Subject: More detailed 'speed change' definitions
Posted by Susanne Wunsch railML on Fri, 18 Nov 2011 09:29:18 GMT
View Forum Message <> Reply to Message

Hello,

During our last meeting in Karlsruhe there raised the need for more detailed speed change definitions.

There is already a comprehensive proposal described in a Trac ticket. [1] It is currently not implemented because of its non-downward-compatible changes. It waits for the next major release.

> The proposed change enables multiple speed profiles per track with its own characterized restrictions regardless the 'train categories' running there. You may define speed profiles for different axle loads, loading gauges, active train protection systems, active tilting systems but also temporary speed restrictions with its temporal and geographical extent.
>
> Each train part may refer to the speed changes it is calculated for. The infrastructure branch may contain even more speed profiles than a train part needs. That's the same case as with the ocps defined in the infrastructure branch.

Please have a look at the proposal in the Trac ticket if you are interested in this enhancement. It seems that we may implement it any time soon.

Any hints, questions, remarks, comments, +/-1, welcome. :-)

Kind regards...
Susanne

[1] http://trac.assembla.com/railML/ticket/41
--
Susanne Wunsch
Schema Coordinator: railML.common

Subject: Re: More detailed 'speed change' definitions
Posted by Christian Rahmig on Mon, 19 Dec 2011 20:01:11 GMT
View Forum Message <> Reply to Message

Hello Susanne,

thank you for bringing up the topic of the improved train speed classification. Indeed, the proposal described in the Trac ticket #41 is

already very detailed and well structured. Therefore we should implement it with version 2.2. Due to compatibility, this is only possible if the speedProfiles are optional elements and speedChanges do not necessarily have to refer to these speedProfiles. The attribute "maxSpeed" that is used in the Trac ticket description needs to be replaced by the already existing "vMax", which is a required attribute. The current attribute "trainCategory" will be marked as deprecated.

Further comments appreciated.

Kind regards

Christian Rahmig
railML.infrastructure coordinator

Subject: Re: More detailed 'speed change' definitions
Posted by Carsten Weber on Tue, 20 Dec 2011 22:53:35 GMT
View Forum Message <> Reply to Message

Dear Christian,

I have prepared a suggestion to bring in (most) of the required changes as described in ticket #41.

Here is "my" source code to be filled in:

changes in infrastructure.xsd:

```
<xs:complexType name="eSpeedProfile">
  <xs:sequence>
   <xs:element name="speedProfile" type="rail:tSpeedProfile"
maxOccurs="unbounded"/>
  </xs:sequence>
 </xs:complexType>
 <xs:complexType name="tSpeedProfile">
  <xs:sequence>
   <xs:element name="id"/>
   <xs:element name="name" minOccurs="0" maxOccurs="1"/>
   <xs:element name="direction">
    <xs:annotation>
     <xs:documentation>enumeration: inTrackDirection;
oppositeTrackDirection; bothDirections look at
tDirValidity</xs:documentation>
    </xs:annotation>
   </xs:element>
   <xs:element name="influence" minOccurs="0" maxOccurs="1">
    <xs:annotation>
```

```
     <xs:documentation>enumeration: increase; decrease</xs:documentation>
    </xs:annotation>
   </xs:element>
   <xs:element name="axleLoad" minOccurs="0" maxOccurs="1">
    <xs:annotation>
     <xs:documentation>If any vehicle at the trains has a load at an axle
higher than this value this speed limit has to be used.</xs:documentation>
    </xs:annotation>
   </xs:element>
   <xs:element name="weightPerLength" minOccurs="0" maxOccurs="1">
    <xs:annotation>
     <xs:documentation>Maybe it has to be defined which length unit is
meant.</xs:documentation>
    </xs:annotation>
   </xs:element>
   <xs:element name="operatingPeriodRef" minOccurs="0" maxOccurs="1">
    <xs:annotation>
     <xs:documentation>Reference to an operating periode which has to be
defined in timetable element.</xs:documentation>
    </xs:annotation>
   </xs:element>
   <xs:element name="tiltingAngle" minOccurs="0" maxOccurs="1">
    <xs:annotation>
     <xs:documentation>Tilting parameters for which this speed profile is
calculated.</xs:documentation>
    </xs:annotation>
   </xs:element>
   <xs:element name="tiltingSpeed" minOccurs="0" maxOccurs="1">
    <xs:annotation>
     <xs:documentation>Tilting parameters for which this speed profile is
calculated.</xs:documentation>
    </xs:annotation>
   </xs:element>
   <xs:element name="monitoringSystems" type="rail:eMonitoringSystems"
minOccurs="0">
    <xs:annotation>
     <xs:documentation>One of the listed monitoring systems has to be used
by the trainPartSequence to use this speed profile.</xs:documentation>
    </xs:annotation>
   </xs:element>
  </xs:sequence>
 </xs:complexType>
 <xs:complexType name="eMonitoringSystems">
  <xs:sequence>
   <xs:element name="MonitoringSystem" type="rail:tNationalSytemsType"
minOccurs="1" maxOccurs="unbounded">
    <xs:annotation>
     <xs:documentation>type = tNationalSystemsType; Maybe it should be
```

changed to a reference to a new base element monitoring system which would be referenced by vehicles, trainPartSequences, speedProfiles and trackElements. But also the defined types can be used as a key.</xs:documentation>
    </xs:annotation>
   </xs:element>
  </xs:sequence>
 </xs:complexType>

and the changes in infrastructureTypes.xsd would be as follows:

```
 <xs:attributeGroup name="aSpeed">
  <xs:attribute name="trainCategory" type="rail:tTrainCategory">
   <xs:annotation>
    <xs:documentation>deprecated</xs:documentation>
   </xs:annotation>
  </xs:attribute>
  <xs:attribute name="status" type="xs:string">
   <xs:annotation>
    <xs:documentation>deprecated</xs:documentation>
   </xs:annotation>
  </xs:attribute>
  <xs:attribute name="profileRef">
   <xs:annotation>
    <xs:documentation>Refernce to the speed profile.</xs:documentation>
   </xs:annotation>
  </xs:attribute>
  <xs:attribute name="trainRelation" use="optional">
   <xs:annotation>
    <xs:documentation>enumeration: headOfTrain; midOfTrain;
endOfTrain</xs:documentation>
   </xs:annotation>
  </xs:attribute>
  <xs:attribute name="obligationalStop" use="optional" default="false">
   <xs:annotation>
    <xs:documentation>trainRelation defines the position of the train which
has to stop here. default="false"</xs:documentation>
   </xs:annotation>
  </xs:attribute>
  <xs:attribute name="vMax" type="rail:tSpeedKmPerHour" use="required"/>
 </xs:attributeGroup>
```

To be used we need an option to do references from the trainPartSequence(s) to these speed lists(!).

To be discussed ...

Best regards.

