
Subject: Speed Panels: types and reference to <speedChange>

Posted by [pierre.simon](#) on Wed, 04 Jul 2012 17:54:48 GMT

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

In the Belgian railway five different speed panels types exists :

- Announcement
- Origin
- reference
- End_zone_green
- End_zone_yellow

An example will be provided in the next days to illustrate these types.

It would be nice to have the <speedPanel> element in a next version of railML, and with the following types

- announcement
- execution
- repetition

In order to make the connection with the speed profile on the tracks, a reference from the <speedPanel> to the <speedChange> is needed.

[de: Es wird ein Element fuer Geschwindigkeitstafeln/Lf-Signale benoetigt. Sie sollten eine Referenz zu dem entsprechenden <speedChange> haben. Zudem werden obige Typen zur Charakterisierung empfohlen.]

--

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

Subject: Re: Speed Panels: types and reference to <speedChange>

Posted by [Christian Rahmig](#) on Sat, 01 Sep 2012 08:38:17 GMT

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

Hello Pierre,

> In the Belgian railway five different speed panels types exists :

- > -Announcement
- > -Origin
- > -reference
- > -End_zone_green
- > -End_zone_yellow

>

> An example will be provided in the next days to illustrate these types.

>

> It would be nice to have the <speedPanel> element in a next version of railML, and with the following types

- > - announcement

- > - execution
- > - repetition

the <speedPanel> element within its container <speedPanels> is a special type of a panel and therefore implemented within the <panels> container in <ocsElements> with railML 2.2 as described in the post [1].

The definition of various speed panel types is a good extension to the panel concept. It can be either realized by defining a new parameter "speedPanelType" having the values 'announcement', 'execution' and 'reminder' or by setting up boolean parameters for each type, i.e.

"announcementPanel",

"executionPanel" and

"reminderPanel".

Any comments about a preferred solution are appreciated.

- > In order to make the connection with the speed profile on the tracks, a
- > reference from the <speedPanel> to the <speedChange> is needed.
- >
- > [de: Es wird ein Element fuer Geschwindigkeitstafeln/Lf-Signale benoetigt.
- > Sie sollten eine Referenz zu dem entsprechenden <speedChange> haben. Zudem
- > werden obige Typen zur Charakterisierung empfohlen.]

As you correctly mentioned, speed panels may refer to actual speed changes, e.g. at a speed restriction section. I suggest a parameter "speedChangeRef" within the element <speedPanel>, which refers to the ID of a <speedChange> element.

However, this concept of referencing other elements may be applied to other types of panels as well. In particular, catenary panels may refer to <electrificationChange> elements or levelcrossing panels can link to <levelCrossing> objects.

Regards

[1]

<http://www.railml.org/forum/ro/index.php?group=1&offset= 0&thread=54&id=148>

--

Christian Rahmig

railML.infrastructure coordinator

Subject: Re: Speed Panels: types and reference to <speedChange>

Posted by [Christian Rahmig](#) on Wed, 24 Oct 2012 13:29:48 GMT

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

Dear Pierre and other railML users,

- > the <speedPanel> element within its container <speedPanels> is a special
- > type of a panel and therefore implemented within the <panels> container
- > in <ocsElements> with railML 2.2 as described in the post [1].

in the meantime the ongoing discussion in thread [1] agreed on combining signals and panels in the <signal> element. For distinguishing between (static) panels and (switchable) signals the boolean parameter "switchable" has been introduced.

- > The definition of various speed panel types is a good extension to the
- > panel concept. It can be either realized by defining a new parameter
- > "speedPanelType" having the values 'announcement', 'execution' and
- > 'reminder' or by setting up boolean parameters for each type, i.e.
- > "announcementPanel",
- > "executionPanel" and
- > "reminderPanel".

With the definition of <signalAspect> sub-elements within <signal>, which may also include panel information, it is useful to rename the boolean parameters into "announcement", "execution" and "reminder" and put them in the <signalAspect> element.

- >> In order to make the connection with the speed profile on the tracks, a
- >> reference from the <speedPanel> to the <speedChange> is needed.
- >>
- > As you correctly mentioned, speed panels may refer to actual speed
- > changes, e.g. at a speed restriction section. I suggest a parameter
- > "speedChangeRef" within the element <speedPanel>, which refers to the ID
- > of a <speedChange> element.
- >
- > However, this concept of referencing other elements may be applied to
- > other types of panels as well. In particular, catenary panels may refer
- > to <electrificationChange> elements or levelcrossing panels can link to
- > <levelCrossing> objects.

In the proposed implementation, which is described in trac ticket [2], the parameter "elementRef" is introduced for the <signalAspect> element. Depending on the above mentioned boolean parameters defining the type of the signal aspect, the appropriate infrastructure element can be referenced, e.g. a <speedChange>.

