stop post itself is a physical element, which is a sign right next
>>> to the track. Therefore I would not use the crossSection element for
>>> specifying stop posts.
>>
>> I agree with you. The crossSection is intended to specify a virtual
>> place not marked by any physical sign.
>>
>>> Instead I would put the stop post element inside the ocsElements
>>> container.
>>
>> I agree.

Yes, that's a good position in the XML tree.

> So I suggest defining a new ocsElement named <stopPost>. Like the
> other ocsElements, it is an optional element and it will be placed in
> a container <stopPosts>. Required attributes for a <stopPost> element
> are:
>   - "id"
>   - "pos"
>
> Further attributes for describing the stop post may be optional:
>   - "serviceSectionRef" for referencing the service section, where the
> stop post is situated.
>   - "stopPostType" for specifying the stop post element.

Please do not repeat the elements' name in the attribute. 'Type' is
often used for 'datatype'. Let's find a more concise term. Which
enumeration should be offered behind this attribute?

> Connected with the last two attributes, the following two questions
> need to be answered:
> 1. Does any stop post exist, which is not referenced to a service
> section (or platform)?
> 2. Is it necessary to further specify a stop post element? If so,
> which types are useful?

Yes, you may define the additional sign.

- train length

- axle count

- wagon count

- verbal definition (S-Bahn Berlin)

- ...

Does anybody know, whether only one of the above constraints may be defined or also combinations of them occur?

```
>>>  The even more difficult question is how the stop post can be
>>>  referenced with a certain platform (=serviceSection). Like with an
>>>  ocpRef, the sign post may directly refer to the ID of a serviceSection
>>>  via an attribute serviceSectionRef. What do you think?
>>
>>  I also agree in general. But as written in the above mentioned post I
>>  would not create a <serviceSection> but a <platform> (splitting
>>  'passenger service sections' and other service sections into different
>>  elements). But this is only a small detail which does not change the
>>  principle.
>
>  This idea sounds reasonable to me. However, what do other users think
>  about it (see also discussion in post "Platforms and ramps for railML
>  2.2")?
```

Good to notice here. Let's discuss this topic in the neighbouring thread.

```
>>   From my side, it would also be ok to add the properties of a platform
>>  (orientation, length, height, a. s. o.) directly into the stop post
>>  element and in that way to eliminate the <serviceSection> or <platform>
>>  at all. But I understand that this is probably too much from the
>>  operational view. So defining a <serviceSection> or <platform> and
>>  referencing it from the stop post would be ok from my side.
>
>  Interesting idea, but I think that we should keep both elements
>  <stopPost> and <serviceSection> or <platform> since it provides us
>  more flexibility in extending this first approach to the platform
>  problem.
```

+1

Kind regards...
Susanne

--
Susanne Wunsch
Schema Coordinator: railML.common

## Subject: Re: Haltetafel / stop post
Posted by on Wed, 04 Apr 2012 17:40:16 GMT
View Forum Message <> Reply to Message

Hi Susanne and Christian,

Am 03.04.2012, 21:07 Uhr, schrieb Christian Rahmig
<coord@infrastructure.railml.org>:
> I am sorry, but my comment was not meant to ignore your post from March,
> 26. On the contrary, it was exactly your post that let me focus on
> Susanne's post again, which has not been answered until then. So, sorry
> again for causing any misunderstandings.

Ok, no problem at all.

Am 04.04.2012, 10:45 Uhr, schrieb Susanne Wunsch <coord@common.railml.org>:

> Let's discuss this topic in the neighbouring thread.

Ok, so I move now to the neighbouring thread. Bye, see you soon...

Dirk.

## Subject: Re: Haltetafel / stop post
Posted by Christian Rahmig on Wed, 04 Apr 2012 21:11:06 GMT
View Forum Message <> Reply to Message

Hello Susanne,

>> So I suggest defining a new ocsElement named<stopPost>. Like the
>> other ocsElements, it is an optional element and it will be placed in
>> a container<stopPosts>. Required attributes for a<stopPost> element
>> are:
>> - "id"
>> - "pos"
>>
>> Further attributes for describing the stop post may be optional:
>> - "serviceSectionRef" for referencing the service section, where the
>> stop post is situated.
>> - "stopPostType" for specifying the stop post element.
>
> Please do not repeat the elements' name in the attribute. 'Type' is
> often used for 'datatype'. Let's find a more concise term. Which
> enumeration should be offered behind this attribute?

As long as we don't know which attributes are required to specify a stop
post in detail, it is difficult to find a more concise term here. Any

comments and suggestions appreciated.

>>  Connected with the last two attributes, the following two questions
>>  need to be answered:
>>  1. Does any stop post exist, which is not referenced to a service
>>  section (or platform)?
>>  2. Is it necessary to further specify a stop post element? If so,
>>  which types are useful?
>
>  Yes, you may define the additional sign.
>
>    - train length
>
>    - axle count
>
>    - wagon count
>
>    - verbal definition (S-Bahn Berlin)
>
>    - ...

Do I understand you right that you want to combine these parameters with
the stop post? What is the idea behind the information about the axle
count or the wagon count? Do you have an example in mind?