Carsten

"Christian Rahmig" <coord@infrastructure.railml.org> schrieb im Newsbeitrag
news:jcoa0e$jh1$1@sifa.ivi.fhg.de...
> Hello Susanne,
>
> thank you for bringing up the topic of the improved train speed
> classification. Indeed, the proposal described in the Trac ticket #41 is
> already very detailed and well structured. Therefore we should implement
> it with version 2.2. Due to compatibility, this is only possible if the
> speedProfiles are optional elements and speedChanges do not necessarily
> have to refer to these speedProfiles. The attribute "maxSpeed" that is
> used in the Trac ticket description needs to be replaced by the already
> existing "vMax", which is a required attribute. The current attribute
> "trainCategory" will be marked as deprecated.
>
> Further comments appreciated.
>
> Kind regards
>
> ---
> Christian Rahmig
> railML.infrastructure coordinator

Subject: Re: More detailed 'speed change' definitions
Posted by Susanne Wunsch railML on Thu, 05 Jan 2012 11:40:52 GMT
View Forum Message <> Reply to Message

Dear Carsten and Christian and all interested in this thread,

wish you all a Happy new year and all the best for your business and
family. :-)

Sorry for the longish response. ;-) I hope, somebody helps out with the
open questions. It is an important development step for railML, I think.

"Carsten Weber" <weber@irfp.de> writes:

> I have prepared a suggestion to bring in (most) of the required changes as
> described in ticket #41.

Thanks for your suggestion, that is a good basis to work on.

I'm sorry for the late follow-up, I was very busy around Christmas and New
Year. (Maybe Christian has similar reasons.)

> Here is "my" source code to be filled in:
>
> changes in infrastructure.xsd:

I will comment your suggestion inline and hope that it stays readable.

> <xs:complexType name="eSpeedProfile">
> <xs:sequence>
> <xs:element name="speedProfile" type="rail:tSpeedProfile"
> maxOccurs="unbounded"/>
> </xs:sequence>
> </xs:complexType>

I assume, that there should be a container element <speedProfiles> for
it, that should be a direct child of <infrastructure>.

<infrastructure>
<infraAttrGroups>...
<tracks>...
<trackGroups>...
<operationControlPoints>...
<controllers>...
<speedProfiles>
</infrastructure>

This container element allows for a xs:sequence of one or more
<speedProfile> elements.

> <xs:complexType name="tSpeedProfile">
> <xs:sequence>
> <xs:element name="id"/>
> <xs:element name="name" minOccurs="0" maxOccurs="1"/>

The "railML way" of modelling these data is using attributes. This type
could be based on the generic railML type "tElementWithIdAndName" which
offers the attributes "id", "name", "code", "description" and "xml:lang".

> <xs:element name="direction">
> <xs:annotation>
> <xs:documentation>enumeration: inTrackDirection;
> oppositeTrackDirection; bothDirections look at
> tDirValidity</xs:documentation>
> </xs:annotation>
> </xs:element>

How about defining the direction of a speed aspect in its
<speedChange> element along the <track>?

Suppose you define some restricted speed aspects for a construction area. You may define one <speedProfile> and refer to it from within various affected <track>s. Their <speedChange> elements contain this reference and the effective direction of the restricted speed aspect.

> <xs:element name="influence" minOccurs="0" maxOccurs="1">
> <xs:annotation>
> <xs:documentation>enumeration: increase; decrease</xs:documentation>
> </xs:annotation>
> </xs:element>

Why do we need this information as an element?

The "railML way" of modelling this kind of information is using an attribute.

The attribute "influence" should be restricted to both enumeration values not allowing anything else. Or do we need "unknown", too?

> <xs:element name="axleLoad" minOccurs="0" maxOccurs="1">
> <xs:annotation>
> <xs:documentation>If any vehicle at the trains has a load at an axle
> higher than this value this speed limit has to be used.</xs:documentation>
> </xs:annotation>
> </xs:element>

Good idea. I would propose it as an attribute and bind it to the railML type "tWeightTons".

What is Jörg von Lingens opinion (as Rollingstock coordinator) about the terminus? Rollingstock already uses "axleLoad" and "axleWeight" for some related information.

> <xs:element name="weightPerLength" minOccurs="0" maxOccurs="1">
> <xs:annotation>
> <xs:documentation>Maybe it has to be defined which length unit is
> meant.</xs:documentation>
> </xs:annotation>
> </xs:element>

Good idea. I would propose it as an attribute "meterLoad" and bind it to the railML type "tMeterloadTonsPerMeter".

> <xs:element name="operatingPeriodRef" minOccurs="0" maxOccurs="1">
> <xs:annotation>
> <xs:documentation>Reference to an operating periode which has to be
> defined in timetable element.</xs:documentation>
> </xs:annotation>

> </xs:element>

I would propose it as an attribute and bind it to the railML type "tGenericRef". It should get an xs:keyref definition for constraining the usage.

If somebody exports only an <infrastructure> with temporal constraints he needs a <timetable> element with at minimum one <timetablePeriod> and <operatingPeriod> element.

> <xs:element name="tiltingAngle" minOccurs="0" maxOccurs="1">
> <xs:annotation>
> <xs:documentation>Tilting parameters for which this speed profile is
> calculated.</xs:documentation>
> </xs:annotation>
> </xs:element>

Good idea. I would spend a child element <tilting> for this and the following kind of information. The attribute "angle" could be bound to the railML type "tAngleDegQuadrant" for allowing 0° to 90°.

Does a speedProfile covers only one tilting angle value or a range of values or some single values?

> <xs:element name="tiltingSpeed" minOccurs="0" maxOccurs="1">
> <xs:annotation>
> <xs:documentation>Tilting parameters for which this speed profile is
> calculated.</xs:documentation>
> </xs:annotation>
> </xs:element>

The terminus "speed" may be a bit misleading. I suppose, that is not related to the "train speed" but to the "rate/speed of tilting", that means the value of tilting degrees per second. I would call this attribute "rate". Are there any other ideas?

This attribute may be bound to the railML type "tSpeedDegreesPerSecond".

There is another kind of information related to the tilting that comes to my mind: the method of tilting. It could go into an attribute "method" that is bound to an enumeration of "active", "passive", "rollCompensation", "unknown", "other:anything".

There is already a type "tTilting" in the rollingstock subschema. We should coordinate these tilting issues with Jörg von Lingen (as rollingstock coordinator).

> <xs:element name="monitoringSystems" type="rail:eMonitoringSystems"

> minOccurs="0">
> <xs:annotation>
> <xs:documentation>One of the listed monitoring systems has to be used
> by the trainPartSequence to use this speed profile.</xs:documentation>
> </xs:annotation>
> </xs:element>
> </xs:sequence>
> </xs:complexType>
> <xs:complexType name="eMonitoringSystems">
> <xs:sequence>
> <xs:element name="MonitoringSystem" type="rail:tNationalSytemsType"
> minOccurs="1" maxOccurs="unbounded">
> <xs:annotation>
> <xs:documentation>type = tNationalSystemsType; Maybe it should be
> changed to a reference to a new base element monitoring system which would
> be referenced by vehicles, trainPartSequences, speedProfiles and
> trackElements. But also the defined types can be used as a
> key.</xs:documentation>
> </xs:annotation>
> </xs:element>
> </xs:sequence>
> </xs:complexType>

That is an interesting idea. It touches Trac Ticket #111 [1].

railML currently requires heavy repetition of obviously equal data applying the <trainProtectionElement> in <ocsElements>. Thomas Albrecht already pointed it out in 2009. [2] But a change of this structure requires a non-downward-compatible change, that we may do with next major release according to our release policy. See also Trac ticket #23 [3].

If it would be a blocking issue, we may mark the current element as "deprecated" and allow for a "definition list"/"library"/"catalog" of train protection elements inside the <infrastructure> element.