[1] <http://www.railml.org/forum/ro/?group=1&id=148>

[2] <https://trac.assembla.com/railML/ticket/173>

Regards

--

Subject: Re: Speed Panels: types and reference to <speedChange>
Posted by [Susanne Wunsch railML](#) on Wed, 24 Oct 2012 14:14:10 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear Christian, Pierre and others,

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

>> the <speedPanel> element within its container <speedPanels> is a special
>> type of a panel and therefore implemented within the <panels> container
>> in <ocsElements> with railML 2.2 as described in the post [1].
>
> in the meantime the ongoing discussion in thread [1] agreed on
> combining signals and panels in the <signal> element. For
> distinguishing between (static) panels and (switchable) signals the
> boolean parameter "switchable" has been introduced.

+1

>> The definition of various speed panel types is a good extension to the
>> panel concept. It can be either realized by defining a new parameter
>> "speedPanelType" having the values 'announcement', 'execution' and
>> 'reminder' or by setting up boolean parameters for each type, i.e.
>> "announcementPanel",
>> "executionPanel" and
>> "reminderPanel".

> With the definition of <signalAspect> sub-elements within <signal>,
> which may also include panel information, it is useful to rename the
> boolean parameters into "announcement", "execution" and "reminder" and
> put them in the <signalAspect> element.

I would appreciate to allow these characteristics only for signals that may have an announcement, execution or reminder. That are quite all. I know. E.g. "real signals" and "speed signals" (panels). But how about a sign for a treadle (de: Schienenkontakt), e.g. at a level crossing? I mean, that sign does neither has an "announcement" nor an "reminder". ;-) But it needs a link to its infrastructure facility (e.g. level crossing).

Therefore I strongly recommend to define sub-elements for the different kinds of signals: speed, catenary, level crossing... with its appropriate attributes (characteristics).

If the above mentioned characteristics are introduced as boolean-typed

attributes, more than one of them may occur. Are there any situations, where a combination of these characteristics (announcement, execution, reminder) is shown for the same aspect? I don't mean an announced speed of 60 kph next to the execution of 80 kph at the same pole.

>> As you correctly mentioned, speed panels may refer to actual speed
>> changes, e.g. at a speed restriction section. I suggest a parameter
>> "speedChangeRef" within the element <speedPanel>, which refers to the ID
>> of a <speedChange> element.
>>
>> However, this concept of referencing other elements may be applied to
>> other types of panels as well. In particular, catenary panels may refer
>> to <electrificationChange> elements or levelcrossing panels can link to
>> <levelCrossing> objects.
>
> In the proposed implementation, which is described in trac ticket [2],
> the parameter "elementRef" is introduced for the <signalAspect>
> element. Depending on the above mentioned boolean parameters defining
> the type of the signal aspect, the appropriate infrastructure element
> can be referenced, e.g. a <speedChange>.

This leads to a very heavy overloaded element "signalAspect" with the possibility to mix topics together that should be better separated.

Not to say, that this element can't be validated in any useful manner. No xs:keyref mechanism works here. You may mix catenary information with speed and level crossing related topics. No XML validation shows an error!

We should try to better model this topic and define semantically separated elements. No problem to use generic types in the background, but not generic elements in the foreground!

> [1] <http://www.railml.org/forum/ro/?group=1&id=148>
> [2] <https://trac.assembla.com/railML/ticket/173>

Kind regards...

Susanne

--

Susanne Wunsch

Schema Coordinator: railML.common

Subject: Re: Speed Panels: types and reference to <speedChange>

Posted by [Christian Rahmig](#) on Sat, 27 Oct 2012 08:40:40 GMT

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

Hello Susanne and other railML users,

>>> The definition of various speed panel types is a good extension to the
>>> panel concept. It can be either realized by defining a new parameter
>>> "speedPanelType" having the values 'announcement', 'execution' and
>>> 'reminder' or by setting up boolean parameters for each type, i.e.
>>> "announcementPanel",
>>> "executionPanel" and
>>> "reminderPanel".
>
>> With the definition of <signalAspect> sub-elements within <signal>,
>> which may also include panel information, it is useful to rename the
>> boolean parameters into "announcement", "execution" and "reminder" and
>> put them in the <signalAspect> element.
>
> I would appreciate to allow these characteristics only for signals that
> may have an announcement, execution or reminder. That are quite all. I
> know. E.g. "real signals" and "speed signals" (panels). But how about a
> sign for a treadle (de: Schienenkontakt), e.g. at a level crossing? I
> mean, that sign does neither has an "announcement" nor an
> "reminder". ;-) But it needs a link to its infrastructure facility
> (e.g. level crossing).
>
> Therefore I strongly recommend to define sub-elements for the different
> kinds of signals: speed, catenary, level crossing... with its appropriate
> attributes (characteristics).

For the ongoing discussion about the different types of signals to be defined, please also regard my last post in the forum thread [1].

>> In the proposed implementation, which is described in trac ticket [2],
>> the parameter "elementRef" is introduced for the <signalAspect>
>> element. Depending on the above mentioned boolean parameters defining
>> the type of the signal aspect, the appropriate infrastructure element
>> can be referenced, e.g. a <speedChange>.
>
> This leads to a very heavy overloaded element "signalAspect" with the
> possibility to mix topics together that should be better separated.
>
> Not to say, that this element can't be validated in any useful
> manner. No xs:keyref mechanism works here. You may mix catenary
> information with speed and level crossing related topics. No XML
> validation shows an error!
>
> We should try to better model this topic and define semantically
> separated elements. No problem to use generic types in the background,
> but not generic elements in the foreground!