---
Christian Rahmig
railML.infrastructure coordinator

---

Subject: Re: Haltetafel / stop post
Posted by Susanne Wunsch railML on Thu, 05 Apr 2012 08:14:12 GMT
View Forum Message <> Reply to Message

Christian Rahmig <coord@infrastructure.railml.org> writes:

>>>  So I suggest defining a new ocsElement named<stopPost>. Like the
>>>  other ocsElements, it is an optional element and it will be placed in
>>>  a container<stopPosts>. Required attributes for a<stopPost>  element
>>>  are:
>>>     - "id"
>>>     - "pos"
>>>
>>>  Further attributes for describing the stop post may be optional:
>>>     - "serviceSectionRef" for referencing the service section, where the
>>>  stop post is situated.
>>>     - "stopPostType" for specifying the stop post element.
>>

>>  Please do not repeat the elements' name in the attribute. 'Type' is
>>  often used for 'datatype'. Let's find a more concise term. Which
>>  enumeration should be offered behind this attribute?
>
>  As long as we don't know which attributes are required to specify a
>  stop post in detail, it is difficult to find a more concise term
>  here. Any comments and suggestions appreciated.

Maybe we don't need the "type" attribute if we offer the more detailed
attributes below.

>>>  Connected with the last two attributes, the following two questions
>>>  need to be answered:
>>>  1. Does any stop post exist, which is not referenced to a service
>>>  section (or platform)?

Maybe for freight trains?

>>>  2. Is it necessary to further specify a stop post element? If so,
>>>  which types are useful?
>>
>>  Yes, you may define the additional sign.
>>
>>    - train length
>>
>>    - axle count
>>
>>    - wagon count
>>
>>    - verbal definition (S-Bahn Berlin)
>>
>>    - ...
>
>  Do I understand you right that you want to combine these parameters
>  with the stop post? What is the idea behind the information about the
>  axle count or the wagon count? Do you have an example in mind?

A stop post may have additional signs which indicate whom stop position
they are requesting. The train length is a widely used example for
that. A short train has to stop at another position than a long train
(for several reasons). The other mentioned constraining characteristics
are not so widely known, but appear in some cases.

A freight train with only few axles has to stop at another position
then a train with a larger number of axles. I don't know why it is
sometimes defined with axle numbers and sometimes with wagon numbers,
maybe for some historical reasons (as always).

de: Eine Halttafel hat in einigen Fällen ein Zusatzschild, welches
angibt, für welche Züge diese Halteposition gilt. Die Angabe einer
Zuglänge ist eine weit verbreitete Zusatzinformation in diesem
Zusammenhang. Die anderen oben aufgeführten Einschränkungen treten
eher selten auf, sind jedoch nicht gänzlich unbekannt.

"100m" (Züge bis 100m Länge),
"40x" (Züge bis zu 40 Achsen)
"Langzug" (S-Bahn-Langzüge)

read you soon...
Susanne

--
Susanne Wunsch
Schema Coordinator: railML.common

---

## Subject: Re: Haltetafel / stop post
## Posted by Christian Rahmig on Sat, 06 Oct 2012 09:43:43 GMT

View Forum Message <> Reply to Message

Dear railML users,

bringing an (temporary) end to a long discussion about stop posts, the
following solution is going to be implemented with railML 2.2:

The new element <stopPost> is introduced (cp. [1]):

- The element stopPost is defined in a container stopPosts within the
ocsElements object.
- The required parameters id, pos and dir are inherited from the type
tOrientedElement.
- Using the optional parameter platformEdgeRef the stop post can be
referenced with a platform for which it is valid.
- Using the optional parameter validForMovements the train types for
which the stop post is relevant can be defined. The enumeration values
freightTrains, passengerTrains, allTrains and shunting are currently
possible.
- If the stop post is only valid for trains with a certain train length,
the optional parameter trainLength needs to be set.
- If the stop post is only valid for trains with a certain number of
axles, the optional parameter axleCount needs to be set.
- If the stop post is only valid for trains with a certain number of
wagons, the optional parameter wagonCount needs to be set.
- If the stop post is only valid for trains fulfilling a certain verbal
constraint, the optional parameter verbalConstraint needs to be set.
- If the stop post is only a virtual element, which is not physically

represented along the track but only in the operational rules, the optional parameter virtual needs to be set to true.

The parameter/additional object for referencing train categories as proposed in [2] hopefully won't be necessary using the optional parameters above.

[1] https://trac.assembla.com/railML/ticket/167
[2]  http://www.railml.org/forum/ro/?group=1&offset=0&thr ead=40&id=110

Regards

--
Christian Rahmig
railML.infrastructure coordinator

## Subject: Re: Haltetafel / stop post

Posted by on Tue, 09 Oct 2012 09:01:47 GMT

View Forum Message <> Reply to Message

Dear Christian,

the summary of attributes/parameters for <stopPost> sounds good. Some short notes:

1) I think we need one more attribute/parameter defining the relation of the stop post to the train. (There were earlier discussions about that feature.) It should be something with the enumerations "frontOfTrain" / "mittleOfTrain" / "endOfTrain":

Most "normal" stop posts we know (from Germany) are valid for the front of the train and are non-virtual. But this is not always the case. For instance in Czech Republic, there are virtual stop posts valid for the end of the train. The virtual stop post stands at the beginning of the platform because the stairs/elevators are at the beginning of the platform. This virtual stop post is valid for the end of the train, which means that the train should not move further forward that that its last door is at the rear end (=beginning) of the platform. This is to avoid an unnecessary long walk for the passengers.

