Subject: Signals vs panels
Posted by pierre.simon on Wed, 04 Jul 2012 17:34:08 GMT
View Forum Message <> Reply to Message

There should be a discussion about the semantics of Â« signals Â» and

value or by by element names.
A future <panels> may hold all types of panels, e.g. speedPanels. Those panels should get a trackside attribute (information important because they are physically installed on the track).
- <LevelCrossingPanels>
- <catenarySectioningPanels>
- <currentPanels>
- <pantographPanels>
- <fixedSignalingChangePanels>
- <linePanels>
- <gsmPanels>
- <radioPanels>

New type of element should be added in the standard to describe the panels from the infrastrucure of the railway.

[de: Wie koennen variable Signale und fixe Signaltafel in railML unterschieden werden? Eine Liste moeglicher Signaltafeln wurde oben vorgeschlagen.]

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

Subject: Re: Signals vs panels
Posted by Christian Rahmig on Sat, 01 Sep 2012 07:58:19 GMT
View Forum Message <> Reply to Message

Hello Pierre,

> There should be a discussion about the semantics of Â« signals Â» and
> â€œpanelsâ€

Subject: Re: Signals vs panels
Posted by Carsten Weber on Thu, 06 Sep 2012 08:25:18 GMT
View Forum Message <> Reply to Message

"Christian Rahmig" <coord@infrastructure.railml.org> schrieb im Newsbeitrag

news:k1spas$1ro$1@sifa.ivi.fhg.de...
> Hello Pierre,
>
>> There should be a discussion about the semantics of « signals » and
>> “panels” / switchable or fixed aspects / distinction by an attribute
>> value or by by element names.
>> A future <panels> may hold all types of panels, e.g. speedPanels. Those
>> panels should get a trackside attribute (information important because
>> they are physically installed on the track).
>> - <LevelCrossingPanels>
>> - <catenarySectioningPanels>
>> - <currentPanels>
>> - <pantographPanels>
>> - <fixedSignalingChangePanels>
>> - <linePanels>
>> - <gsmPanels>
>> - <radioPanels>
>>
>>
>> New type of element should be added in the standard to describe the
>> panels
>> from the infrastrucure of the railway.
>>
>> [de: Wie koennen variable Signale und fixe Signaltafel in railML
>> unterschieden werden? Eine Liste moeglicher Signaltafeln wurde oben
>> vorgeschlagen.]
>
> yes, you are right. Currently, the panel elements are not modelled within
> the railML infrastructure schema. However, their functionality often
> relates to signals and therefore, panels should be added to the
> <ocsElements> container. Since several types of panels exist, I suggest to
> distinguish between them by using sub-elements. The syntax would be:
>
> <ocsElements>
> <panels>
> <speedPanels>
> <speedPanel id="pan1" />
> ...
> </speedPanels>
> </panels>
> </ocsElements>
>
> These may be panel types we may define:
> - <speedPanels>
> - <etcsPanels>
> - <levelCrossingPanels>
> - <gsmPanels>
> - <catenaryPanels>

> - <signalingSystemChangePanels>
>
> <catenaryPanels> may include the <currentPanels> and <pantographPanels>
> that you suggested, or what is your reason for distuingish between these
> types? Plese specify the content and the usage of <linePanels>.
>
> Because, <panels> will be an optional element within the <ocsElements>
> container, we may implement it with the upcoming railML 2.2. Therefore it
> is important to hear and read the other users' opinions on this topic and
> eventually extend this first approach. Any comments appreciated...

For me it looks better to implement the panels as signals too. So a signal should contain a subelement "signal aspect(s)" within an attribute "switchable" which differes between aspects that are fix (and may be a panel) or aspects which are shown in special situations. This way you can easily change between panels or boards and switchable signal aspects without looking at different positions inside the scheme.
So a reading tool only needs to look into signals and it's subelements signalaspects to find informations given to the driver/engine. In your case a reading tool has to look into panels and signals.
Another thing is kept in the grouping of signal aspects. This might only be useful if anybody defines it for RailML(R). Anything else might be a wild mixture of positioning signal aspects into different groups as seen from the software creator and not from a railway scope.

Best regards.