I totally agree with your idea of semantically separated elements. But this separation requires definite categories for signal types. As mentioned in my post in [1] it is often difficult to exactly link the signal with a signal type on a macroscopic level.

So, our task for railML 2.2 is to define these categories and to define them in such a way that there won't be any misunderstandings when choosing a signal type. In the trac ticket [2] I proposed the following categories and ask for your approval/denial/addition:

- speed,
- etcs,
- levelCrossing,
- gsm,
- catenary and
- signalingSystem

[1]

<http://www.railml.org/forum/ro/index.php?group=1&offset=0&thread=54&id=148>

[2] <https://trac.assembla.com/railML/ticket/173>

Regards

--

Christian Rahmig
railML.infrastructure coordinator

Subject: Re: Speed Panels: types and reference to <speedChange>
Posted by [Susanne Wunsch railML](#) on Thu, 01 Nov 2012 14:01:31 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Christian and others,

>>> In the proposed implementation, which is described in trac ticket [2],
>>> the parameter "elementRef" is introduced for the <signalAspect>
>>> element. Depending on the above mentioned boolean parameters defining
>>> the type of the signal aspect, the appropriate infrastructure element
>>> can be referenced, e.g. a <speedChange>.

>>

>> This leads to a very heavy overloaded element "signalAspect" with the
>> possibility to mix topics together that should be better separated.

>>

>> Not to say, that this element can't be validated in any useful
>> manner. No xs:keyref mechanism works here. You may mix catenary
>> information with speed and level crossing related topics. No XML
>> validation shows an error!

>>

>> We should try to better model this topic and define semantically

>> separated elements. No problem to use generic types in the background,
>> but not generic elements in the foreground!
>
> I totally agree with your idea of semantically separated elements. But
> this separation requires definite categories for signal types. As
> mentioned in my post in [1] it is often difficult to exactly link the
> signal with a signal type on a macroscopic level.
>
> So, our task for railML 2.2 is to define these categories and to
> define them in such a way that there won't be any misunderstandings
> when choosing a signal type.

Good idea. Let's start.

> In the trac ticket [2] I proposed the
> following categories and ask for your approval/denial/addition:
>
> - speed,

The main problem is that all "traditional signals" are more or less
"switchable speed signals" for certain ranges.

Maybe we can distinct between "pure fixed speed signals" and "switchable
speed signals" with country-specific signal aspects.

> - etcs,

- * Fixed marker boards

- * light signals indicating a new movement authority (for ETCS level 1)

- * (level crossing signals), likely don't exist

- * no balises or balise groups, but they may transfer the "traditional
signal aspect" as "speed restriction for a certain range".

> - levelCrossing,

several kinds of signals for ringing the bell, whistle, announcing,
activating the level crossing ...

> - gsm,

more general: trainRadio

- * fixed train radio signals indicating a channel

- * fixed GSM signals indicating an GSM-R-equipped area, also the

negation indicating a non-GSM-R-equipped area, maybe with an attribute to "enter"/"leave", also used for a "radio hole"

> - catenary

- * no catenary section
- * no current section / main power switch off
- * lower pantograph section
- * additional signal (de: ICE-Schaltmerkhilfe)

- trainProtectionSystem

- * indication a block number (de: LZB-Blockkennzeichen)

- line

- * indicating the line number

- milepost (de:Hektometertafel)

- * indicating the mileage of a track, often fixed at a pole

- braking

- * non-stopping section
- * no regenerative braking
- * no eddy-current braking
- * no magnetic-shoe braking

> - signalingSystem

Currently no idea how to harmonize it in another way than an enumeration of signal systems with their possible signal aspects.

It can get messy to cover all these special kinds of signals, have a look at <http://www.stellwerke.de> for the German, French and Luxemburg (explanations in German only). But not to characterize them results messy for the data exchange.

> [1]

> <http://www.railml.org/forum/ro/index.php?group=1&offset= 0&thread=54&id=148>

> [2] <https://trac.assembla.com/railML/ticket/173>

Sorry for the detailed lists, I tried to find some examples for the categories for some better understanding.

Feel free to comment, move the details and categories around, add some new...

Kind regards...
Susanne

--
Susanne Wunsch
Schema Coordinator: railML.common

Subject: Re: Speed Panels: types and reference to <speedChange>
Posted by on Thu, 01 Nov 2012 15:29:15 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear Susanne and Christian,

two small thoughts to your conversation on modeling panels:

>> - gsm,
> more general: trainRadio
>
> * fixed GSM signals indicating an GSM-R-equipped area, also the
> negation indicating a non-GSM-R-equipped area, maybe with an
> attribute to "enter"/"leave", also used for a "radio hole"

1)
Please consider how to handle the "virtual" aspects of all these kinds of information: There may be line-side places where to leave/enter sections of train radio, ETCS, catenary or such _without_ any sign/panel!