Such virtual stop posts are programmed in the Czech "black boxes" (ATO), and they are therefor "really" existing in software (in practice).

It is also a matter of some national ETCS instances but (unfortunately, in my opinion) not very common in Germany due to the "relying on old INDUSI philosophy". But that should not mean that we forget or ignore it in RailML.

2) Concerning the attributes trainLength, axleCount, wagonCount I would prefer a clarification of the relation: maxAxleCount, maxTrainLength, maxWagonCount. Theoretically you could then also add the other boundaries: minAxleCount, minTrainLength, minWagonCount but they would not have equivalence in practice as far as I know - but again: We should not rely too much at German experience but in case of doubt rather be a little bit more theoretical.

3) Concerning the attribute "validForMovements" is should be possible to enumerate more than one value.

Would be nice if you could consider these notes before releasing 2.2.

Thank you,
Best regards,
Dirk.

---

## Subject: Re: Haltetafel / stop post
## Posted by Christian Rahmig on Fri, 19 Oct 2012 10:52:46 GMT

View Forum Message <> Reply to Message

Dear Dirk and other railML users,

> 1) I think we need one more attribute/parameter defining the relation of
> the stop post to the train. (There were earlier discussions about that
> feature.) It should be something with the enumerations "frontOfTrain" /
> "mittleOfTrain" / "endOfTrain":

In the latest commit regarding trac ticket [1] I added the exisiting attribute "trainRelation" to the element <stopPost>.

> 2) Concerning the attributes trainLength, axleCount, wagonCount I would
> prefer a clarification of the relation: maxAxleCount, maxTrainLength,
> maxWagonCount. Theoretically you could then also add the other
> boundaries: minAxleCount, minTrainLength, minWagonCount but they would
> not have equivalence in practice as far as I know - but again: We should
> not rely too much at German experience but in case of doubt rather be a
> little bit more theoretical.

Let's think of the tram scenario where there are two stop posts situated at a station, one telling "30m" and the next one "45m". If a short train stops at the stop post entitled "45m", the passengers would have to walk more meters. Regarding your explanation of the "maxTrainLength" the short train would be principally allowed to stop at the "45m-stop-post", because its length is shorter.
However, I would skip these detailed view and leave it open to the user

if he/she refers to a minimum or maximum value.

> 3) Concerning the attribute "validForMovements" is should be possible to
> enumerate more than one value.

In the latest commit regarding trac ticket [1] I put the attribute "validForMovement" into a sequence. Now it is possible to define multiple movement validities for the same stop post.

[1] https://trac.assembla.com/railML/ticket/167

Regards

--
Christian Rahmig
railML.infrastructure coordinator

## Subject: Re: Haltetafel / stop post
## Posted by on Fri, 19 Oct 2012 13:58:26 GMT

View Forum Message <> Reply to Message

Dear Christian,

thank you for implementing the extended attributes.

> However, I would skip these detailed view and leave it open to the user
> if he/she refers to a minimum or maximum value.

I think this is a solution which should not be intended by RailML. In the last developing steps of RailML, we tried hard to clarify existing possible misunderstandings. Here, we create a new one.

Please be aware that normally there are at least _two_ "users" of the RailML scheme: The writing and the reading software. How should the writing software tell to the reading software whether "he/she refers to a minimum or maximum value"? It was agreed in RailML that any clarification by scheme is better than one by Wiki.

So, please help the writing software to clarify whether "he/she refers to a minimum or maximum value" by creating the min/max... attributes.

Regards,
Dirk.

P.S.: To answer your tramway example: If the writing software does want to avoid the 30 m train using the 45 m stop post: It can place a "minTrainLength=31 m" and "maxTrainLength=45 m" attribute at the 45 m stop post, so the 30 m train has only one choice.

Subject: Re: Haltetafel / stop post
Posted by thomas.kauer on Thu, 08 Nov 2012 11:16:22 GMT
View Forum Message <> Reply to Message

Dear Christian, dear Dirk

at the SBB we work most time with an absolut "H" independt of train length or with number of units where the length refers to the usualy used vehicle-type in the region (but this value is not indicated). There are also AxleCount and Length ("m") real stop posts. They all indicate where the train header has to stop.
But they are normaly not used as min/max - if you get a 100m and a 200m stop post, a train of 150m has to stop in the middle.

In different programms there are also virtual stop post in use. They can be marked with the enumarations below (front/middle/end) for the same reasons as discussed. There are no real stop posts but there are rules for the station that tell where the train has to stop in that station.

There are also situations where on real stop post is valid for the track on both sides - in this case there will be added an virtual one in the same place but with position on the other track.

Special problems with stop posts in ETCS are not yet fixed as far as I know.