Carsten

Subject: Re: Signals vs panels
Posted by Christian Rahmig on Sat, 08 Sep 2012 08:21:57 GMT
View Forum Message <> Reply to Message

Hello Carsten,

> For me it looks better to implement the panels as signals too. So a signal
> should contain a subelement "signal aspect(s)" within an attribute
> "switchable" which differes between aspects that are fix (and may be a
> panel) or aspects which are shown in special situations. This way you can
> easily change between panels or boards and switchable signal aspects without
> looking at different positions inside the scheme.
> So a reading tool only needs to look into signals and it's subelements
> signalaspects to find informations given to the driver/engine. In your case
> a reading tool has to look into panels and signals.
> Another thing is kept in the grouping of signal aspects. This might only be
> useful if anybody defines it for RailML(R). Anything else might be a wild
> mixture of positioning signal aspects into different groups as seen from the

> software creator and not from a railway scope.

Thank you for sharing your interesting idea of the signal aspects. Let me try to put it in a structured syntax for a simple example of a signal with a panel:

```
<signal id="s01" pos="42.0" type="main">
   <signalAspect id="s01a01" switchable="true" type="block">
   </signalAspect>
   <signalAspect id="s01a02" switchable="false" type="speed">
   </signalAspect>
</signal>
```

Did I understand you correctly? An important question resulting from this modelling brings us back to the initial point of the discussion: How do we model different (functional) types of panels (or signals)? In the example above I packed this information inside the type of the signal aspect. Any comments appreciated...

Regards

--
Christian Rahmig
railML.infrastructure coordinator

Subject: Re: Signals vs panels
Posted by Carsten Weber on Mon, 10 Sep 2012 14:34:11 GMT
View Forum Message <> Reply to Message

Hello Christian,

"Christian Rahmig" <coord@infrastructure.railml.org> schrieb im Newsbeitrag news:k2f9foagl1@sifa.ivi.fhg.de...
> Hello Carsten,
>
>> For me it looks better to implement the panels as signals too. So a
>> signal
>> should contain a subelement "signal aspect(s)" within an attribute
>> "switchable" which differes between aspects that are fix (and may be a
>> panel) or aspects which are shown in special situations. This way you can
>> easily change between panels or boards and switchable signal aspects
>> without
>> looking at different positions inside the scheme.
>> So a reading tool only needs to look into signals and it's subelements
>> signalaspects to find informations given to the driver/engine. In your
>> case
>> a reading tool has to look into panels and signals.

> Thank you for sharing your interesting idea of the signal aspects. Let me
> try to put it in a structured syntax for a simple example of a signal with
> a panel:
>
> <signal id="s01" pos="42.0" type="main">
> <signalAspect id="s01a01" switchable="true" type="block">
> </signalAspect>
> <signalAspect id="s01a02" switchable="false" type="speed">
> </signalAspect>
> </signal>
>
> Did I understand you correctly?

Nearly. No it looks the way I would prefer.

An important question resulting from
> this modelling brings us back to the initial point of the discussion: How
> do we model different (functional) types of panels (or signals)? In the
> example above I packed this information inside the type of the signal
> aspect. Any comments appreciated...

I think there should be an extension like:

```
<signal id="s01" pos="42.0" operationalType="main" visibleAtTrack="yes">
  <signalInformation id="s01a010" switchable="true" block="stop"/>
  <signalInformation id="s01a011" switchable="true" block="approach"/>
  <signalInformation id="s01a012" switchable="true" block="clear"/>
  <signalInformation id="s01a020" switchable="true" speed="40"/>
  <signalInformation id="s01a021" switchable="true" speed="60"/>
  <signalInformation id="s01a022" switchable="true" speed="80"/>
  <signalInformation id="s01a023" switchable="true" speed="100"/>
  <signalInformation id="s01a030" switchable="true" direction="F"/>
  <signalInformation id="s01a031" switchable="true" direction="R"/>
   ...
</signal>
```

This makes it easy to handle the informations especially for timetable tools (which might be the largest group in using RailML). You create a list of possible aspects which can be given from this signal. And you can create a reference from the train parts ocp sequence to this signal aspect to define which speed is calculated for this slot at this signal. Another thing might be to create the release point of this speed restriction.