In Germany, we note such "train radio entering/leaving/changing places" in the driver's timetable but there are normally no line-side panels. To fully exchange or describe a timetable (may be for something like EBUa) we have to describe the virtual places too.

So please declare how to handle such cases:

a) Your conversation seems to be started because of the line-side "panels" - not because of the virtual places. So may be you only want to handle the real panels here only. But then we would need to have some kind of "train radio change place" elements anywhere else in RailML to handle the virtual cases, and both would be possibly redundant.

b) Alternatively, it may be wanted to handle the virtual and non-virtual places in one structure, just to avoid redundancy. So, the "virtual: boolean" attribute would be needed and the terminology should avoid suggesting that there must be a real panel.

2)

In the cases we already have the "infrastructure property change element" list, as with everything in <track>.<trackElements>... such as <electrificationChange>:

Are you aware that you create a big amount of redundancy with the new <panels> with type=.... enumeration which enumerates nearly all the same we already have in <track>.<trackElements>.<...Change> ?

A reading software would always have the (additional) problems of the kind: "What do to if I pass a panel 'electrification change' but there is no <electrificationChange> at the same place?"

I am aware that there may be panels not exactly at the same place where the change is (announcement or just problems of space). So that's why I take into account "optionally". But for the many, many cases where panels and changes are at the same place we should avoid redundancy _or_ we should explain that it is not redundancy.

If you decide that redundancy cannot be avoided here, please declare exactly how to handle the mixture of virtual and non-virtual change places in practice. I do not want to get RailML files where <panels> are used to describe virtual changes just because of an undefined situation in RailML.

Thank you,
best regards,
Dirk.

Subject: Re: Speed Panels: types and reference to <speedChange>
Posted by [Susanne Wunsch railML](#) on Thu, 01 Nov 2012 17:34:08 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear Dirk, Christian and others,

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

> 1)
> Please consider how to handle the "virtual" aspects of all these kinds
> of information: There may be line-side places where to leave/enter
> sections of train radio, ETCS, catenary or such _without_ any
> sign/panel!

For all those "virtual" changes the special <trackElements> can be

used.

- Currently railML lacks the train radio change element. [1] It should be introduced at minimum with a "channel" attribute based on the "tOrientedElement" type.
- railML further lacks some ETCS area element suitable for different ETCS levels.

Maybe the track/trackTopology/border/@borderType attribute could be used and enhanced for this.

- The typical catenary panels can be defined by the element track/trackElements/trackConditions/trackCondition with the attributes length and type (lowerPantograph, mainPowerSwitchOff).

Otherwise use electrificationChange/@type for non-electrified sections.

- > 2)
- > In the cases we already have the "infrastructure property change element" list, as with everything in <track>.<trackElements>... such
- > as <electrificationChange>:
- >
- > Are you aware that you create a big amount of redundancy with the new
- > <panels> with type=.... enumeration which enumerates nearly all the
- > same we already have in <track>.<trackElements>.<...Change> ?

Yes we are aware of this fact. We don't want to repeat the information on the panel but to refer to its definition, e.g.

```
<signal id="s1" pos="10.5" dir="up" assembly="pole">
  <signalFrame height="2">
    <speed kind="announcement" switchable="false"
      speedChangeRef="sc1"/>
  </signalFrame>
  <signalFrame height="2.3">
    <trackCondition kind="execution" switchable="false"
      trackConditionRef="tr1"/>
  </signalFrame>
</signal>
```

That's probably not the latest state of the signal/panel discussion.

The above examples shows why I prefer to define special sub-elements for each panel type instead of some general panel type.

> A reading software would always have the (additional) problems of the
> kind: "What do to if I pass a panel 'electrification change' but there
> is no <electrificationChange> at the same place?"

The XML validator should highlight an error if there is no such electrificationChange element referred to. But it can't detect if the appropriate <electrificationChange> element is to far away from its announcement panel.

I hope to clarify a bit and to correctly understand Christian's idea. ;-)

Any comments are highly appreciated.

Kind regards...
Susanne

[1] <https://trac.assembla.com/railML/ticket/43>

--

Susanne Wunsch
Schema Coordinator: railML.common

Subject: Re: Speed Panels: types and reference to <speedChange>
Posted by [Christian Rahmig](#) on Sat, 10 Nov 2012 12:12:05 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear Susanne and other railML users,

>> In the trac ticket [2] I proposed the
>> following categories and ask for your approval/denial/addition:
>>
>> - speed,
>
> The main problem is that all "traditional signals" are more or less
> "switchable speed signals" for certain ranges.
>
> Maybe we can distinct between "pure fixed speed signals" and "switchable
> speed signals" with country-specific signal aspects.

As you mentioned in your example, it may look like this:
<speedSignal kind="announcement" switchable="false" speedChangeRef="sc1" />
All information about the country specific signal aspects are more
connected with the operational view of the signal.