<trainProtectionSystem id="tpsLZB" name="LZB"
 description="Linienförmige Zugbeeinflussung" xml:lang="de"
 type="LZB" medium="inductive" monitoring="continuous"/>

Along the <track> we could define the <trainProtectionElement> refering to one of the "definition list elements".

The same way we could refer to various train protection systems from within a <speedProfile>.

Does this approach also work for different ETCS levels?

This issues should be harmonized with the rollingstock implementation (tNationalSystemsType). Contact: Jörg von Lingen.

Maybe we could solve this issue a bit later apart from the speed profile discussion.

> and the changes in infrastructureTypes.xsd would be as follows:
> <xs:attributeGroup name="aSpeed">
> <xs:attribute name="trainCategory" type="rail:tTrainCategory">
> <xs:annotation>
> <xs:documentation>deprecated</xs:documentation>
> </xs:annotation>
> </xs:attribute>

Good idea. "Define special speed profiles instead."

> <xs:attribute name="status" type="xs:string">
> <xs:annotation>
> <xs:documentation>deprecated</xs:documentation>
> </xs:annotation>
> </xs:attribute>

Good idea. "???"

> <xs:attribute name="profileRef">
> <xs:annotation>
> <xs:documentation>Refernce to the speed profile.</xs:documentation>
> </xs:annotation>
> </xs:attribute>

Good idea.

> <xs:attribute name="trainRelation" use="optional">
> <xs:annotation>
> <xs:documentation>enumeration: headOfTrain; midOfTrain;
> endOfTrain</xs:documentation>
> </xs:annotation>
> </xs:attribute>

How about the terminus "startsAt"?

How about a <speedChange> for "both" directions and "headOfTrain" relation? Does it change its meaning regarding the track definition direction? A <speedChange> for "both" directions is a short-cut railML definition for two distinct speed definition positions each for another driving direction, they are typically a block distance away from each other.

Is this information bound to a speed aspect along the track or
does it additionally differs for various trains?

I'm not sure, if this is the right place for this kind of
information. Who could help out with some more experiences?

> <xs:attribute name="obligationalStop" use="optional" default="false">
> <xs:annotation>
> <xs:documentation>trainRelation defines the position of the train which
> has to stop here. default="false"</xs:documentation>
> </xs:annotation>
> </xs:attribute>

Could you explain the use case, please? Are there any speed aspect
changes where all trains have to stop?

> <xs:attribute name="vMax" type="rail:tSpeedKmPerHour" use="required"/>
> </xs:attributeGroup>

I would like to rename this attribute to "maxSpeed", but that only comes
with next major release. (Trac ticket #108) ;-)

> To be used we need an option to do references from the trainPartSequence(s)
> to these speed lists(!).

The <trainPartSequence> is part of a <train> as the whole "thing" that
is composed of one or many <trainPart>s along a specified
distance. That's a good place for general aspects regarding the whole
"train". We could put one to many <speedProfileRef> elements there.

Currently there is a possibility to define the section between <ocp>s
within the <sectionTT> element. Its <trackRef> element may also contain
new <speedRef> elements for refering to the applying speed
profiles. This idea is shown in Trac ticket #41 [4].

The second approach allows for trains that use a speed profile only for
parts of the "train part sequence" but not for the whole distance,
despite the fact that there are speed changes defined for this speed
profile along the rest of the distance, but these don't influence the
"rest train part sequence".

What do you think about these two approaches? Could we take the easier
and more clear first approach?

Thanks for your suggestions, they are really helpful for further
development. Let's take them up to railML style XML structures for
providing sustainable XML schemas. :-)

Kind regards...
Susanne

[1] http://trac.assembla.com/railML/ticket/111
[2]
http://www.railml.org/web/index.php/previous-events.html?fil
e=tl_files/railML.org/documents/events/slides/2009-10-06_tud resden_albrecht-interlocking.pdf
[3] http://trac.assembla.com/railML/ticket/23
[4] http://trac.assembla.com/railML/ticket/41

--
Susanne Wunsch
Schema Coordinator: railML.common

Subject: Re: More detailed 'speed change' definitions
Posted by Carsten Weber on Mon, 09 Jan 2012 14:35:44 GMT
View Forum Message <> Reply to Message

"Susanne Wunsch" <coord@common.railml.org> schrieb im Newsbeitrag
news:bb2ty4a74xn.fsf@remi.heep.sax.de...
> Dear Carsten and Christian and all interested in this thread,
>
> wish you all a Happy new year and all the best for your business and
> family. :-)
>
Me too.

> Sorry for the longish response. ;-) I hope, somebody helps out with the
> open questions. It is an important development step for railML, I think.
>
> "Carsten Weber" <weber@irfp.de> writes:
>
> I'm sorry for the late follow-up, I was very busy around Christmas and New
> Year. (Maybe Christian has similar reasons.)
>
>> <xs:complexType name="tSpeedProfile">
>> <xs:sequence>
>> <xs:element name="id"/>
>> <xs:element name="name" minOccurs="0" maxOccurs="1"/>
>
> The "railML way" of modelling these data is using attributes. This type
> could be based on the generic railML type "tElementWithIdAndName" which
> offers the attributes "id", "name", "code", "description" and "xml:lang".

Well, as I had written before: It is only a suggestion. It would also be
possible via an attribute too.
>

>>   <xs:element name="direction">
>>   <xs:annotation>
>>   <xs:documentation>enumeration: inTrackDirection;
>> oppositeTrackDirection; bothDirections look at
>> tDirValidity</xs:documentation>
>>   </xs:annotation>
>>   </xs:element>
>
> How about defining the direction of a speed aspect in its
> <speedChange> element along the <track>?
>
What would that mean? Every <speedChange> has to have speed information for
both directions?

> Suppose you define some restricted speed aspects for a construction
> area. You may define one <speedProfile> and refer to it from within
> various affected <track>s. Their <speedChange> elements contain this
> reference and the effective direction of the restricted speed aspect.
>
Well a speedChange would be a child element of a speedProfile. But to bring
this into RailML you need to change structure. The suggestion is based on
the current structure without any change inside.

>>   <xs:element name="influence" minOccurs="0" maxOccurs="1">
>>   <xs:annotation>
>>   <xs:documentation>enumeration: increase; decrease</xs:documentation>
>>   </xs:annotation>
>>   </xs:element>
>
> Why do we need this information as an element?
>
> The "railML way" of modelling this kind of information is using an
> attribute.
>
> The attribute "influence" should be restricted to both enumeration
> values not allowing anything else. Or do we need "unknown", too?
>
Maybe not unknown but "both" but this will bring more trouble then it would
help. It is easy to write the speed information for both in two sections for
increase and decrease.

>>   <xs:element name="axleLoad" minOccurs="0" maxOccurs="1">
>>   <xs:annotation>
>>   <xs:documentation>If any vehicle at the trains has a load at an axle
>> higher than this value this speed limit has to be
>> used.</xs:documentation>
>>   </xs:annotation>
>>   </xs:element>

>
> Good idea. I would propose it as an attribute and bind it to the railML
> type "tWeightTons".
>

> terminus? Rollingstock already uses "axleLoad" and "axleWeight" for some
> related information.
>
>> <xs:element name="weightPerLength" minOccurs="0" maxOccurs="1">
>> <xs:annotation>
>> <xs:documentation>Maybe it has to be defined which length unit is
>> meant.</xs:documentation>
>> </xs:annotation>
>> </xs:element>
>
> Good idea. I would propose it as an attribute "meterLoad" and bind it to
> the railML type "tMeterloadTonsPerMeter".
>
Ok.