In case of a certain signal it could be done like this:

```
<signal id="s01" pos="42.0" operationalType="main" visibleAtTrack="yes"
name="A12">
```

```
  <signalInformation id="s01a010" signalAspect="Hl13" switchable="true"
block="stop"/>
  <signalInformation id="s01a011" signalAspect="Hl10" switchable="true"
block="approach"/>
  <signalInformation id="s01a012" signalAspect="Hl1" switchable="true"
block="clear"/>
  <signalInformation id="s01a020" signalAspect="Hl2" switchable="true"
speed="100" block="clear"/>
  <signalInformation id="s01a021" signalAspect="Hl3a" switchable="true"
speed="40" block="clear"/>
   ...
</signal>
```

Maybe the block attribute should be changed to numberOfFreeBlocksInRegularBreakingDistance (or a little bit shorter) because some signal systems give you more information than two block distances behind the current signal. It might have been in Poland I think. It might also be shortened as a second approach type e.g. "approach2".

Depending on the details you want to bring into RailML you might need a third level of modelling signals. It is positioned between signal and signalInformation and groups the signal informations into a "frame" (or something like this).
So a signal of KS might be modelled like this:

```
<signal id="s01" pos="42.0" operationalType="main" visibleAtTrack="yes"
name="A12">
   <signalFrames>
      <signalFrame name="main">
         <signalInformations>
            <signalInformation id="s01a010" signalAspect="Hp0"
switchable="true" block="stop" runType="all"/>
            <signalInformation id="s01a011" signalAspect="Ks1"
switchable="true" block="approach" runType="train"/>
            <signalInformation id="s01a012" signalAspect="Ks2"
switchable="true" block="clear" runType="train"/>
            <signalInformation id="s01a013" signalAspect="Zs1"
switchable="true" speed="45" runType="train"/>
            <signalInformation id="s01a014" signalAspect="Sh1"
switchable="true" speed="25" runType="shunting"/>
         </signalInformations>
      </signalFrame>
      <signalFrame name="additional">
         <signalInformations>
            <signalInformation id="s01a010" signalAspect="Zs3"
switchable="false" speed="80"/>
         </signlInformations>
      </signalFrame>
```

```
      <signalFrame name="additional">
         <signalInformations>
            <signalInformation id="s01a010" signalAspect="Zs2"
switchable="false" direction="S"/>
         <signalInformations>
      </signalFrame>
   </signalFrames>
</signal>
```

I hope it gets clear what would be done with this level inside of the signal modeling. This opens also the way to create a model of fixing the signals around the tracks.

Best regards.

Carsten

Subject: Re: Signals vs panels
Posted by Susanne Wunsch railML on Tue, 11 Sep 2012 10:25:58 GMT
View Forum Message <> Reply to Message

Hello Carsten and Christian and all others,

"Carsten Weber" <weber@irfp.de> writes:
> "Christian Rahmig" <coord@infrastructure.railml.org> schrieb im Newsbeitrag
> news:k2f9foagl1@sifa.ivi.fhg.de...

>>> For me it looks better to implement the panels as signals too. So a
>>> signal should contain a subelement "signal aspect(s)" within an
>>> attribute "switchable" which differes between aspects that are fix
>>> (and may be a panel) or aspects which are shown in special
>>> situations. This way you can easily change between panels or boards
>>> and switchable signal aspects without looking at different positions
>>> inside the scheme. So a reading tool only needs to look into signals
>>> and it's subelements signalaspects to find informations given to the
>>> driver/engine. In your case a reading tool has to look into panels
>>> and signals.

>> An important question resulting from this modelling brings us back to
>> the initial point of the discussion: How do we model different
>> (functional) types of panels (or signals)? In the example above I
>> packed this information inside the type of the signal aspect.

I like the idea to aggregate panels and signals. There are lots of aspects that are sometimes shown with fixed panels and sometimes with light signals but each time have the same influence on operation.

For offering special attributes in the appropriate context we should group the signals into functional categories like catenarySignals, speedSignals, levelCrossingSignals, radioSignals... Pierre Simon already gave a starting point for this approach, I mean.

For light signal aspects we need an attribute like "switchable" - I agree.

[interlocking related signal attributes snipped]

> Depending on the details you want to bring into RailML you might need
> a third level of modelling signals. It is positioned between signal
> and signalInformation and groups the signal informations into a
> "frame" (or something like this).

One approach is the construction based. Another would be from the GIS-related world: Define the signals by its types and function-based attributes without saying how they look like and how they are joined on the same pole (mast?). Then define the construction (assembling?) like pole, hight, distance from track and refer the signals there. This is just a thought and I would be interested in the pros and cons of both approaches and totally different approaches.