>> - etcs,
>

- > * Fixed marker boards
- >
- > * light signals indicating a new movement authority (for ETCS level 1)
- >
- > * (level crossing signals), likely don't exist
- >
- > * no balises or balise groups, but they may transfer the "traditional signal aspect" as "speed restriction for a certain range".

Example:

```
<etcsSignal kind="execution" switchable="false" level="1" />
```

- >> - levelCrossing,
- >
- > several kinds of signals for ringing the bell, whistle, announcing,
- > activating the level crossing ...

Example:

```
<levelCrossingSignal type="whistle" />
```

- >> - gsm,
- >
- > more general: trainRadio
- >
- > * fixed train radio signals indicating a channel
- >
- > * fixed GSM signals indicating an GSM-R-equipped area, also the negation indicating a non-GSM-R-equipped area, maybe with an attribute to "enter"/"leave", also used for a "radio hole"

Example:

```
<trainRadioSignal switchable="false" type="enter" />
```

- >> - catenary
- >
- > * no catenary section
- >
- > * no current section / main power switch off
- >
- > * lower pantograph section
- >
- > * additional signal (de: ICE-Schaltmerkhilfe)

Example:

```
<catenarySignal type="lowerPantograph" />
```

- > - trainProtectionSystem
- >

> * indication a block number (de: LZB-Blockkennzeichen)

Example:

```
<trainProtectionSystemSignal />
```

> - line

>

> * indicating the line number

Example:

```
<lineSignal lineNumber="1234" />
```

> - milepost (de:Hektometertafel)

>

> * indicating the mileage of a track, often fixed at a pole

Example:

```
<milepostSignal value="42.0" />
```

> - braking

>

> * non-stopping section

>

> * no regenerative braking

>

> * no eddy-current braking

>

> * no magnetic-shoe braking

Example:

```
<brakingSignal type="nonStoppingSection" />
```

>> - signalingSystem

Use the <trainProtectionSystemSignal> instead.

Does this concept please everybody's needs at least for a railML 2.2?

Regards

--

Christian Rahmig

railML.infrastructure coordinator

Subject: Panels in general (was: Speed Panels: types and reference to
<speedChange>)

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

```
>>> In the trac ticket [2] I proposed the
>>> following categories and ask for your approval/denial/addition:
>>>
>>> - speed,
>
> As you mentioned in your example, it may look like this:
> <speedSignal kind="announcement" switchable="false" speedChangeRef="sc1" />
```

Quite fine. I would name it only 'speed' because it's some child element of a 'signal' element. That hint could be rolled out for all of the following examples.

```
>>> - etcs,
>>
>> * Fixed marker boards
>>
>> * light signals indicating a new movement authority (for ETCS level 1)
>>
>> * (level crossing signals), likely don't exist
>>
>> * no balises or balise groups, but they may transfer the "traditional
>>   signal aspect" as "speed restriction for a certain range".
>
> Example:
> <etcsSignal kind="execution" switchable="false" level="1" />
```

Sorry, if you tried to describe the marker boards, a marker board is not ETCS-level-specific.

```
>>> - levelCrossing,
>>
>> several kinds of signals for ringing the bell, whistle, announcing,
>> activating the level crossing ...
>
> Example:
> <levelCrossingSignal type="whistle" />
```

How about the 'levelCrossingRef'?

```
>>> - gsm,
>>
>> more general: trainRadio
>>
>> * fixed train radio signals indicating a channel
```



```
>>
>>  * fixed GSM signals indicating an GSM-R-equipped area, also the
>>    negation indicating a non-GSM-R-equipped area, maybe with an
>>    attribute to "enter"/"leave", also used for a "radio hole"
>
> Example:
> <trainRadioSignal switchable="false" type="enter" />
```

How about a "trainRadioRef"?

Yes, we need such an element in 'trackElements' for indicating GSM-R and/or analogue train radio with specified channels/frequencies. [1]

```
>>> - catenary
>>
>>  * no catenary section
>>
>>  * no current section / main power switch off
>>
>>  * lower pantograph section
>>
>>  * additional signal (de: ICE-Schaltmerkhilfe)
>
> Example:
> <catenarySignal type="lowerPantograph" />
```

How about a 'trackConditionRef' instead of the 'type' attribute in order to reduce redundancy?

```
>> - line
>>
>>  * indicating the line number
>
> Example:
> <lineSignal lineNumber="1234" />
```

Please don't repeat the elements name in the attribute: 'code' would be fine. Maybe some country uses combinations of numbers and letters for their lines.

```
>> - milepost (de:Hektometertafel)
>>
>>  * indicating the mileage of a track, often fixed at a pole
>
> Example:
> <milepostSignal value="42.0" />
```

The mile post needs at minimum two values, one is printed large and one

(the correction) is printed in small letters.

```
>> - braking
>>
>> * non-stopping section
>>
>> * no regenerative braking
>>
>> * no eddy-current braking
>>
>> * no magnetic-shoe braking
>
> Example:
> <brakingSignal type="nonStoppingSection" />
```

How about a 'trackConditionRef' instead of the 'type' attribute in order to reduce redundancy?

