Subject: Platforms and ramps for railML 2.2
Posted by Christian Rahmig on Thu, 22 Mar 2012 21:19:09 GMT
View Forum Message <> Reply to Message

Hello everyone,

considering the Trac ticket #122 [1], we are thinking about the introduction of a datatype for platforms and ramps with the next release 2.2. So, this is how it may look like:

1. <trackElements> will be extended with a new element named <serviceSection> for modelling platforms, ramps and related facilities.

2. The new element <serviceSection> describes the part ("section") of a track, which can be used for the exchange of passengers, goods or similar.

3. Attributes for this new element include:
- position information: defines the starting position and direction of the serviceSection
- length: the section length which is defined as the actually usable length (NOT the physically existing platform/ramp which can be longer)
- height: the objects's height in mm above rails (for feasible types only)
- type: enumeration of
  "platform" (passanger platform)
  "ramp" (ramp for loading / unloading goods)
  "maintenance" (maintenance facilities e. g. in a depot)
  "loadingFacility" (Goods can be (un-)loaded from the wagon's top / underfloor)
  "cleaning" (washing facility for cars and engines)
  "fueling" (facility for re-fuelling of engines)
  "parking" (section for parking of rolling stock)
  "preheating" (electricity or steam for pre-heating purposes)
  "other"
- side: right, left (seen in positive mileage direction)

4. Each <serviceSection> can have multiple types. Tracks with platforms on both sides need two <serviceSection> elements to force the definition of different platform names.

5. Additional semantics:
- name: the name contains the platform number (e.g. "3")
- The associated OCP can be de-referenced from the OCP's <trackGroup>

It would be nice to hear your opinion about this model approach. Are there any important parameters missing?

Best regards.

[1] https://trac.assembla.com/railML/ticket/122

Christian Rahmig
railML.infrastructure coordinator

Subject: Re: Platforms and ramps for railML 2.2
Posted by on Mon, 26 Mar 2012 10:49:35 GMT
View Forum Message <> Reply to Message

Hallo Christian and all others,

your suggestion about platforms sounds good. Especially about
> - The associated OCP can be de-referenced from the OCP's <trackGroup>
I think this is very important and not to be missed.

Concerning
> - name: the name contains the platform number (e.g. "3")
we have to take into account that, in some countries like Czech Republic, to distinguish between a platform name and a 'track at platform' name (in German we would say "Bahnsteigkante"). I am not aware whether the 'track at platform' name is always the same as the operational track number (which would make it easier). But since you want to assign the <serviceSection> to a certain track, the Czech principle means that there will be two <serviceSections> sharing the same name (which then means that they share the same physical platform at opposite sides).

Additionally, I want to remember to Susanne's post from 2011-11-04 where she pleads for a stop markers (Haltetafeln) in RailML. I think that the following has to be possible:
- to link a platform with an ocp (as you have already written),
- to link stop markers with the associated platform,
- to link a train's stop with a stop marker and therefore with a platform,
- to define the relation between a train position (front, end, first door...) and the stop marker (and therefore the platform).

All these links shall be done by references (meaning syntactically forced as long as these information are provided in a RailML file at all).

To make it more easier in RailML, I think that it is not important to have a general base type for platforms, ramps and maintenance facilities. A platform is (in my opinion) such a familiar phenomen to all railways that we can provide a special type for it.

All in all, my recommendation is:
- to define an element called 'platform marker' with the attributes (you suggested) 'length', 'direction', 'height', 'platformName',

'platformTrackName', 'validForTrainLenghs'. The marker may be in the middle of the platform so that we may have two lengths and two directions. With 'validForTrainLenghs' I mean the typical additional plates with "100 m" or so written on it.
  - to define an optional reference from an ocpTT to a 'platform marker' with the relation of a position in the train (front, end, first door...)..

This leaves the ramps, loading- and maintenance facilities for other elements so far.

We still have the problem Susanne wrote, where several coupled train parts may have inconsistent 'platform relations' (e. g. The first train part says 'I want to stop with my front at the end of the platform' and the last train part says 'I want to stop with my end at the beginning of the platform'. This may be functional if the platform length is the same as the train length but if not... it creates at least a gap in the train or, more worst, a gap in the space-time-continuum.)

However, we shall consider that train parts may have different operating days. So it may definitely make sense if all train parts want to stop at the end of the platform because there may be days when each of the train parts is the first one in the train. Considering this, we shall leave the inconsistent cases to the reading software - we cannot save us from everything and there are more worst possible inconsistencies in RailML.