>> <xs:element name="operatingPeriodRef" minOccurs="0" maxOccurs="1">
>> <xs:annotation>
>> <xs:documentation>Reference to an operating periode which has to be
>> defined in timetable element.</xs:documentation>
>> </xs:annotation>
>> </xs:element>
>
> I would propose it as an attribute and bind it to the railML type
> "tGenericRef". It should get an xs:keyref definition for constraining
> the usage.
>
> If somebody exports only an <infrastructure> with temporal constraints
> he needs a <timetable> element with at minimum one <timetablePeriod> and
> <operatingPeriod> element.
>
Yes.

>> <xs:element name="tiltingAngle" minOccurs="0" maxOccurs="1">
>> <xs:annotation>
>> <xs:documentation>Tilting parameters for which this speed profile is
>> calculated.</xs:documentation>
>> </xs:annotation>
>> </xs:element>
>
> Good idea. I would spend a child element <tilting> for this and the
> following kind of information. The attribute "angle" could be bound to

>

> Does a speedProfile covers only one tilting angle value or a range of
> values or some single values?
>
It is a question of definition. In my sight this value means: your vehicle
has to be able to reach a tilting angle of ... (or more).

>> <xs:element name="tiltingSpeed" minOccurs="0" maxOccurs="1">
>> <xs:annotation>
>> <xs:documentation>Tilting parameters for which this speed profile is
>> calculated.</xs:documentation>
>> </xs:annotation>
>> </xs:element>
>
> The terminus "speed" may be a bit misleading. I suppose, that is not
> related to the "train speed" but to the "rate/speed of tilting", that
> means the value of tilting degrees per second. I would call this
> attribute "rate". Are there any other ideas?
>
> This attribute may be bound to the railML type "tSpeedDegreesPerSecond".
>
> There is another kind of information related to the tilting that comes
> to my mind: the method of tilting. It could go into an attribute
> "method" that is bound to an enumeration of "active", "passive",
> "rollCompensation", "unknown", "other:anything".
>
I know this. But is it really important here? Do I need to know which way
the vehicle tilts?

> There is already a type "tTilting" in the rollingstock subschema. We

> rollingstock coordinator).
>
>> <xs:element name="monitoringSystems" type="rail:eMonitoringSystems"
>> minOccurs="0">
>> <xs:annotation>
>> <xs:documentation>One of the listed monitoring systems has to be
>> used
>> by the trainPartSequence to use this speed profile.</xs:documentation>
>> </xs:annotation>
>> </xs:element>
>> </xs:sequence>
>> </xs:complexType>
>> <xs:complexType name="eMonitoringSystems">
>> <xs:sequence>
>> <xs:element name="MonitoringSystem" type="rail:tNationalSytemsType"
>> minOccurs="1" maxOccurs="unbounded">
>> <xs:annotation>
>> <xs:documentation>type = tNationalSystemsType; Maybe it should be

>> changed to a reference to a new base element monitoring system which
>> would
>> be referenced by vehicles, trainPartSequences, speedProfiles and
>> trackElements. But also the defined types can be used as a
>> key.</xs:documentation>
>> </xs:annotation>
>> </xs:element>
>> </xs:sequence>
>> </xs:complexType>
>
> That is an interesting idea. It touches Trac Ticket #111 [1].
>
> railML currently requires heavy repetition of obviously equal data
> applying the <trainProtectionElement> in <ocsElements>. Thomas Albrecht
> already pointed it out in 2009. [2] But a change of this structure
> requires a non-downward-compatible change, that we may do with next
> major release according to our release policy. See also Trac ticket #23
> [3].
>
> If it would be a blocking issue, we may mark the current element as
> "deprecated" and allow for a "definition list"/"library"/"catalog" of
> train protection elements inside the <infrastructure> element.
>
> <trainProtectionSystem id="tpsLZB" name="LZB"

> type="LZB" medium="inductive" monitoring="continuous"/>
>
> Along the <track> we could define the <trainProtectionElement>
> refering to one of the "definition list elements".
>
> The same way we could refer to various train protection systems from
> within a <speedProfile>.
>
> Does this approach also work for different ETCS levels?
>
> This issues should be harmonized with the rollingstock implementation

>
> Maybe we could solve this issue a bit later apart from the speed
> profile discussion.
>
Ok.

>> and the changes in infrastructureTypes.xsd would be as follows:
>> <xs:attributeGroup name="aSpeed">
>> <xs:attribute name="trainCategory" type="rail:tTrainCategory">
>> <xs:annotation>
>> <xs:documentation>deprecated</xs:documentation>

>>   </xs:annotation>
>>   </xs:attribute>
>
> Good idea. "Define special speed profiles instead."
>
>>   <xs:attribute name="status" type="xs:string">
>>   <xs:annotation>
>>   <xs:documentation>deprecated</xs:documentation>
>>   </xs:annotation>
>>   </xs:attribute>
>
> Good idea. "???"
>
>>   <xs:attribute name="profileRef">
>>   <xs:annotation>
>>   <xs:documentation>Refernce to the speed profile.</xs:documentation>
>>   </xs:annotation>
>>   </xs:attribute>
>
> Good idea.
>
>>   <xs:attribute name="trainRelation" use="optional">
>>   <xs:annotation>
>>   <xs:documentation>enumeration: headOfTrain; midOfTrain;
>> endOfTrain</xs:documentation>
>>   </xs:annotation>
>>   </xs:attribute>
>
> How about the terminus "startsAt"?
>
We can also use "validAt" ...

> How about a <speedChange> for "both" directions and "headOfTrain"
> relation? Does it change its meaning regarding the track definition
> direction? A <speedChange> for "both" directions is a short-cut railML
> definition for two distinct speed definition positions each for another
> driving direction, they are typically a block distance away from each
> other.
>
> Is this information bound to a speed aspect along the track or
> does it additionally differs for various trains?
>
> I'm not sure, if this is the right place for this kind of
> information. Who could help out with some more experiences?
>

I'm sorry. But I can not follow you here. For me it seems better to define
the direction inside of the speed profile and not at every <speedChange>.

>> <xs:attribute name="obligationalStop" use="optional" default="false">
>> <xs:annotation>
>> <xs:documentation>trainRelation defines the position of the train
>> which
>> has to stop here. default="false"</xs:documentation>
>> </xs:annotation>
>> </xs:attribute>
>
> Could you explain the use case, please? Are there any speed aspect
> changes where all trains have to stop?
>
Yes. In this case you would need a second speed information. "Stop here and
go on with ...". This will be a little bit tricky if you would have a speed1
and a speed2-information.
So you only fill in a mark which says: Every train (using this speed
profile) has to stop here. So normally all the trains would have to stop
here.
You can find examples at branch lines at which you have to activate barriers
via key or a laser "pistol" or something like that.

>> <xs:attribute name="vMax" type="rail:tSpeedKmPerHour" use="required"/>
>> </xs:attributeGroup>
>
> I would like to rename this attribute to "maxSpeed", but that only comes
> with next major release. (Trac ticket #108) ;-)
>
Hm. You told me no abbreviations inside RailML?!