I currently don't like to go to deep into the functionality of interlocking related signals but to think more about a general approach for defining special signals and panels as asked by Pierre Simon.

Any comments, questions, ideas appreciated...

Kind regards...
Susanne

--
Susanne Wunsch
Schema Coordinator: railML.common

Subject: Re: Signals vs panels
Posted by Carsten Weber on Mon, 17 Sep 2012 18:14:18 GMT
View Forum Message <> Reply to Message

"Susanne Wunsch" <coord@common.railml.org> schrieb im Newsbeitrag news:m1ipbkzwi1.fsf@cygnus.heep.sax.de...
> "Carsten Weber" <weber@irfp.de> writes:
>
> I like the idea to aggregate panels and signals. There are lots of
> aspects that are sometimes shown with fixed panels and sometimes with
> light signals but each time have the same influence on operation.

>
So we should not differ the way of signaling in such a high level.

> For offering special attributes in the appropriate context we should
> group the signals into functional categories like catenarySignals,
> speedSignals, levelCrossingSignals, radioSignals... Pierre Simon already
> gave a starting point for this approach, I mean.
>
I hope I wrote it some days before: Some signal aspects belong to both groups. So what to do? Who needs this groups (for what)? Who brings the different aspects into predefined groups?
So in my eyes it would not be good to group signal aspects. If you import signal aspects you do not know .. maybe you do not need them and can skip them.

> For light signal aspects we need an attribute like "switchable" - I
> agree.
>
This way we do not need to differ between "signals" and panels. The attribute "switchable" tells us if the aspect is shown as a fixed sign at a panel. Maybe we can call it also "panel" (yes/no)

> [interlocking related signal attributes snipped]
>
>> Depending on the details you want to bring into RailML you might need
>> a third level of modelling signals. It is positioned between signal
>> and signalInformation and groups the signal informations into a
>> "frame" (or something like this).
>
Just some hints at my answer to my own statement.

> One approach is the construction based. Another would be from the
> GIS-related world: Define the signals by its types and function-based
> attributes without saying how they look like and how they are joined on
> the same pole (mast?). Then define the construction (assembling?) like
> pole, hight, distance from track and refer the signals there. This is
> just a thought and I would be interested in the pros and cons of both
> approaches and totally different approaches.
>
I can remember long discussions about types of signals at another modeling group.

> I currently don't like to go to deep into the functionality of
> interlocking related signals but to think more about a general approach
> for defining special signals and panels as asked by Pierre Simon.

Of course. But we should avoid redefining this if we know the problems before defining the xml-model of a signal inside of RailML.

Best regards.

Carsten

Subject: Re: Signals vs panels
Posted by Carsten Weber on Mon, 17 Sep 2012 18:21:29 GMT
View Forum Message <> Reply to Message

"Carsten Weber" <weber@irfp.de> schrieb im Newsbeitrag
news:k2l835$c06$1@sifa.ivi.fhg.de...

> Depending on the details you want to bring into RailML you might need a
> third level of modelling signals. It is positioned between signal and
> signalInformation and groups the signal informations into a "frame" (or
> something like this).
> So a signal of KS might be modelled like this:
>
> <signal id="s01" pos="42.0" operationalType="main" visibleAtTrack="yes"
> name="A12">
> <signalFrames>
> <signalFrame name="main">
> <signalInformations>
> <signalInformation id="s01a010" signalAspect="Hp0"
> switchable="true" block="stop" runType="all"/>
> <signalInformation id="s01a011" signalAspect="Ks1"
> switchable="true" block="approach" runType="train"/>
> <signalInformation id="s01a012" signalAspect="Ks2"
> switchable="true" block="clear" runType="train"/>
> <signalInformation id="s01a013" signalAspect="Zs1"
> switchable="true" speed="45" runType="train"/>
> <signalInformation id="s01a014" signalAspect="Sh1"
> switchable="true" speed="25" runType="shunting"/>
> </signalInformations>
> </signalFrame>
> <signalFrame name="additional">
> <signalInformations>
> <signalInformation id="s01a010" signalAspect="Zs3"
> switchable="false" speed="80"/>
> </signlInformations>
> </signalFrame>
> <signalFrame name="additional">
> <signalInformations>
> <signalInformation id="s01a010" signalAspect="Zs2"
> switchable="false" direction="S"/>
> <signalInformations>
> </signalFrame>

> </signalFrames>
> </signal>
>

Two options about this way of modeling:
- If you do not know the order of "signalInformations" at the "signalFrames" you can also bring all "signalInformations" in one "signalFrame". The real order is only important for them who build or maintain the signals.
- The "signalFrames" could also be created outside of the signals as an own element. So you can have optional references to them from the "signalInformation". But I think it is not easier to handle this in a case you need to know the right positioning of signalInformation to the frames.