Best regards
Thomas

Dirk BrÃ¤uer wrote:
>
> Dear Christian,
>
> the summary of attributes/parameters for <stopPost> sounds good. Some
> short notes:
>
> 1) I think we need one more attribute/parameter defining the relation of
> the stop post to the train. (There were earlier discussions about that
> feature.) It should be something with the enumerations "frontOfTrain" /
> "mittleOfTrain" / "endOfTrain":
>
> Most "normal" stop posts we know (from Germany) are valid for the front of
> the train and are non-virtual. But this is not always the case. For
> instance in Czech Republic, there are virtual stop posts valid for the end

> of the train. The virtual stop post stands at the beginning of the
> platform because the stairs/elevators are at the beginning of the
> platform. This virtual stop post is valid for the end of the train, which
> means that the train should not move further forward that that its last
> door is at the rear end (=beginning) of the platform. This is to avoid an
> unnecessary long walk for the passengers.
>
> Such virtual stop posts are programmed in the Czech "black boxes" (ATO),
> and they are therefor "really" existing in software (in practice).
>
> It is also a matter of some national ETCS instances but (unfortunately, in
> my opinion) not very common in Germany due to the "relying on old INDUSI
> philosophy". But that should not mean that we forget or ignore it in
> RailML.
>
> 2) Concerning the attributes trainLength, axleCount, wagonCount I would
> prefer a clarification of the relation: maxAxleCount, maxTrainLength,
> maxWagonCount. Theoretically you could then also add the other boundaries:
> minAxleCount, minTrainLength, minWagonCount but they would not have
> equivalence in practice as far as I know - but again: We should not rely
> too much at German experience but in case of doubt rather be a little bit
> more theoretical.
>
> 3) Concerning the attribute "validForMovements" is should be possible to
> enumerate more than one value.
>
> Would be nice if you could consider these notes before releasing 2.2.
>
> Thank you,
> Best regards,
> Dirk.
>
>

--
----== posted via PHP Headliner ==----

Subject: Re: Haltetafel / stop post
Posted by on Thu, 08 Nov 2012 19:31:31 GMT
View Forum Message <> Reply to Message

Dear Thomas,

thank you for the very interesting information.

> ...they are normaly not used as min/max - if you get a 100m and a 200m
> stop post, a train of 150m has to stop in the middle.

In Germany, the "H"s are used quite contrary: A train always has to move forward until the "H" (or main signal) but is not allowed to "interpolate" or such. The reason is the reliance of the very anachronistic "Indusi", especially it's 500-Hz-programme ("INA") to the place where a train stops.

Of course, this is strange, not very practically and especially bad for travelers because the driver is not allowed to stop "convenient for the stairs" (leading to the platform). But, it is not the only thing in Germany which is not very beneficial for railway travelers... So many driver's do even ignore this rule and stop where they want, especially with small rail-cars at 400 m platforms - may be they fear to be beaten by the passengers.

Anyway, it tells us once more that in RailML there should be a possibility to describe "H" with min/max (as in Germany) _and_ "interpolating" (as in Switzerland).

Best regards,
Dirk.

P.S.:
I was aware that there are (normally or up to) three "H" at a platform but I was not aware that

> ...they are normaly not used as min/max - if you get a 100m and a 200m
> stop post, a train of 150m has to stop in the middle.

I'm afraid the SBB teaching people at ETH are also not aware of that because they have told me different and, more important, they requested the ETH's software (which I have been programming) to work differently... So the trains there always stop at the next suitable "H" but do not stop "interpolated" as you describe.

Can you send me a source (may be a clause of the Fahrdienstvorschriften) which I can refer to at ETH? This is of course a non-RailML-question, so may be you can send it via E-Mail. Thank you very much!

---

## Subject: Re: Haltetafel / stop post
Posted by Christian Rahmig on Sat, 10 Nov 2012 09:33:25 GMT

View Forum Message <> Reply to Message

Dear Thomas, Dirk and other railML users,

>> ...they are normaly not used as min/max - if you get a 100m and a 200m
>> stop post, a train of 150m has to stop in the middle.
>
> In Germany, the "H"s are used quite contrary: A train always has to move
> forward until the "H" (or main signal) but is not allowed to
> "interpolate" or such. The reason is the reliance of the very
> anachronistic "Indusi", especially it's 500-Hz-programme ("INA") to the
> place where a train stops.
>
> [...]
>
> Anyway, it tells us once more that in RailML there should be a
> possibility to describe "H" with min/max (as in Germany) _and_
> "interpolating" (as in Switzerland).

in order to be able to handle all three types of stop post validity, I suggest to define three boolean parameters "min", "max" and "interpolating". They are all optional, which allows for having stop posts without further specification, too. The advantage of the boolean parameters is that they can be combined with the axleCount as well as with the trainLength and the wagonCount. Alternatively, the definition of a minimum, a maximum and an interpolating parameter for each of the attributes is required and thus, does not allow for unspecified stop posts.

Further comments appreciated...

Regards

--
Christian Rahmig
railML.infrastructure coordinator

## Subject: Re: Haltetafel / stop post
## Posted by Susanne Wunsch railML on Mon, 12 Nov 2012 09:28:40 GMT

View Forum Message <> Reply to Message

Christian Rahmig <coord@infrastructure.railml.org> writes:
>>> ...they are normaly not used as min/max - if you get a 100m and a 200m
>>> stop post, a train of 150m has to stop in the middle.
>>
>> In Germany, the "H"s are used quite contrary: A train always has to move
>> forward until the "H" (or main signal) but is not allowed to
>> "interpolate" or such. The reason is the reliance of the very
>> anachronistic "Indusi", especially it's 500-Hz-programme ("INA") to the
>> place where a train stops.
>>
>> [...]