>> To be used we need an option to do references from the
>> trainPartSequence(s)
>> to these speed lists(!).
>
> The <trainPartSequence> is part of a <train> as the whole "thing" that
> is composed of one or many <trainPart>s along a specified
> distance. That's a good place for general aspects regarding the whole
> "train". We could put one to many <speedProfileRef> elements there.
>
> Currently there is a possibility to define the section between <ocp>s
> within the <sectionTT> element. Its <trackRef> element may also contain
> new <speedRef> elements for refering to the applying speed
> profiles. This idea is shown in Trac ticket #41 [4].
>
> The second approach allows for trains that use a speed profile only for
> parts of the "train part sequence" but not for the whole distance,
> despite the fact that there are speed changes defined for this speed
> profile along the rest of the distance, but these don't influence the
> "rest train part sequence".

>
> What do you think about these two approaches? Could we take the easier
> and more clear first approach?
>
Well. No one blocks you to build for every <sectionTT> an own
<trainPartSequence> so this can be done with the first approach too. The
other way means that you would have to repeat the reference to the speed
profiles at every section. That would not be usefull.

By the way: We need to define an "endOfSpeedList". So for example if you
have to run slow with a train e.g. above a bridge the allowed speed goes
back to track speed after you left the bridge. So you need an option to say:
the speed profile ends here. Other speed profiles are now in progress. So
you can use speed profiles along the whole train run but they have one one
speed limitation at a short bridge or something like this.

> Thanks for your suggestions, they are really helpful for further
> development. Let's take them up to railML style XML structures for
> providing sustainable XML schemas. :-)
>
Just do it. ;)

Best regards.

Carsten

> Kind regards...
> Susanne
>
> [1] http://trac.assembla.com/railML/ticket/111
> [2]
> http://www.railml.org/web/index.php/previous-events.html?fil
e=tl_files/railML.org/documents/events/slides/2009-10-06_tud resden_albrecht-interlocking.pdf
> [3] http://trac.assembla.com/railML/ticket/23
> [4] http://trac.assembla.com/railML/ticket/41
>
> --
> Susanne Wunsch
> Schema Coordinator: railML.common
>

Subject: Re: More detailed 'speed change' definitions
Posted by Susanne Wunsch railML on Mon, 09 Jan 2012 16:37:05 GMT
View Forum Message <> Reply to Message

Hi Carsten, and all who are interested,

"Carsten Weber" <weber@irfp.de> writes:
> "Susanne Wunsch" <coord@common.railml.org> schrieb im Newsbeitrag
> news:bb2ty4a74xn.fsf@remi.heep.sax.de...
>> "Carsten Weber" <weber@irfp.de> writes:
>>> <xs:element name="direction">
>>> <xs:annotation>
>>> <xs:documentation>enumeration: inTrackDirection;
>>> oppositeTrackDirection; bothDirections look at
>>> tDirValidity</xs:documentation>
>>> </xs:annotation>
>>> </xs:element>
>>
>> How about defining the direction of a speed aspect in its
>> <speedChange> element along the <track>?
>>
> What would that mean? Every <speedChange> has to have speed information for
> both directions?

If your track may be operated in both directions - yes. I try to put a
small example here, hope it helps.

<speedChange pos="10" dir="up" vMax="40"/>
<speedChange pos="10" dir="down" vMax="60"/>
<speedChange pos="200" dir="both" vMax="80"/>

```
pos         10                              200
track dir   ------------------------------------------------------------ >
vMax ->     40->                              80->
vMax <-     <-?                              <-60            <-80
```

The <speedChange> information defines the maximum speed aspect for the
next track section, means up to the next <speedChange> element in the
track definition direction. A <speedChange> for both directions means,
that the speed aspect at the next track section is the same for both
train running directions. It is _not meant_ to be the same speed aspect
from this point in both directions - that is really misleading!

The above definition does not include the newly <profileRef>, that comes
in addition. It shows the current usage of <speedChange> elements. If
your <speedProfile> includes a "direction" attribute, you have to split
the <speedChange> elements with current 'dir="both"' attributes.

Are there any speed profiles that apply only in one direction? If you
have a speed restriction along a bridge, you may define different
<speedChange> elements in each direction refering to the same speed
profile with different speed aspects. The <speedChange> elements already
need the direction attribute. Why do we need to duplicate it?

>> Suppose you define some restricted speed aspects for a construction
>> area. You may define one <speedProfile> and refer to it from within
>> various affected <track>s. Their <speedChange> elements contain this
>> reference and the effective direction of the restricted speed aspect.
>>
> Well a speedChange would be a child element of a speedProfile. But to bring
> this into RailML you need to change structure. The suggestion is based on
> the current structure without any change inside.

No. A <speedProfile> element goes as a grand-child element to the
<infrastructure> element.

```
<infrastructure>
 <infraAttrGroups>...
 <tracks>
  <track id="t1">
   <trackTopology>...
   <trackElements>
    <speedChanges>
     <speedChange id="sc1" profileRef="sp1" vMax="40"/>
    </speedChanges>
   </trackElements>
  </track>
 </tracks>
 <trackGroups>...
 <operationControlPoints>...
 <controllers>...
 <speedProfiles>
  <speedProfile id="sp1" name="constructionArea"/>
 </speedProfiles>
</infrastructure>
```

>>> <xs:element name="influence" minOccurs="0" maxOccurs="1">
>>> <xs:annotation>
>>> <xs:documentation>enumeration: increase; decrease</xs:documentation>
>>> </xs:annotation>
>>> </xs:element>
>>
>> Why do we need this information as an element?
>>
>> The "railML way" of modelling this kind of information is using an
>> attribute.
>>
>> The attribute "influence" should be restricted to both enumeration
>> values not allowing anything else. Or do we need "unknown", too?
>>
> Maybe not unknown but "both" but this will bring more trouble then it would
> help. It is easy to write the speed information for both in two sections for

> increase and decrease.

A <train> may specify multiple speed profiles that may be "increasing"
or "decreasing".

<speedProfiles>
  <speedProfile id="sp1" influence="increasing"/>
  <speedProfile id="sp2" influence="decreasing"/>
</speedProfiles>

<train>
  <trainPartSequence>
    <trainPartRef>...
    <speedProfileRef ref="sp1"/>
    <speedProfileRef ref="sp2"/>
  </trainPartSequence>
</train>

In order to calculate the maximum speed aspect for a train along a track
section you have to follow the speedChange elements along the track
(referenced in the sectionTT) and take only the elements refering to the
specified speed profiles. If you get "overlapping" speed aspect you take
the highest "increasing" minus the deepest "decreasing" speed aspect.

This way you don't need "both". "unknown" would be a fixed value that
can't be used for calculating with other overlapping speed aspects.

>>>   <xs:element name="tiltingAngle" minOccurs="0" maxOccurs="1">
>>>   <xs:annotation>
>>>   <xs:documentation>Tilting parameters for which this speed profile is
>>> calculated.</xs:documentation>
>>>   </xs:annotation>
>>>   </xs:element>
>>
>> Good idea. I would spend a child element <tilting> for this and the
>> following kind of information. The attribute "angle" could be bound to
>> the railML type "tAngleDegQuadrant" for allowing 0° to 90°.
>>
>> Does a speedProfile covers only one tilting angle value or a range of
>> values or some single values?
>>
> It is a question of definition. In my sight this value means: your vehicle
> has to be able to reach a tilting angle of ... (or more).