With best regards,
Dirk.

Am 22.03.2012, 22:19 Uhr, schrieb Christian Rahmig <coord@infrastructure.railml.org>:

> Hello everyone,
>
> considering the Trac ticket #122 [1], we are thinking about the
> introduction of a datatype for platforms and ramps with the next release
> 2.2. So, this is how it may look like:
>
> 1. <trackElements> will be extended with a new element named
> <serviceSection> for modelling platforms, ramps and related facilities..
>
> 2. The new element <serviceSection> describes the part ("section") of a
> track, which can be used for the exchange of passengers, goods or
> similar.
>
> 3. Attributes for this new element include:
>   - position information: defines the starting position and direction

> of the serviceSection
> - length: the section length which is defined as the actually usable
> length (NOT the physically existing platform/ramp which can be longer)
> - height: the objects's height in mm above rails (for feasible types
> only)
> - type: enumeration of
> "platform" (passanger platform)
> "ramp" (ramp for loading / unloading goods)
> "maintenance" (maintenance facilities e. g. in a depot)
> "loadingFacility" (Goods can be (un-)loaded from the wagon's
> top / underfloor)
> "cleaning" (washing facility for cars and engines)
> "fueling" (facility for re-fuelling of engines)
> "parking" (section for parking of rolling stock)
> "preheating" (electricity or steam for pre-heating purposes)
> "other"
> - side: right, left (seen in positive mileage direction)
>
> 4. Each <serviceSection> can have multiple types. Tracks with platforms
> on both sides need two <serviceSection> elements to force the definition
> of different platform names.
>
> 5. Additional semantics:
> - name: the name contains the platform number (e.g. "3")
> - The associated OCP can be de-referenced from the OCP's <trackGroup>
>
> It would be nice to hear your opinion about this model approach. Are
> there any important parameters missing?
>
> Best regards.
>
> [1] https://trac.assembla.com/railML/ticket/122
>
> ---
> Christian Rahmig
> railML.infrastructure coordinator

--
Erstellt mit Operas revolutionärem E-Mail-Modul: http://www.opera.com/mail/

Subject: joined continue: "Platforms and ramps for railML 2.2" and "Haltetafel / stop post"
Posted by on Wed, 04 Apr 2012 17:40:20 GMT
View Forum Message <> Reply to Message

....here we are again from the neighbouring thread.

This is the continuation of both the topics "Platforms and ramps for railML 2.2" and "Haltetafel / stop post" as ordered by the overall "Lady Of Common RailML".

> 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>.:

I agree.

> 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.

I agree with your suggestion and also with Susanne's advice:

> 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?

I don't think that we should name it 'type' at all. We should give it only attributes for these properties which are really (physically) existing at the sign post itself. Any properties which rely to the platform rather then to the stop post should be written at the <serviceSection> or <platform> element.

The attributes of a stop post should be from my side:
--> "id"
--> "pos"
--> "serviceSectionRef" (optional, see below)
--> direction (at the track - up or down) (mandatory)
--> additional conditions (this relies to Susannes train length, axle count, wagon count, verbal remark) (optional)
--> 'sign code' or 'rule number' or so (this shall allow to define the 'Ne5' or 'So8' of DS/DV301 or something like that)
--> valid for train categories (categoryRefs, none, one or more)

Concerning the attributes, we should ask us the questions:
 - Should it be possible to define a <stopPost> as 'virtual'? Virtual means that there is no real stop post sign but it is a place where one could or should stay (where trains have to stop). This may be important for the reference of a train to a <stopPost>: If a train has to stop at a platform where there is no <stopPost> (may be because the starter signal directly standing at the end or simply because somebody of DB Projektbau has forgotten to plan it) - what shall the <trainPart> in RailML do? My recommendation: Allow virtual <stopPosts> with an attribute

--> "virtual" (Boolean).

- Do we want to include an attribute whether the <stopPost> is valid for shunting movements and/or trains? This would include the German "Ra10" into stop posts. In Germany, we do normally not think that a Ra10 is a Haltetafel but one could have the opinion that the word Halttafel does include "RangierHALTTAFEL" ;-). However, in other countries these "shunting limits" may be more naturally stop posts. I would recommend to include them, which brings us to an attribute

--> "validForMovements" (enumeration: FreightTrains, PassengerTrains, AllTrains, Shunting, both?).

(It is all 'Arbeitstitel' only.)