>>
>>  Anyway, it tells us once more that in RailML there should be a
>>  possibility to describe "H" with min/max (as in Germany) _and_
>>  "interpolating" (as in Switzerland).
>
>  in order to be able to handle all three types of stop post validity, I
>  suggest to define three boolean parameters "min", "max" and
>  "interpolating".

Do you want to allow multiple value to be used for one stop post?

If only one constraint should be allowed (min, max, interpolating _or_
unknown) than there should be an attribute with an extensible
enumeration list be provided.

>  The advantage of the boolean parameters is that they can be combined
>  with the axleCount as well as with the trainLength and the
>  wagonCount. Alternatively, the definition of a minimum, a maximum and
>  an interpolating parameter for each of the attributes is required and
>  thus, does not allow for unspecified stop posts.

I'm sorry, I don't understand _this advantage_. I would asume that one
attribute with an enumeration list would help out in a similar way.

```
<stopPost id="sp1" axleCount="40" ruled="max" .../>
<stopPost id="sp2" axleCount="80" ruled="max" .../>
<stopPost id="sp3" trainLength="100" ruled="interpolating" .../>
<stopPost id="sp4" trainLength="200" ruled="interpolating" ../>
```

I would prefer a better name for this attribute: "ruled",
"constraint"...

Kind regards...
Susanne

--
Susanne Wunsch
Schema Coordinator: railML.common

Subject: Re: Haltetafel / stop post
Posted by on Mon, 12 Nov 2012 11:34:19 GMT
View Forum Message <> Reply to Message

Dear Christian,

I guess there was a misunderstanding of my sentence

>>  Anyway, it tells us once more that in RailML there should be a
>>  possibility to describe "H" with min/max (as in Germany) _and_
>>  "interpolating" (as in Switzerland).

What I meant is: Give the programmes a hint how to interpret the stop posts.

What I did not mean: Fully describe the operational rules which are valid for stop posts.

In general, we always have to ask us: "Is it infrastructure property or operational rule?" May be in future when everything else is settled and we don't know what to do anymore, we create a RailML scheme for operational rules. But until then, we should keep them out of <infrastructure> and <timetable> and <rollingStock>.

Of course there is a "gray zone" where we cannot be sure whether it is infrastructure property or operational rule. The following questions should help us:

- What we can touch and read (outside our offices) should normally be infrastructure.
- If we cannot touch it: Is the origin an objective physical property or an arbitrary man-made rule?
- If we cannot touch it: Does a driver or signalman has to know it or can he deduce it from something else?

If he has to know it --> rather an operational rule.
If he can deduce or read it --> rather infrastructure.

A speed change we cannot touch, but at least the basic permitted speeds are of physical nature. The drivers do not need to know all the (virtual) speed changes from their minds (normally they can read them from a timetable), so this is infrastructure. But if he cannot read it but has to know or deduce it (e. g. end of speed restriction of some main signals in Germany), it is rather not a speed change, even not a virtual one.

How to interpret stop posts (interpolating, cascaded) the driver normally has to know. If there are any additional signs at the stop post one could read, these would be infrastructure ("30 m", "Passenger trains only" or such). The rule alone ("please stop interpolated between two signs") is not infrastructure.

Additionally, the rules are normally the same in a greater area, so they are "factored out" and expressed indirectly by something like <trackElements><operationModeChange>. To repeat the rule at each stop post would be redundant.

So: Please provide attributes for the additional signs at stop posts but
nothing more. I herewith withdraw my wish for "min", "max" as long as
these words are not actually written at the stop posts.

Best regards,
Dirk.

## Subject: Re: Haltetafel / stop post

Posted by Christian Rahmig on Thu, 22 Nov 2012 09:55:44 GMT

View Forum Message <> Reply to Message

Dear Susanne and other railML users,

>> in order to be able to handle all three types of stop post validity, I
>> suggest to define three boolean parameters "min", "max" and
>> "interpolating".
>
> Do you want to allow multiple value to be used for one stop post?
>
> If only one constraint should be allowed (min, max, interpolating _or_
> unknown) than there should be an attribute with an extensible
> enumeration list be provided.

I agree.

>> The advantage of the boolean parameters is that they can be combined
>> with the axleCount as well as with the trainLength and the
>> wagonCount. Alternatively, the definition of a minimum, a maximum and
>> an interpolating parameter for each of the attributes is required and
>> thus, does not allow for unspecified stop posts.
>
> I'm sorry, I don't understand _this advantage_. I would asume that one
> attribute with an enumeration list would help out in a similar way.

Of course, you are right. The advantage I meant referred to the direct
combination of the constraint with the attribute in one word, e.g.
"minAxleCount" or "maxAxleCount".

> <stopPost id="sp1" axleCount="40" ruled="max" .../>
> <stopPost id="sp2" axleCount="80" ruled="max" .../>
> <stopPost id="sp3" trainLength="100" ruled="interpolating" .../>
> <stopPost id="sp4" trainLength="200" ruled="interpolating" ../>
>
> I would prefer a better name for this attribute: "ruled",
> "constraint"...