If your vehicle tilts up to 3.5° it has to pass the curve slower than
at an tilting angle of 8°. Maybe you have a speed aspect for a range of
tilting angle like:

0° .. 3.4° 80 km/h
3.5° .. 5.4° 90 km/h
5.5° .. 8° 100 km/h

I don't know, how the different speed aspects are defined in real
life. Do you know how it is managed in Germany and Switzerland?

>>> <xs:element name="tiltingSpeed" minOccurs="0" maxOccurs="1">
>>> <xs:annotation>
>>> <xs:documentation>Tilting parameters for which this speed profile is
>>> calculated.</xs:documentation>
>>> </xs:annotation>
>>> </xs:element>
>>
>> The terminus "speed" may be a bit misleading. I suppose, that is not
>> related to the "train speed" but to the "rate/speed of tilting", that
>> means the value of tilting degrees per second. I would call this
>> attribute "rate". Are there any other ideas?
>>
>> This attribute may be bound to the railML type "tSpeedDegreesPerSecond".
>>
>> There is another kind of information related to the tilting that comes
>> to my mind: the method of tilting. It could go into an attribute
>> "method" that is bound to an enumeration of "active", "passive",
>> "rollCompensation", "unknown", "other:anything".
>>
> I know this. But is it really important here? Do I need to know which way
> the vehicle tilts?

I thought that it is another type of speed restriction. If you go with
"roll compensation" you may increase your speed, if you support "active
tilting", you have to switch the system off and go with default speed
aspect. This may be a Swiss use case, I mean.

>> How about a <speedChange> for "both" directions and "headOfTrain"
>> relation? Does it change its meaning regarding the track definition
>> direction? A <speedChange> for "both" directions is a short-cut railML
>> definition for two distinct speed definition positions each for another
>> driving direction, they are typically a block distance away from each
>> other.
>>
>> Is this information bound to a speed aspect along the track or
>> does it additionally differs for various trains?
>>
>> I'm not sure, if this is the right place for this kind of
>> information. Who could help out with some more experiences?
>>
>

> I'm sorry. But I can not follow you here. For me it seems better to define
> the direction inside of the speed profile and not at every <speedChange>.

The direction at <speedChange> is currently in use. I try to explain
extending the above example:

<speedChange pos="10" dir="up" vMax="40" trainRelation="headOfTrain"/>
<speedChange pos="10" dir="down" vMax="60" trainRelation="midOfTrain"/>
<speedChange pos="200" dir="both" vMax="80" trainRelation="endOfTrain"/>

```
pos          10                              200
track dir     ------------------------------------------------------- >
vMax ->        40->                             80->
train rel ->  head->                            end->
vMax <-        <-?                             <-60            <-80
train rel <-  <-?                              <-mid           <-end
```

The "trainRelation" attribute applies to the starting point of the "speed
aspect track section".

>>> <xs:attribute name="obligationalStop" use="optional" default="false">
>>> <xs:annotation>
>>> <xs:documentation>trainRelation defines the position of the train
>>> which
>>> has to stop here. default="false"</xs:documentation>
>>> </xs:annotation>
>>> </xs:attribute>
>>
>> Could you explain the use case, please? Are there any speed aspect
>> changes where all trains have to stop?
>>
> Yes. In this case you would need a second speed information. "Stop here and
> go on with ...". This will be a little bit tricky if you would have a speed1
> and a speed2-information.
> So you only fill in a mark which says: Every train (using this speed
> profile) has to stop here. So normally all the trains would have to stop
> here.
> You can find examples at branch lines at which you have to activate barriers
> via key or a laser "pistol" or something like that.

Thanks for the explanation. Maybe we could shorten the name to "stop"?

>>> <xs:attribute name="vMax" type="rail:tSpeedKmPerHour" use="required"/>
>>> </xs:attributeGroup>
>>
>> I would like to rename this attribute to "maxSpeed", but that only comes
>> with next major release. (Trac ticket #108) ;-)
>>

> Hm. You told me no abbreviations inside RailML?!

We allow 'max' for 'maximum' and 'min' for 'minimum'. They are not misleading and widely known.

> By the way: We need to define an "endOfSpeedList". So for example if you
> have to run slow with a train e.g. above a bridge the allowed speed goes
> back to track speed after you left the bridge. So you need an option to say:
> the speed profile ends here. Other speed profiles are now in progress. So
> you can use speed profiles along the whole train run but they have one one
> speed limitation at a short bridge or something like this.

No, that use case is covered by allowing multiple speed profiles that overlap each other. The software export has to ensure that there are valid speed definitions for all sections of the track.

>> Thanks for your suggestions, they are really helpful for further
>> development. Let's take them up to railML style XML structures for
>> providing sustainable XML schemas. :-)
>>
> Just do it. ;)

I think we are on the right way. ;-)

Kind regards...
Susanne

--
Susanne Wunsch
Schema Coordinator: railML.common

Subject: Re: More detailed 'speed change' definitions
Posted by Carsten Weber on Wed, 11 Jan 2012 19:42:10 GMT
View Forum Message <> Reply to Message

"Susanne Wunsch" <coord@common.railml.org> schrieb im Newsbeitrag news:bb28vlg7rym.fsf@remi.heep.sax.de...
> Hi Carsten, and all who are interested,
>
> "Carsten Weber" <weber@irfp.de> writes:
>> "Susanne Wunsch" <coord@common.railml.org> schrieb im Newsbeitrag
>> news:bb2ty4a74xn.fsf@remi.heep.sax.de...
>>> "Carsten Weber" <weber@irfp.de> writes:
>>>> <xs:element name="direction">
>>>> <xs:annotation>
>>>> <xs:documentation>enumeration: inTrackDirection;
>>>> oppositeTrackDirection; bothDirections look at

>>>> tDirValidity</xs:documentation>
>>>>   </xs:annotation>
>>>>   </xs:element>
>>>
>>> How about defining the direction of a speed aspect in its
>>> <speedChange> element along the <track>?
>>>
>> What would that mean? Every <speedChange> has to have speed information
>> for
>> both directions?
>
> If your track may be operated in both directions - yes. I try to put a
> small example here, hope it helps.
>
> <speedChange pos="10" dir="up" vMax="40"/>
> <speedChange pos="10" dir="down" vMax="60"/>
> <speedChange pos="200" dir="both" vMax="80"/>
>
> pos   10   200
> track dir   -- >
> vMax ->   40->   80->
> vMax <-   <-?   <-60   <-80
>
> The <speedChange> information defines the maximum speed aspect for the
> next track section, means up to the next <speedChange> element in the
> track definition direction. A <speedChange> for both directions means,
> that the speed aspect at the next track section is the same for both
> train running directions. It is _not meant_ to be the same speed aspect
> from this point in both directions - that is really misleading!
>
So it becomes tricky to use this structure.

> The above definition does not include the newly <profileRef>, that comes
> in addition. It shows the current usage of <speedChange> elements. If
> your <speedProfile> includes a "direction" attribute, you have to split
> the <speedChange> elements with current 'dir="both"' attributes.
>
> Are there any speed profiles that apply only in one direction?

Yes. For example if you have a railway crossing which is operated by sight
(without any technical system) you may have speed restrictions only for one
direction caused by the sight to the track.

```
         |   |
A============B
   xxxx |   |
```

So you need a speed restriction in direction A to B but not in direction B
to A. Such cases exist.