The reason for "PassengerTrains" and "AllTrains" is: In Germany normally a H-Tafel is valid for passenger trains only except if there is no starter signal in the track where it is valid for all trains.

> - 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?

In Germany, only a train length is allowed in nowadays. But there may be axle count at older signs or in other countries.

We should allow combinations to ease future discussions...

> 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)?

Yes, of course. There are stop posts in tracks w/o platform. In Germany at least they must be in main tracks which have no starter signal. (In these cases, the stop post replaces the starter signal, making it to a very important security technology element. For instance, the overlap starts at the stop post.)

There may be also stop posts in tracks w/o platform but with starter if there is a Reisendenzugang over the track (leading to a platform at other tracks).

> ...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.

No objection.

Best regards,
Dirk.

Subject: Re: joined continue: "Platforms and ramps for railML 2.2" and "Haltetafel / stop post"
Posted by Christian Rahmig on Wed, 04 Apr 2012 21:42:32 GMT
View Forum Message <> Reply to Message

Hello Dirk,

>> 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.
>
> I agree with your suggestion and also with Susanne's advice:
>
>> 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?
>
> I don't think that we should name it 'type' at all. We should give it
> only attributes for these properties which are really (physically)
> existing at the sign post itself. Any properties which rely to the
> platform rather then to the stop post should be written at the
> <serviceSection> or <platform> element.

I absolutely agree with your statement.

> The attributes of a stop post should be from my side:
> --> "id"
> --> "pos"
> --> "serviceSectionRef" (optional, see below)
> --> direction (at the track - up or down) (mandatory)
> --> additional conditions (this relies to Susannes train length, axle
> count, wagon count, verbal remark) (optional)
> --> 'sign code' or 'rule number' or so (this shall allow to define the
> 'Ne5' or 'So8' of DS/DV301 or something like that)
> --> valid for train categories (categoryRefs, none, one or more)

Still I do not fully understand the idea behind the additional conditions "axle count", "wagon count" and "verbal remark" (cp. post in subject "Haltetafel / stop post").

The sign code is important, but I would use the parameter "code" for it,
which already exists for any ocsElement.

> Concerning the attributes, we should ask us the questions:
> - Should it be possible to define a <stopPost> as 'virtual'? Virtual
> means that there is no real stop post sign but it is a place where one
> could or should stay (where trains have to stop). This may be important
> for the reference of a train to a <stopPost>: If a train has to stop at
> a platform where there is no <stopPost> (may be because the starter
> signal directly standing at the end or simply because somebody of DB
> Projektbau has forgotten to plan it) - what shall the <trainPart> in
> RailML do? My recommendation: Allow virtual <stopPosts> with an attribute
>
> --> "virtual" (Boolean).

That is an interesting idea. The question is if there physically exists
a "substitute" for the stop post, which can be used for stopping a
train. Since we always talk about infrastructure, I'd rather see a
physical element providing the function of a stop post instead of
defining virtual elements. However, we need to analyse this problem.

> - Do we want to include an attribute whether the <stopPost> is valid for
> shunting movements and/or trains? This would include the German "Ra10"
> into stop posts. In Germany, we do normally not think that a Ra10 is a
> Haltetafel but one could have the opinion that the word Halttafel does
> include "RangierHALTTAFEL" ;-). However, in other countries these
> "shunting limits" may be more naturally stop posts. I would recommend to
> include them, which brings us to an attribute
>
> --> "validForMovements" (enumeration: FreightTrains, PassengerTrains,
> AllTrains, Shunting, both?).
>
> (It is all 'Arbeitstitel' only.)
>
> The reason for "PassengerTrains" and "AllTrains" is: In Germany normally
> a H-Tafel is valid for passenger trains only except if there is no
> starter signal in the track where it is valid for all trains.

Another important idea that you are bringing up here. For clarification
we have to ask ourselves whether we in fact talk about a combination of
two different signs/signals just at the same position along the track or
about a certain feature of a stop post, which cannot be found in a
separate context.

>> 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)?

>
> Yes, of course. There are stop posts in tracks w/o platform. In Germany
> at least they must be in main tracks which have no starter signal. (In
> these cases, the stop post replaces the starter signal, making it to a
> very important security technology element. For instance, the overlap
> starts at the stop post.)
>
> There may be also stop posts in tracks w/o platform but with starter if
> there is a Reisendenzugang over the track (leading to a platform at
> other tracks).

Since there are stop posts without any reference to a <platform> or <serviceSection> element, the parameter "serviceSectionRef" can only be optional as you already mentioned.