How about an enumeration parameter named "bounding"?

Regards

--
Christian Rahmig
railML.infrastructure coordinator

## Subject: Re: Haltetafel / stop post
Posted by Christian Rahmig on Sat, 24 Nov 2012 09:38:17 GMT
View Forum Message <> Reply to Message

Dear Dirk,

```
>>>  Anyway, it tells us once more that in RailML there should be a
>>>  possibility to describe "H" with min/max (as in Germany) _and_
>>>  "interpolating" (as in Switzerland).
>
>  What I meant is: Give the programmes a hint how to interpret the stop
>  posts.
>
>  What I did not mean: Fully describe the operational rules which are
>  valid for stop posts.
>
>  [...]
>
>  So: Please provide attributes for the additional signs at stop posts but
>  nothing more. I herewith withdraw my wish for "min", "max" as long as
>  these words are not actually written at the stop posts.
```

thank you for your detailed explanation about the difference between infrastructure and operational rules, with which I totally agree.
So, we will come back to the definition of the parameter "bounding" with values "min", "max" or "interpolating", when there are stop posts that somehow physically include this information.

Regards

--
Christian Rahmig
railML.infrastructure coordinator

## Subject: Re: Haltetafel / stop post
Posted by Susanne Wunsch railML on Mon, 26 Nov 2012 10:45:25 GMT
View Forum Message <> Reply to Message

Christian Rahmig <coord@infrastructure.railml.org> writes:

>> <stopPost id="sp1" axleCount="40" ruled="max" .../>
>> <stopPost id="sp2" axleCount="80" ruled="max" .../>
>> <stopPost id="sp3" trainLength="100" ruled="interpolating" .../>
>> <stopPost id="sp4" trainLength="200" ruled="interpolating" ../>
>>
>> I would prefer a better name for this attribute: "ruled",
>> "constraint"...
>
> How about an enumeration parameter named "bounding"?

+1

Kind regards...
Susanne

--
Susanne Wunsch
Schema Coordinator: railML.common

Subject: Re: Haltetafel / stop post
Posted by on Mon, 26 Nov 2012 11:45:57 GMT
View Forum Message <> Reply to Message

Dear Christian,

> thank you for your detailed explanation about the difference between
> infrastructure and operational rules, with which I totally agree.

+1

> So, we will come back to the definition of the parameter "bounding" with
> values "min", "max" or "interpolating", when there are stop posts that
> somehow physically include this information.

I am not very satisfied with this solution but - I have no better idea. Obviously we are in the "gray zone" between operational rules and infrastructure, combined with some "illogic" usage of H-posts in practice.

Your suggestion seams to be the best we can do for now. But there would be still many questions left, e. g.:

- If a simulation software founds several H-posts at a track with different properties (min/max train length, axle count, waggon count), the relation between the properties were not clear.

- The terms "min" and "max" could be misunderstood as boundaries of interpolation (outermost H-posts in Switzerland) and as "for all trains up to a length of..." (H-posts in Germany). We intend to mean the latter, but this cannot necessarily be deduced from the terms. So we possibly regulate nothing with the new attribute. The enumeration values of the attribute you suggest would (theoretically) have to be named "for all trains up to" (=max), "for all trains with more than" (=min), "only for trains with exactly" (=interpolating).

Since these problems become clear, I could also imagine that we leave it with the absolute minimum of information which is obviously infrastructure: The additional values or strings which are written at H-posts but nothing more. Since "interpolating", "min", and "max" are (currently) not written at the H-posts, this attribute would be off.

This would possibly mean to provide an optional string attribute for H-posts only. And leave the interpretation of that string either to the reading software (for the moment) or to an <operationalRules> scheme (for the future).

Of course I would also accept the attribute you suggested in spite of the problems which come with it. If you want to introduce it, please consider a self-explaining naming of the enumeration values.

Best regards,
Dirk.

---

Subject: Re: Haltetafel / stop post
Posted by Christian Rahmig on Sun, 02 Dec 2012 10:41:42 GMT
View Forum Message <> Reply to Message

Dear Dirk,

> Your suggestion seams to be the best we can do for now. But there would
> be still many questions left, e. g.:
>
> - If a simulation software founds several H-posts at a track with
> different properties (min/max train length, axle count, waggon count),
> the relation between the properties were not clear.

What kind of relation do you think about: The relation between different stop posts or the relation between different constraints/properties of a stop post?

> - The terms "min" and "max" could be misunderstood as boundaries of
> interpolation (outermost H-posts in Switzerland) and as "for all trains
> up to a length of..." (H-posts in Germany). We intend to mean the

> latter, but this cannot necessarily be deduced from the terms. So we
> possibly regulate nothing with the new attribute. The enumeration values
> of the attribute you suggest would (theoretically) have to be named "for
> all trains up to" (=max), "for all trains with more than" (=min), "only
> for trains with exactly" (=interpolating).

To keep it short we may think about the "bounding" parameter with the values 'upTo', 'moreThan' and 'exactly'.