Christian, would you please summarize this thread and update the Trac ticket? [2]

[1] <http://trac.assembla.com/railML/ticket/43>

[2] <http://trac.assembla.com/railML/ticket/173>

Kind regards...
Susanne

--
Susanne Wunsch
Schema Coordinator: railML.common

Subject: Re: Panels in general
Posted by [Christian Rahmig](#) on Mon, 03 Dec 2012 12:56:35 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear Susanne and other railML users,

regarding the recent aspects mentioned within this forum thread, I updated the Trac ticket [1] with the following aspects:

- In the implementation of railML 2.2 a <signal> contains exactly one signal information. Therefore, in case of several signals/panels on one pole, several <signal> elements need to be defined.

- For the different types of signal information, we propose signals/panels for speed, etcs, levelCrossing, trainRadio, catenary, line, signpost. They are modelled as sub-elements of signal.

- All the signal information sub-elements do not inherit the parameters "id", "name" and "code" since they are already given by the parent signal element.
- All the signal information sub-elements contain a boolean parameter "switchable", which allows for distinguishing between fixed (panel) or switchable (e.g. light signal), signal information.
- A <speed> signal information further contains the parameters "kind" (values: announcement, execution and end) and "speedChangeRef".
- A <etcs> signal information further contains the parameter "level" with values '1', '2' and '3'.
- A <levelCrossing> signal information further contains the parameters "type" (values: 'bell', 'whistle', 'announcing' and 'activating') and "levelCrossingRef".
- A <trainRadio> signal information further contains the parameter "trackConditionRef", which refers to a <trackCondition> where the attribute type allows for the value 'radioHole'.
- A <catenary> signal information further contains the parameter "trackConditionRef", which refers to a <trackCondition> where the attribute type allows for the values 'lowerPantograph' and 'mainPowerSwitchOff'.
- A <line> signal information further contains the parameter "lineRef", which refers to a <line> element.
- A <milepost> signal information further contains the parameters "shownValue" and "realValue".
- A <braking> signal information further contains the parameter "trackConditionRef", which refers to a <trackCondition> where the attribute "type" allows for the values 'nonStoppingSection', 'noRegenerativeBraking', 'noEddyCurrentBraking', 'noMagneticShoeBraking'.
- All the signal information sub-elements contain the "any" attribute.

[1] <https://trac.assembla.com/railML/ticket/173>

Regards

--

Christian Rahmig
railML.infrastructure coordinator

Subject: Re: Panels in general

Posted by on Mon, 03 Dec 2012 15:17:05 GMT

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

Dear Susanne and Christian,

> - A <speed> signal information further contains the parameters "kind"
> (values: announcement, execution and end) and "speedChangeRef".

A list of <speedChangeRefs> please. (Several speed profiles, each referencing the same <speedChange>.)

I suggest not to distinguish between "execution" and "end" because of this may depend on the speed profile. Rather, an attribute /kind/"announcement"|"execution" should be enough also for the "ends".

Additionally, we need both the "valid for head of train" (not the same as /kind/!) and "virtual" information as well, as already discussed and agreed.

> - A <milepost> signal information further contains the parameters
> "shownValue" and "realValue".

Is "realValue" the same as the /pos/ of this element at the track, isn't it?

> - A <braking> signal information further contains the parameter
> "trackConditionRef", which refers to a <trackCondition> where the
> attribute "type" allows for the values 'nonStoppingSection',
> 'noRegenerativeBraking', 'noEddyCurrentBraking', 'noMagneticShoeBraking'.

I suggest to use a brake type already existing (tVehicleBrakes or such) and an attribute declaring "not for vehicles with the following brakes...".

Please do already think at the German places for "Betriebsbremsung" or "Zwangshalt" - should we use such a <signal> there, possibly with /virtual/'true'? If so, we would also use a /type/ therefore.

Best regards,
Dirk.

Subject: Re: Panels in general

Posted by [Christian Rahmig](#) on Sat, 12 Jan 2013 17:10:50 GMT

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

Dear Dirk and other railML users,

Am 03.12.2012 16:17, schrieb Dirk Bräuer:

> Dear Susanne and Christian,

>

>> - A <speed> signal information further contains the parameters

>> "kind" (values: announcement, execution and end) and "speedChangeRef".

>

> A list of <speedChangeRefs> please. (Several speed profiles, each

> referencing the same <speedChange>.)

+1

> I suggest not to distinguish between "execution" and "end" because of

> this may depend on the speed profile. Rather, an attribute

> /kind="announcement"|"execution" should be enough also for the "ends".

+1

> Additionally, we need both the "valid for head of train" (not the same

> as /kind/!) and "virtual" information as well, as already discussed and

> agreed.

The attribute @virtual already exists for the <signal> element and does not have to be declared again in the sub-element. For the information about the point of validity of the signal/panel, I added the attribute @trainRelation to the signal's sub-element <speed>. Probably, it is even useful to implement the attribute @trainRelation in the <signal> element as it might be of interest for other types of signals as well.