> If you have a speed restriction along a bridge, you may define different
> <speedChange> elements in each direction refering to the same speed
> profile with different speed aspects. The <speedChange> elements already
> need the direction attribute. Why do we need to duplicate it?
>

Because of more clearance.
In my sight a speedChange should be a child of a speedProfile.

```
<speedProfile direction="up" ...>
   <speedChange position="0" speed="60" .../>
   <speedChange position="5" speed="120" .../>
   <speedChange position="123" speed="100" .../>
   <.../>
</speedProfile>
```

So you can see: the speedChanges if you run the track in one direction in a line.
But this structure requires a break in downward compatibility which is not leagl at the moment.

>>> Suppose you define some restricted speed aspects for a construction
>>> area. You may define one <speedProfile> and refer to it from within
>>> various affected <track>s. Their <speedChange> elements contain this
>>> reference and the effective direction of the restricted speed aspect.
>>>
>> Well a speedChange would be a child element of a speedProfile. But to
>> bring
>> this into RailML you need to change structure. The suggestion is based on
>> the current structure without any change inside.
>
> No. A <speedProfile> element goes as a grand-child element to the
> <infrastructure> element.
>

```
> <infrastructure>
>  <infraAttrGroups>...
>  <tracks>
>    <track id="t1">
>      <trackTopology>...
>      <trackElements>
>        <speedChanges>
>          <speedChange id="sc1" profileRef="sp1" vMax="40"/>
>        </speedChanges>
>      </trackElements>
>    </track>
>  </tracks>
>  <trackGroups>...
>  <operationControlPoints>...
```

> <controllers>...
> <speedProfiles>
> <speedProfile id="sp1" name="constructionArea"/>
> </speedProfiles>
> </infrastructure>
>
As written above. The solution here is usefull for the current scheme.

> In order to calculate the maximum speed aspect for a train along a track
> section you have to follow the speedChange elements along the track
> (referenced in the sectionTT) and take only the elements refering to the
> specified speed profiles. If you get "overlapping" speed aspect you take
> the highest "increasing" minus the deepest "decreasing" speed aspect.
>
> This way you don't need "both". "unknown" would be a fixed value that
> can't be used for calculating with other overlapping speed aspects.
>
Ok. So what should we do if the keyword unknown appears?

>>>> <xs:element name="tiltingAngle" minOccurs="0" maxOccurs="1">
>>>> <xs:annotation>
>>>> <xs:documentation>Tilting parameters for which this speed profile
>>>> is
>>>> calculated.</xs:documentation>
>>>> </xs:annotation>
>>>> </xs:element>
>>>
>>> Good idea. I would spend a child element <tilting> for this and the
>>> following kind of information. The attribute "angle" could be bound to

>>>
>>> Does a speedProfile covers only one tilting angle value or a range of
>>> values or some single values?
>>>
>> It is a question of definition. In my sight this value means: your
>> vehicle
>> has to be able to reach a tilting angle of ... (or more).
>

> tilting angle like:
>

>
Yes. So you would have to define 3 different speed profiles. A speed profile without any data to tiliting or a tilting angle of "0", a second one with

> I don't know, how the different speed aspects are defined in real
> life. Do you know how it is managed in Germany and Switzerland?
>
So far as I know you can transmitt only one speed profile for tilting
vehicles with current known systems.
But it would not be a problem to have several speed lists for different
tilting angles.

>>>> <xs:element name="tiltingSpeed" minOccurs="0" maxOccurs="1">
>>>> <xs:annotation>
>>>> <xs:documentation>Tilting parameters for which this speed profile
>>>> is
>>>> calculated.</xs:documentation>
>>>> </xs:annotation>
>>>> </xs:element>
>>>
>>> The terminus "speed" may be a bit misleading. I suppose, that is not
>>> related to the "train speed" but to the "rate/speed of tilting", that
>>> means the value of tilting degrees per second. I would call this
>>> attribute "rate". Are there any other ideas?
>>>
>>> This attribute may be bound to the railML type "tSpeedDegreesPerSecond".
>>>
>>> There is another kind of information related to the tilting that comes
>>> to my mind: the method of tilting. It could go into an attribute
>>> "method" that is bound to an enumeration of "active", "passive",
>>> "rollCompensation", "unknown", "other:anything".
>>>
>> I know this. But is it really important here? Do I need to know which way
>> the vehicle tilts?
>
> I thought that it is another type of speed restriction. If you go with
> "roll compensation" you may increase your speed, if you support "active
> tilting", you have to switch the system off and go with default speed
> aspect. This may be a Swiss use case, I mean.
>

something like that) and you may have a second one for higher tilting

only run the "slow" profile. The way you reach the tilting angle is not
really important. It is more a question of the tilting speed but it is also
not important there.

>>> How about a <speedChange> for "both" directions and "headOfTrain"
>>> relation? Does it change its meaning regarding the track definition

>>> direction? A <speedChange> for "both" directions is a short-cut railML
>>> definition for two distinct speed definition positions each for another
>>> driving direction, they are typically a block distance away from each
>>> other.
>>>
>>> Is this information bound to a speed aspect along the track or
>>> does it additionally differs for various trains?
>>>
>>> I'm not sure, if this is the right place for this kind of
>>> information. Who could help out with some more experiences?
>>>
>>
>> I'm sorry. But I can not follow you here. For me it seems better to
>> define
>> the direction inside of the speed profile and not at every <speedChange>.
>
> The direction at <speedChange> is currently in use. I try to explain
> extending the above example:
>
Ok. But it is only optional and not required so you can leave it away. Or it
is a question of handling of diffictulties between speedProfiles direction
and speedChange direction. So if it is defined that a speedProfile direction
has the higher priority it is clear. Or not? We can also define that you are
not allowed to use the direction attribute at the speedChange if you use
speedProfiles ...

> <speedChange pos="10" dir="up" vMax="40" trainRelation="headOfTrain"/>
> <speedChange pos="10" dir="down" vMax="60" trainRelation="midOfTrain"/>
> <speedChange pos="200" dir="both" vMax="80" trainRelation="endOfTrain"/>
>
> pos   10   200
> track dir   -- >
> vMax ->   40->   80->
> train rel -> head->   end->
> vMax <-   <-?   <-60   <-80
> train rel <- <-?   <-mid
> <-end
>
> The "trainRelation" attribute applies to the starting point of the "speed
> aspect track section".
>
[...]
> Thanks for the explanation. Maybe we could shorten the name to "stop"?
>
Maybe, but obligational stresses the importance of this stop.

>>>>   <xs:attribute name="vMax" type="rail:tSpeedKmPerHour"
>>>> use="required"/>

>>>> </xs:attributeGroup>
>>>
>>> I would like to rename this attribute to "maxSpeed", but that only comes
>>> with next major release. (Trac ticket #108) ;-)
>>>
>> Hm. You told me no abbreviations inside RailML?!
>
> We allow 'max' for 'maximum' and 'min' for 'minimum'. They are not
> misleading and widely known.
>
[no comment]

>> By the way: We need to define an "endOfSpeedList". So for example if you
>> have to run slow with a train e.g. above a bridge the allowed speed goes
>> back to track speed after you left the bridge. So you need an option to
>> say:
>> the speed profile ends here. Other speed profiles are now in progress. So
>> you can use speed profiles along the whole train run but they have one
>> one
>> speed limitation at a short bridge or something like this.
>
> No, that use case is covered by allowing multiple speed profiles that
> overlap each other. The software export has to ensure that there are
> valid speed definitions for all sections of the track.
>
This maybe a missunderstanding.
For example the bridge with there speed limit. So you define a speedProfile
for a high axleLoad or something like this and say at the beginning of the
bridge: maxSpeed="60". So what do you write at the end of the bridge where
the speed raises up to a minimum of all the other speedProfiles?