> Since these problems become clear, I could also imagine that we leave it
> with the absolute minimum of information which is obviously
> infrastructure: The additional values or strings which are written at
> H-posts but nothing more. Since "interpolating", "min", and "max" are
> (currently) not written at the H-posts, this attribute would be off.

I agree. This bounding-issue probably needs more time to think about it and we may have another try with railML 2.3?

> This would possibly mean to provide an optional string attribute for
> H-posts only. And leave the interpretation of that string either to the
> reading software (for the moment) or to an <operationalRules> scheme
> (for the future).

How about using the already introduced stop post parameter "verbalConstraints", which is of type xs:string?

Regards

--
Christian Rahmig
railML.infrastructure coordinator

## Subject: Re: Haltetafel / stop post
Posted by Christian Rahmig on Sun, 02 Dec 2012 11:15:43 GMT
View Forum Message <> Reply to Message

Dear railML users,

> The new element <stopPost> is introduced (cp. [1]):
>
> - The element stopPost is defined in a container stopPosts within the
> ocsElements object.
> - The required parameters id, pos and dir are inherited from the type
> tOrientedElement.
> - Using the optional parameter platformEdgeRef the stop post can be
> referenced with a platform for which it is valid.
> - Using the optional parameter validForMovements the train types for

> which the stop post is relevant can be defined. The enumeration values
> freightTrains, passengerTrains, allTrains and shunting are currently
> possible.
> - If the stop post is only valid for trains with a certain train length,
> the optional parameter trainLength needs to be set.
> - If the stop post is only valid for trains with a certain number of
> axles, the optional parameter axleCount needs to be set.
> - If the stop post is only valid for trains with a certain number of
> wagons, the optional parameter wagonCount needs to be set.
> - If the stop post is only valid for trains fulfilling a certain verbal
> constraint, the optional parameter verbalConstraint needs to be set.
> - If the stop post is only a virtual element, which is not physically
> represented along the track but only in the operational rules, the
> optional parameter virtual needs to be set to true.

just for completeness I want to mention that a stop post can refer to several signals by using the sequence of "signalRef" objects (cp. [2]).

> [1] https://trac.assembla.com/railML/ticket/167

[2] https://trac.assembla.com/railML/ticket/167#comment:16

Regards

--
Christian Rahmig
railML.infrastructure coordinator

---

Subject: Re: Haltetafel / stop post
Posted by Susanne Wunsch railML on Sun, 02 Dec 2012 21:22:40 GMT
View Forum Message <> Reply to Message

Christian Rahmig <coord@infrastructure.railml.org> writes:

> just for completeness I want to mention that a stop post can refer to
> several signals by using the sequence of "signalRef" objects
> (cp. [2]).

What is the 'sequence' attribute used for?

It's needed that a stop post may refer to several signals, as Carsten mentioned at the railML-conference in Zurich. But I don't understand the use case where these referred signals have any 'sequence' relation to the stop post.

Carsten or any other along-reader, please help out with an example from the practice. ;-)

Any comments appreciated.

> [2] https://trac.assembla.com/railML/ticket/167#comment:16

Kind regards...
Susanne

--
Susanne Wunsch
Schema Coordinator: railML.common

## Subject: Re: Haltetafel / stop post
Posted by Christian Rahmig on Mon, 03 Dec 2012 17:59:51 GMT
View Forum Message <> Reply to Message

Dear Susanne,

Am 02.12.2012 22:22, schrieb Susanne Wunsch:
> Christian Rahmig <coord@infrastructure.railml.org> writes:
>
>> just for completeness I want to mention that a stop post can refer to
>> several signals by using the sequence of "signalRef" objects
>> (cp. [2]).
>
> What is the 'sequence' attribute used for?
>
> It's needed that a stop post may refer to several signals, as Carsten
> mentioned at the railML-conference in Zurich. But I don't understand the
> use case where these referred signals have any 'sequence' relation to the
> stop post.

The sequence is not needed. With the latest commit towards Trac ticket
#167 I changed the type of the <signalRef> element to
tElementWithReference leaving out the sequence (cp. [3]).

>> [2] https://trac.assembla.com/railML/ticket/167#comment:16
[3] https://trac.assembla.com/railML/ticket/167#comment:20

Regards

--
Christian Rahmig
railML.infrastructure coordinator

## Subject: Re: Haltetafel / stop post

Posted by Carsten Weber on Mon, 03 Dec 2012 21:44:53 GMT
View Forum Message <> Reply to Message

Here I am.

"Christian Rahmig" <coord@infrastructure.railml.org> schrieb im Newsbeitrag
news:k9j5ch$kb1$1@sifa.ivi.fhg.de...
> Dear Susanne,
>
> Am 02.12.2012 22:22, schrieb Susanne Wunsch:
>> Christian Rahmig <coord@infrastructure.railml.org> writes:
>>
>> It's needed that a stop post may refer to several signals, as Carsten
>> mentioned at the railML-conference in Zurich. But I don't understand the
>> use case where these referred signals have any 'sequence' relation to the
>> stop post.

Such a case exist. (I know only one, but there is one.)

```
-#----
-#----\
-#---------\-----------
             \       |-o
-#------------\-------
                     |-o
# - stop post
\ - point
|-o - main signal
```

The real situation includes some more points. But I hope you can see enough
at this sketch.