>> - A <milepost> signal information further contains the parameters

>> "shownValue" and "realValue".

>

> Is "realValue" the same as the /pos/ of this element at the track, isn't

> it?

You are right. It is sufficient to only declare the attribute @shownValue.

>> - A <braking> signal information further contains the parameter

>> "trackConditionRef", which refers to a <trackCondition> where the

>> attribute "type" allows for the values 'nonStoppingSection',

>> 'noRegenerativeBraking', 'noEddyCurrentBraking', 'noMagneticShoeBraking'.

>

> I suggest to use a brake type already existing (tVehicleBrakes or such)

> and an attribute declaring "not for vehicles with the following brakes...".

Since we are only referencing to a <trackCondition> element, this would mean to change the attributes of the <trackCondition>.

> Please do already think at the German places for "Betriebsbremsung" or

> "Zwangshalt" - should we use such a <signal> there, possibly with
> /virtual/='true'? If so, we would also use a /type/ therefore.

Don't we have implemented the <speedChange> attributes @mandatoryStop and @mandatoryBraking for that? Consequently, we need to add a reference from a braking signal to a <speedChange> element as we defined it for all speed signals.

> Best regards,
> Dirk.

Thank you for your comments! I considered them in the Trac ticket [1] as well.

[1] <https://trac.assembla.com/railML/ticket/173>

Regards

--

Christian Rahmig
railML.infrastructure coordinator

Subject: Re: Panels in general
Posted by [Susanne Wunsch railML](#) on Fri, 18 Jan 2013 11:49:06 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear Christian, Dirk and others,

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

> Am 03.12.2012 16:17, schrieb Dirk Bräuer:
>>> - A <speed> signal information further contains the parameters
>>> "kind" (values: announcement, execution and end) and "speedChangeRef".
>> A list of <speedChangeRefs> please. (Several speed profiles, each
>> referencing the same <speedChange>.)

+1, implemented. [1]

>> I suggest not to distinguish between "execution" and "end" because of
>> this may depend on the speed profile. Rather, an attribute
>> /kind/="announcement"|"execution" should be enough also for the "ends".

+1, to be implemented.

>> Additionally, we need both the "valid for head of train" (not the same
>> as /kind/!) and "virtual" information as well, as already discussed and
>> agreed.
>

> The attribute @virtual already exists for the <signal> element and
> does not have to be declared again in the sub-element.

+1

> For the information about the point of validity of the signal/panel, I
> added the attribute @trainRelation to the signal's sub-element
> <speed>. Probably, it is even useful to implement the attribute
> @trainRelation in the <signal> element as it might be of interest for
> other types of signals as well.

Disagree. The @trainRelation is already defined in the 'stopPost' and 'speedChange' elements. The 'trainRelation' of a "speed signal" may be found following signal/speed/@speedChangeRef.

What other kinds of signals may need a @trainRelation?

>>> - A <milepost> signal information further contains the parameters
>>> "shownValue" and "realValue".
>> Is "realValue" the same as the /pos/ of this element at the track, isn't
>> it?

+1

>>> - A <braking> signal information further contains the parameter
>>> "trackConditionRef", which refers to a <trackCondition> where the
>>> attribute "type" allows for the values 'nonStoppingSection',
>>> 'noRegenerativeBraking', 'noEddyCurrentBraking',
>>> 'noMagneticShoeBraking'.
>> I suggest to use a brake type already existing (tVehicleBrakes or
>> such) and an attribute declaring "not for vehicles with the following
>> brakes...".

-1, there are no signals/panels next to the track that define the possible braking characteristics for this track. But there are sometimes panels for such rough characteristics like the list above (copied from the ETCS specification for "track conditions").

>> Please do already think at the German places for "Betriebsbremsung" or
>> "Zwangshalt" - should we use such a <signal> there, possibly with
>> /virtual/= 'true'? If so, we would also use a /type/ therefore.

>

> Don't we have implemented the <speedChange> attributes @mandatoryStop
> and @mandatoryBraking for that? Consequently, we need to add a
> reference from a braking signal to a <speedChange> element as we
> defined it for all speed signals.

I'm a bit confused about this.

Where should one define the "mandatory stop" in front of a level crossing?

trackElements/speedChanges/speedChange/@mandatoryStop and
ocsElements/signals/signal/speed/speedChangeRef/@ref

or

trackElements/speedChanges/speedChange/@mandatoryStop and
ocsElements/signals/signal/braking/speedChangeRef/@ref

I would not add an subelement 'speedChangeRef' to signal/braking, but refer to "signal/speed/speedChangeRef" for the mandatory stop and braking at the "speed/braking" wiki page.

Does somebody should define a 'virtual' signal if the characteristic is already defined somewhere else?

I mean that the 'speedChange' definition may be enough, the 'signal/speed' could be only used in case of "known speed panels".

There is already @signalised in the element 'speedChange' that has the same meaning like signal[@virtual=true]/speed.

I would welcome to clarify the use cases for
speedChange[@signalised=false] and signal[@virtual=true]/speed.