Best regards

Christian Rahmig
railML.infrastructure coordinator

Subject: Re: joined continue: "Platforms and ramps for railML 2.2" and "Haltetafel / stop post"
Posted by on Thu, 05 Apr 2012 00:09:38 GMT
View Forum Message <> Reply to Message

> Still I do not fully understand the idea behind the additional
> conditions "axle count", "wagon count" and "verbal remark" (cp. post in
> subject "Haltetafel / stop post").

These refer to the small white additional signs below the normal black German H-Tafel. It is necessary to have these additional signs if there ist more than one stop post at one station track per direction. Nowadays, there is always a (maximum) length of train written at these additional signs. In former times it was more common to write an axle count. May be you have seen a writing like "Rz 48x" below a H-Tafel which means "for passenger trains with up to 48 axles".

It should not be conditions but simple properties.

> The sign code is important, but I would use the parameter "code" for it,
> which already exists for any ocsElement.

Since 'code' is an inherited attribute, it does naturally not rely to any special property of a stop post. It should be saved for more general usage, like an external primary key or so. For instance, if you describe a point (Weiche) you should use 'code' for the point number (Weichennummer)

which is the external primary key. You should not write the rule book number of the point's signal into 'code'.

> That is an interesting idea. The question is if there physically exists
> a "substitute" for the stop post, which can be used for stopping a
> train. Since we always talk about infrastructure, I'd rather see a
> physical element providing the function of a stop post instead of
> defining virtual elements. However, we need to analyse this problem.

Even if there would be a substitute it would be useless in this context. My explanation was "what shall the <trainPart> in RailML do?". I assume that there will be a possibility for a <trainPart> to reference a stop post in future - like a <trainPart>.stopPostRef attribute. I assume that you can only fill in references to stop posts - not to any other infrastructure elements. That's why I think it is necessary to have virtual stop posts.

Please, RailML is not a database for bureaucracy. It is functional and for data exchange. So please do think functional.

What do you want to tell me with "we need to analyse this problem"?

There are virtual stop posts with and without "substitutes". A substitute may be a Ra11 or So5/Ne1 in Germany. Often there is no stop post because there is a starter signal directly at the end of the platform. You can decide for yourself if a starter signal is a substitute for a stop post. But in many cases I know there is simply nothing but the end of a platform. I can send pictures for all these examples but I think we all know them.

For instance, there are normally no stop posts in stations like Leipzig Hbf. (Have you ever seen a stop post at the platform tracks of Leipzig Hbf.?) I assume that there will be a possibility to define the relation of a train to the platform by referencing a stop post (e. g. train stands with its front at the end of the platform or with its rear at the beginning of the platform or with its middle at the middle of the platform). I think that it should be possible to define these relations also in Leipzig Hbf. That's why I think we should allow virtual stop posts. These relations are essential for run time calculation if you have a MU of 35 m length at a 400 m platform with permitted speed of 40 km/h...

Enough arguing. There is a demand and so there should be a solution.

>> The reason for "PassengerTrains" and "AllTrains" is: In Germany normally
>> a H-Tafel is valid for passenger trains only except if there is no
>> starter signal in the track where it is valid for all trains.
>
> Another important idea that you are bringing up here. For clarification

> we have to ask ourselves whether we in fact talk about a combination of
> two different signs/signals just at the same position along the track or
> about a certain feature of a stop post, which cannot be found in a
> separate context.

There are no two different signs at the same location. You have these "all train stop posts" also in other countries. I think they are a kind of normal, typical for railways, natural. For instance I think at the big markers 'Stop - obtain token to proceed...' with the big red splash at RETB lines.

Dirk.

Subject: Re: Platforms and ramps for railML 2.2
Posted by Christian Rahmig on Sun, 16 Sep 2012 07:48:53 GMT
View Forum Message <> Reply to Message

Dear railML users,

> 1. <trackElements> will be extended with a new element named
> <serviceSection> for modelling platforms, ramps and related facilities.
>
> 2. The new element <serviceSection> describes the part ("section") of a
> track, which can be used for the exchange of passengers, goods or similar.
>
> 3. Attributes for this new element include:
> - position information: defines the starting position and direction
> of the serviceSection
> - length: the section length which is defined as the actually usable
> length (NOT the physically existing platform/ramp which can be longer)
> - height: the objects's height in mm above rails (for feasible types
> only)
> - type: enumeration of
> "platform" (passanger platform)
> "ramp" (ramp for loading / unloading goods)
> "maintenance" (maintenance facilities e. g. in a depot)
> "loadingFacility" (Goods can be (un-)loaded from the wagon's
> top / underfloor)
> "cleaning" (washing facility for cars and engines)
> "fueling" (facility for re-fuelling of engines)
> "parking" (section for parking of rolling stock)
> "preheating" (electricity or steam for pre-heating purposes)
> "other"
> - side: right, left (seen in positive mileage direction)
>
> 4. Each <serviceSection> can have multiple types. Tracks with platforms
> on both sides need two <serviceSection> elements to force the definition