Do not hesitate to ask if something is not clear.

Best regards.

Carsten

Subject: Stop posts for different train types (was: Haltetafel / stop post)
Posted by Susanne Wunsch railML on Mon, 11 Mar 2013 16:49:20 GMT
View Forum Message <> Reply to Message

Dear railML users,

During our railML conference in Berlin a new proposal for train
types/kinds was articulated for the context of stop posts.

It is necessary to distinguish between "commercial passenger trains" (with passengers) and "operational passenger trains" (without passengers, only with staff). Former have to stop, latter may pass without any stop. That is an important issue for simulation software.

Christian announced the last changes in this direction with the following posting.

Christian Rahmig <coord@infrastructure.railml.org> writes:
>>  3) Concerning the attribute "validForMovements" is should be possible
>>  to enumerate more than one value.
>
>  In the latest commit regarding trac ticket [1] I put the attribute
>  "validForMovement" into a sequence. Now it is possible to define
>  multiple movement validities for the same stop post.
>
>  [1] https://trac.assembla.com/railML/ticket/167

I summarized the issue in Trac ticket #167 [2]

As stated at the railML conference in Berlin (2013-03-06) further categories for stop posts are needed then currently defined in the "kind" attribute.

Additionally to "passengerTrains" or "freightTrains" or "allTrains" or "shunting" a flag for "not_valid_for_empty_run" is needed.

This is often used in combination with "passengerTrains".

The current implementation quite re-uses the "trainUsage" attribute of the timetabling "category" element. This element also offers a "deadrun" boolean-typed attribute, that would fulfill the new request.

How about referring to an "category" with "categoryRef" instead of the current "kind" attribute?

That would mean to list all possible categories (ICE, ICE_empty, RE, RE_empty...) each stop post or to
define some general categories (passenger, passenger_empty, freight, mixed) therefore.

Any comments* appreciated.

Kind regards...
Susanne

[2] http://trac.assembla.com/railML/ticket/167#comment:22
* +1, -1, hints, questions...
Crosspost: railML.timetable

--
Susanne Wunsch
Schema Coordinator: railML.common

Subject: Re: Stop posts for different train types (was: Haltetafel / stop post)
Posted by on Tue, 12 Mar 2013 19:05:34 GMT
View Forum Message <> Reply to Message

Dear Susanne,

I do not think that it is good to link infrastructure elements with timetable categories. The categories have the character of “flexible, quickly changing, depending on operators, sometimes a matter of marketing and corporate identity, not very technical”.

The infrastructure, in contrary, should be rather “neutral”, independent from operators. If there is a need for an infrastructure element to express any restriction, it should name the actual physical source.

So it is ok to make attributes like “passenger-carrying” or “non-passenger-carrying” or such, but it should be independent from whether they name their trains “ICE” or “RB” or whatever. Please remember that an IM (EIU) has to be prepared to accept any new TOC (EVU) not knowing how they categorise their trains.

Or the other way ‘round: Imagine you are a TOC preparing a bid for an competition. You have the RailML infrastructure file of the IM but you can only see stop posts for “ICE” and “IC”. You do not know which one to take because you do not see the physical background. So what to do? You could ask the IM but this is not in the sense of RailML files (asking per telephone). And, possibly you do not want to ask because some hours later, everybody would know that you are bidding at this competition (via Drehscheibe Online)…

It is the same reason why I pleaded for deleting of the categoryRefs from <speedChanges>: Infrastructure has to be independent from operators.

Concerning the usage so far existing, it is common not to tread a train consisting of coaching stock but non-passenger-carrying (empty coaching stock) as a “passengerTrain”. Or, with other words: A “passengerTrain“ in the technological sense is only a “passengerTrain“ if it is allowed for passengers to use it…

So I do not see any need for a categoryRef.

If you want to extend something: At infrastructure please always try to describe the physical background / reason.

Best regards,
Dirk.

---

Subject: Re: Haltetafel / stop post
Posted by thomas.kauer on Thu, 25 Apr 2013 12:08:31 GMT
View Forum Message <> Reply to Message

Dear Dirk

You asked me for the source for my statement:

>
> P.S.:
> I was aware that there are (normally or up to) three "H" at a platform but
> I was not aware that
>
>> ...they are normaly not used as min/max - if you get a 100m and a 200m
>> stop post, a train of 150m has to stop in the middle.
>
> I'm afraid the SBB teaching people at ETH are also not aware of that
> because they have told me different and, more important, they requested
> the ETH's software (which I have been programming) to work differently...
> So the trains there always stop at the next suitable "H" but do not stop
> "interpolated" as you describe.
>
> Can you send me a source (may be a clause of the Fahrdienstvorschriften)
> which I can refer to at ETH? This is of course a non-RailML-question, so
> may be you can send it via E-Mail. Thank you very much!

I was told so some years ago by specialists working for CUS: they need to
calculate the exact distance from a signal to the stop to predict at which
time the train will open its doors so they can start the announcements at
the platform at the right moment.

To make sure I now asked two engine drivers from SBB how they would stop -
and they also confirmed to interpolate the stop position between the stop
signals (with length indications) according to the real length of their
train.

Best regards
Thomas

--
----== posted via PHP Headliner ==----