Best regards.

Carsten

Subject: Re: Signals vs panels
Posted by Christian Rahmig on Wed, 24 Oct 2012 10:48:50 GMT
View Forum Message <> Reply to Message

Dear Carsten and other railML users,

> An important question resulting from
>> this modelling brings us back to the initial point of the discussion: How
>> do we model different (functional) types of panels (or signals)? In the
>> example above I packed this information inside the type of the signal
>> aspect. Any comments appreciated...
>
> I think there should be an extension like:
>
> <signal id="s01" pos="42.0" operationalType="main" visibleAtTrack="yes">
> <signalInformation id="s01a010" switchable="true" block="stop"/>
> <signalInformation id="s01a011" switchable="true" block="approach"/>
> <signalInformation id="s01a012" switchable="true" block="clear"/>
> <signalInformation id="s01a020" switchable="true" speed="40"/>
> <signalInformation id="s01a021" switchable="true" speed="60"/>
> <signalInformation id="s01a022" switchable="true" speed="80"/>
> <signalInformation id="s01a023" switchable="true" speed="100"/>
> <signalInformation id="s01a030" switchable="true" direction="F"/>
> <signalInformation id="s01a031" switchable="true" direction="R"/>
> ...
> </signal>

in your example you put the defined signal information in the <signal> element. Like Susanne, I am not a big supporter of this solution,

because it is more related to the interlocking point of view for the signal. For the infrastructure it is important to define the signal as a physical unit, which is situated (with orientation) next to the track. Therefore I would like to limit the additional signal information to the various <signalAspect> elements referencing categories of signal information (like "speed" or "etcs" or "catenary"), which are shown by this signal.

I created a trac ticket [1], which summarizes the basic idea of the signal/panel enhancement.

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

Regards

--
Christian Rahmig
railML.infrastructure coordinator

Subject: Re: Signals vs panels
Posted by Christian Rahmig on Sat, 27 Oct 2012 08:20:02 GMT
View Forum Message <> Reply to Message

Dear Susanne and other railML friends,

> [interlocking related signal attributes snipped]
>
>> Depending on the details you want to bring into RailML you might need
>> a third level of modelling signals. It is positioned between signal
>> and signalInformation and groups the signal informations into a
>> "frame" (or something like this).
>
> One approach is the construction based. Another would be from the
> GIS-related world: Define the signals by its types and function-based
> attributes without saying how they look like and how they are joined on
> the same pole (mast?). Then define the construction (assembling?) like
> pole, hight, distance from track and refer the signals there. This is
> just a thought and I would be interested in the pros and cons of both
> approaches and totally different approaches.

considering the discussion in this and related forum threads, I think that the GIS-related solution of separating the signals' physical assembling and their operational functionality is the best way of implementation. Although this approach sounds like railML 3.0, I want to give an idea of that separation regarding some of the discussed parameters:

[signal construction]

- position along the track
- coodinates
- type of assembling (pole, bridge...)
- height

[operational signal]
- reference to signal construction
- signalling system
- name of the signal (e.g. A12)
- signal frames (main, additional...) containing several signal information elements
- signal information:
    - switcheable vs. panel
    - the shown signal aspect (e.g. speed=40)
    - valid for train movements (train, shunting...)
    - reference to infrastructure element (e.g. level crossing)

The original post opening this thread asked for a more detailed definition of various types of signals, whereas the word "type" refers to the operational view of a signal. As already mentioned by Carsten, it is sometimes difficult to exactly define a signal type on a macroscopic scale. At the level of "signal information" it should always be possible.

So, to summarize the problem: we either want to exactly distinguish between different types of signals, which somehow require to define operational signals, or we allow for interpretations of the (macroscopic) signal types in combination with the current signal implementation? What's your opinion, Susanne, Carsten, Pierre?

Regards

--
Christian Rahmig
railML.infrastructure coordinator