> of different platform names.
>
> 5. Additional semantics:
> - name: the name contains the platform number (e.g. "3")
> - The associated OCP can be de-referenced from the OCP's <trackGroup>

including the comments from Dirk and the discussions with the other schema coordinators, platforms and service sections will be implemented in railML 2.2 as follows (c.f. [1]):

1. <trackElements> will be extended with two new elements:
<platformEdges> for modelling platforms
<serviceSections> for modelling ramps and related facilities like maintenance and fueling areas.

2. A <platformEdge> contains the following attributes:
- position information: defines the starting position and direction of the platform
- length: the length of the platform from an operational view (usable length)
- height: the platform's height in mm above rails
- side: right, left (seen in positive direction of the track)

3. A <serviceSection> contains the following attributes:
- position information, length, height and side as described for the platformEdge
- various boolean attributes for a functional classification of the service section. A service section may fulfill multiple functions and therefore multiple boolean attributes may be set true.

[1] https://trac.assembla.com/railML/ticket/122

Regards

--
Christian Rahmig
railML.infrastructure coordinator

Subject: Re: Platforms and ramps for railML 2.2
Posted by on Tue, 02 Oct 2012 16:55:06 GMT
View Forum Message <> Reply to Message

Dear Christian,

> 2. A <platformEdge> contains the following attributes:
> - position information: defines the starting position and direction
> of the platform

> - length: the length of the platform from an operational view
> (usable length)
> - height: the platform's height in mm above rails
> - side: right, left (seen in positive direction of the track)

This leaves the question 'Where shall a train to stop at a platform which is longer than the train?' for future RailML extensions.

(This question is relatively important in practice - it can even influence the official run time of a train. This is also the question about 'stop posts' - H-Tafeln - and whether there can be none or more than one at a platform and how they are valid.)

Dirk.

Subject: Re: joined continue: "Platforms and ramps for railML 2.2" and "Haltetafel / stop post"
Posted by Christian Rahmig on Sat, 06 Oct 2012 09:58:36 GMT
View Forum Message <> Reply to Message

Hello Dirk,

in trac ticket [1] I concluded the whole subject on stop posts for an implementation in railML 2.2 considering most of the ideas mentioned in the forum entries here and in the original [2].

However, few things are still open, e.g.:
>> The sign code is important, but I would use the parameter "code" for
>> it, which already exists for any ocsElement.
>
> Since 'code' is an inherited attribute, it does naturally not rely to
> any special property of a stop post. It should be saved for more general
> usage, like an external primary key or so. For instance, if you describe
> a point (Weiche) you should use 'code' for the point number
> (Weichennummer) which is the external primary key. You should not write
> the rule book number of the point's signal into 'code'.

How about using the parameter "name" for this? In fact, the element's type abbreviation is not only interesting for stop posts, but it is also attached to points (EW, IBW, DKW...) or signals in general (Ra10, So12...). So, I think, a more global parameter is required? Any comments appreciated...

[1] https://trac.assembla.com/railML/ticket/167
[2] http://www.railml.org/forum/ro/index.php?group=1&offset= 20&thread=34

Regards

--
Christian Rahmig
railML.infrastructure coordinator

Subject: Re: Platforms and ramps for railML 2.2
Posted by Christian Rahmig on Sat, 06 Oct 2012 10:02:41 GMT
View Forum Message <> Reply to Message

Hello Dirk,

> This leaves the question 'Where shall a train to stop at a platform
> which is longer than the train?' for future RailML extensions.
>
> (This question is relatively important in practice - it can even
> influence the official run time of a train. This is also the question
> about 'stop posts' - H-Tafeln - and whether there can be none or more
> than one at a platform and how they are valid.)

does the implementation of the stop post subject as described in [1]
concluding the ideas of the forum discussion in [2] fulfill all your
requirements?