Any comments appreciated.

Kind regards...
Susanne

--

Susanne Wunsch
Schema Coordinator: railML.common

Subject: Mile post properties (was: Panels in general)
Posted by [Susanne Wunsch railML](#) on Tue, 12 Mar 2013 11:18:17 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear railML users,

During our meeting in Berlin some questions popped up around the new element "milepost". Christian introduced its implementation in this thread.

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

> Am 03.12.2012 16:17, schrieb Dirk Bräuer:

>>> - A <milepost> signal information further contains the parameters

>>> "shownValue" and "realValue".

>> Is "realValue" the same as the /pos/ of this element at the track, isn't

>> it?

> You are right. It is sufficient to only declare the attribute @shownValue.

I summarized the open points in Trac ticket #228. [1]

The implementation of mile post panels may be back-traced following ticket #173.

During the railML conference in Berlin (2013-03-06) two aspects were shortly discussed:

* Are there any "switchable" mile posts in any country?

It seems a bit strange to change the shown mileage depending on whatever.

Current idea is, to remove the "switchable" attribute from the milepost element.

* Is a reference to mileageChanges needed?

Are there any mile post panels, that show both mileages (incoming and outgoing)?

If that is true, a "mileageChangeRef" and a "nextShowValue" attribute have to be added.

Any comments* appreciated.

Kind regards...

Susanne

[1] <http://trac.assembla.com/railML/ticket/228>

* +1, 1, hints, questions...

--

Susanne Wunsch

Schema Coordinator: railML.common

Subject: Re: Mile post properties (was: Panels in general)

Posted by on Fri, 19 Jul 2013 18:09:47 GMT

> * Is a reference to mileageChanges needed?

I would prefer such a reference. The mile post panel may be situated at a place where it should not be - some meters or more away from the theoretical position. This is the normal case in Germany: At electrified lines, they are mounted at the catenary support pylon. But catenary support pylons do not necessarily stand at a difference of exactly 100 m and also not at the integer hectometers.

You can normally read the exact position of the post written in small digits at the bottom-right corner of the panel.

So w/o the reference to mileageChanges, it may be difficult to assign the panel to the right mileage section. (In cases where one cannot deduce this from the mileage value at the post.)

By the way: You should clarify whether the relative position of the mile post panel in RailML should be that one where it really stands or that one which corresponds to the absolute value written on it. I assume that one where it really stands.

> Are there any mile post panels, that show both mileages (incoming
> and outgoing)?
>
> If that is true, a "mileageChangeRef" and a "nextShowValue"
> attribute have to be added.

Yes, there are: I put it up for you at [1].

The mile post panel (which is on metric miles in this case - also know as kilometers outside the British Empire) shows 72,760 = 72,634.

But please be very careful with this picture: Please do not forward or publish it, please respect the authorship. It is for NSA purposes only, since it also shows some other highly confidential information which are not common for the general public:

- Your name does not need to be Usain Bolt or such to walk 100 m in under 10 s. Here, everybody can do it even at rough ballasted railway sleepers.. If this would be generally known, the sales of doping drugs would slide into free fall.

- The line to Eisenberg (Thüringen) is not really closed nor abandoned. On the contrary, it even sees BR 612 operation with tilting technology.

For short: This photo is the proof that "tears in the space-time continuum" really happen, namely in Gera Hbf. Of course we all did already

know that there must be something wrong with places such as Gera.

You will understand that I cannot keep such photos on-line for a longer period. So, if you read these lines sometime after they have been written, the tear in the space-time continuum may already be closed again.

With best regards,
Dirk.

Legend:

[1] <http://download.irfp.de/UGera-120312-05028.jpg>

Subject: Re: Mile post properties

Posted by [Christian Rahmig](#) on Sun, 10 Nov 2013 21:07:52 GMT

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

Dear Dirk,

Am 19.07.2013 20:09, schrieb Dirk Bräuer:

>> * Is a reference to mileageChanges needed?

>

> I would prefer such a reference. The mile post panel may be situated at
> a place where it should not be - some meters or more away from the
> theoretical position. This is the normal case in Germany: At electrified
> lines, they are mounted at the catenary support pylon. But catenary
> support pylons do not necessarily stand at a difference of exactly 100 m
> and also not at the integer hectometers.

>

> You can normally read the exact position of the post written in small
> digits at the bottom-right corner of the panel.

>

> So w/o the reference to mileageChanges, it may be difficult to assign
> the panel to the right mileage section. (In cases where one cannot
> deduce this from the mileage value at the post.)

The reference from the <milepost> to the <mileageChange> has been implemented following the remarks collected in Trac ticket [1] within railML 2.2.

> By the way: You should clarify whether the relative position of the mile
> post panel in RailML should be that one where it really stands or that
> one which corresponds to the absolute value written on it. I assume that
> one where it really stands.

+1

When speaking about the relative position of the milepost, I would first think of the physical element's position.

[1] <http://trac.assembla.com/railML/ticket/228>

Best regards

--

Christian Rahmig
railML.infrastructure coordinator