[1] https://trac.assembla.com/railML/ticket/167
[2] http://www.railml.org/forum/ro/index.php?group=1&offset= 20&thread=34

Regards

--
Christian Rahmig
railML.infrastructure coordinator

Subject: Re: Platforms and ramps for railML 2.2
Posted by on Tue, 09 Oct 2012 09:03:21 GMT
View Forum Message <> Reply to Message

Hello Christian,

> does the implementation of the stop post subject as described in [1]
> concluding the ideas of the forum discussion in [2] fulfill all your
> requirements?

Yes, thank you, with the attention to the notes I wrote there.

Dirk.

Subject: "Sign code" for stop posts (was: joined continue: "Platforms and ramps for railML 2.2" and "Haltetafel / stop post")
Posted by Susanne Wunsch on Tue, 13 Nov 2012 09:03:49 GMT
View Forum Message <> Reply to Message

Dirk Bräuer <dirk.braeuer@irfp.de> writes:

> The attributes of a stop post should be from my side:
[...]
> --> 'sign code' or 'rule number' or so (this shall allow to define the
> Ne5' or 'So8' of DS/DV301 or something like that)

If filed a Trac ticket for this issue in order to clarify it with railML
2.2:

http://trac.assembla.com/railML/ticket/198

Kind regards...
Susanne

--
Susanne Wunsch
Schema Coordinator: railML.common

Subject: Re: "Sign code" for stop posts
Posted by Christian Rahmig on Mon, 03 Dec 2012 19:05:58 GMT
View Forum Message <> Reply to Message

Dear Susanne and Dirk,

Am 13.11.2012 10:03, schrieb Susanne Wunsch:
> Dirk Bräuer <dirk.braeuer@irfp.de> writes:
>
>> The attributes of a stop post should be from my side:
> [...]
>> --> 'sign code' or 'rule number' or so (this shall allow to define the
>> Ne5' or 'So8' of DS/DV301 or something like that)
>
> If filed a Trac ticket for this issue in order to clarify it with railML
> 2.2:
>
> http://trac.assembla.com/railML/ticket/198

the so-called "sign code" is a parameter, which belongs to more elements
than just the stop post. Considering the DS/DV301 the name "stop post"
is identical with the codes 'Ne5' or 'So8'.

I suggest to extend the concept of "sign code" to all <signal> and <stopPost> elements within the <ocsElements> container. Instead of "sign code" I would name it "ocsCode".

Any comments appreciated...

Regards

--
Christian Rahmig
railML.infrastructure coordinator

Subject: Re: "Sign code" for stop posts
Posted by on Mon, 03 Dec 2012 19:14:54 GMT
View Forum Message <> Reply to Message

Dear Christian,

> I suggest to extend the concept of "sign code" to all <signal> and
> <stopPost> elements within the <ocsElements> container. Instead of "sign
> code" I would name it "ocsCode".

Agreed. But more general, this applies not only to <ocsElements> but also to (at least some) <trackElements> so please do not call it "ocsCode".

I would prefer to use the name for notifying that we mean a rule book number. So what about "ruleBookCode" or "ruleCode"? (Could later become "ruleRef" if we possibly will have <operationalRules> scheme in far futuer...)

Best regards,
Dirk.

Subject: Re: "Sign code" for stop posts
Posted by Susanne Wunsch on Fri, 18 Jan 2013 10:32:37 GMT
View Forum Message <> Reply to Message

Dear Dirk and Christian,

Dirk Bräuer <dirk.braeuer@irfp.de> writes:
>> I suggest to extend the concept of "sign code" to all <signal> and
>> <stopPost> elements within the <ocsElements> container. Instead of
>> "sign code" I would name it "ocsCode".
>
> Agreed. But more general, this applies not only to <ocsElements> but

> also to (at least some) <trackElements> so please do not call it
> "ocsCode".

Which other elements do also apply to get the new attribute?

I just implemented the optional attribute 'ruleCode' of type xs:string
for the elements 'stopPost' and 'signal'. [1] [2]

Please check out, if this implementation fulfills your needs.

Kind regards...
Susanne

[1] http://trac.assembla.com/railML/changeset/529
[2] https://trac.assembla.com/railML/ticket/198

Subject: Re: "Sign code" for stop posts
Posted by Ferri Leberl on Wed, 19 Apr 2017 13:25:16 GMT
View Forum Message <> Reply to Message

@ruleCode has been added to <derailer>, too.

The attribute has meanwhile been documented in the wiki at its three occurrences.

In my view, Ticket #198 can be closed.

Yours,
Ferri Leberl