>>> Thanks for your suggestions, they are really helpful for further
>>> development. Let's take them up to railML style XML structures for
>>> providing sustainable XML schemas. :-)
>>>
>> Just do it. ;)
>
> I think we are on the right way. ;-)
>
I hope so.

Best regards.

Carsten

Subject: Re: More detailed 'speed change' definitions

Hi,

The current thread evolves to some really fundamental discussion about future infrastructure (track layout) definitions in railML.

Anybody who already uses railML infrastructure or plans to implement it, please feel personally invited for taking part in this discussion. Any questions, comments, opinions are highly appreciated.

"Carsten Weber" <weber@irfp.de> writes:
> "Susanne Wunsch" <coord@common.railml.org> schrieb im Newsbeitrag
> news:bb28vlg7rym.fsf@remi.heep.sax.de...
>> Hi Carsten, and all who are interested,
>>
>> "Carsten Weber" <weber@irfp.de> writes:
>>> "Susanne Wunsch" <coord@common.railml.org> schrieb im Newsbeitrag
>>> news:bb2ty4a74xn.fsf@remi.heep.sax.de...
>>>> "Carsten Weber" <weber@irfp.de> writes:
>>
>> If your track may be operated in both directions - yes. I try to put a
>> small example here, hope it helps.
>>
>> <speedChange pos="10" dir="up" vMax="40"/>
>> <speedChange pos="10" dir="down" vMax="60"/>
>> <speedChange pos="200" dir="both" vMax="80"/>
>>
>> pos 10 200
>> track dir ---------- >
>> vMax -> 40-> 80->
>> vMax <- <-? <-60 <-80
>>
>> The <speedChange> information defines the maximum speed aspect for the
>> next track section, means up to the next <speedChange> element in the
>> track definition direction. A <speedChange> for both directions means,
>> that the speed aspect at the next track section is the same for both
>> train running directions. It is _not meant_ to be the same speed aspect
>> from this point in both directions - that is really misleading!
>>
> So it becomes tricky to use this structure.

Yes. That may be current practice.

Nevertheless I asked some current railML IS user for their current practice and opinion about the above issue. Most of them appreciated a clearer structure without the possibility to define speed aspects for both directions changing at a certain "point". That's really too much

confusing.

If nobody disagrees with good reasons I would file a Track ticket for
"deprecating" the "both" enumeration value from the "dir" attribute in
all "*Change" elements.

New or already used practice is to define seperate elements for each
running direction. The above example would be:

<speedChange pos="10" dir="up" vMax="40"/>
<speedChange pos="200" dir="up" vMax="80"/>
<speedChange pos="2xx" dir="down" vMax="80"/>
<speedChange pos="200" dir="down" vMax="60"/>

That means that the semantic for the same XML content changes. That is a
very hard cut that can't be recognized by any parser!

>> The above definition does not include the newly <profileRef>, that comes
>> in addition. It shows the current usage of <speedChange> elements. If
>> your <speedProfile> includes a "direction" attribute, you have to split
>> the <speedChange> elements with current 'dir="both"' attributes.
>>
>> Are there any speed profiles that apply only in one direction?
>
> Yes. For example if you have a railway crossing which is operated by sight
> (without any technical system) you may have speed restrictions only for one
> direction caused by the sight to the track.
> | |
> A============B
>   xxxx | |
>
> So you need a speed restriction in direction A to B but not in direction B
> to A. Such cases exist.

Ack.

>> If you have a speed restriction along a bridge, you may define different
>> <speedChange> elements in each direction refering to the same speed
>> profile with different speed aspects. The <speedChange> elements already
>> need the direction attribute. Why do we need to duplicate it?
>>
>
> Because of more clearance.
> In my sight a speedChange should be a child of a speedProfile.
> <speedProfile direction="up" ...>
>   <speedChange position="0" speed="60" .../>
>   <speedChange position="5" speed="120" .../>
>   <speedChange position="123" speed="100" .../>

> <.../>
> </speedProfile>
>
> So you can see: the speedChanges if you run the track in one direction in a
> line.
> But this structure requires a break in downward compatibility which is not
> leagl at the moment.

The compatibility break is only one side effect of the above mentioned example. Another aspect is that speed profiles are much more general definitions than the detailed speed changes along the tracks.

You may define some speed profiles for different axle weights that are referenced at multiple tracks. If you would define these speed profiles at each track you get heavy redundancies and may "lose track of the big picture".

What's the difference between these two approaches from the timetable point of view?

Let's define a train (part sequence) that goes along three tracks deploying two speed profiles.

```
(pos)    0   150   345    468   768   1035
track    t1->      t2->   t3->
vMax sp1 80->      80->   80->
vMax sp2     40->  40->   40->  end
```

That means the train (part sequence) deploying both speed profiles is restricted to the following speed profile:

```
(pos)    0   150   345    468   768   1035
vMax     80->40->  40->   40->  80->
```

Now let's have a look at your approach (how I understood):

```
<track id="t1">
  <trackTopology>...(pos=0..345)
  <trackElements>
    <speedProfile id="sp1" description="Default" dir="up"
                  influence="increasing">
      <speedChange id="sc1" pos="0" vMax="80"/>
    </speedProfile>
    <speedProfile id="sp2" description="temporary restriction" dir="up"
                  influence="reducing">
      <speedChange id="sc2" pos="150" vMax="40"/>
    </speedProfile>
  </trackElements>
```

```
</track>
<track id="t2">
  <trackTopology>...(pos=0..123)
  <trackElements>
    <speedProfile id="sp3" description="Default" dir="up"
                  influence="increasing">
      <speedChange id="sc3" pos="0" vMax="80"/>
    </speedProfile>
    <speedProfile id="sp4" description="temporary restriction" dir="up"
                  influence="reducing">
      <speedChange id="sc4" pos="0" vMax="40"/>
    </speedProfile>
  </trackElements>
</track>
<track id="t3">
  <trackTopology>...(pos="0..567)
  <trackElements>
    <speedProfile id="sp5" description="Default" dir="up"
                  influence="increasing">
      <speedChange id="sc5" pos="0" vMax="80"/>
    </speedProfile>
    <speedProfile id="sp6" description="temporary restriction" dir="up"
                  influence="reducing">
      <speedChange id="sc6" pos="0" vMax="40"/>
      <speedChange id="sc7" pos="300" vMax="end"/>
    </speedProfile>
  </trackElements>
</track>
```

Reference from within the timetable:

```
<trainPartSequence>
  <trainPartRef ref="tp1"/>
  <trainPartRef ref="tp2"/>
  <trainPartRef ref="tp3"/>
  <speedProfileRef ref="sp1"/>
  <speedProfileRef ref="sp2"/>
  <speedProfileRef ref="sp3"/>
  <speedProfileRef ref="sp4"/>
  <speedProfileRef ref="sp5"/>
  <speedProfileRef ref="sp6"/>
</trainPartSequence>

<trainPart id="tp1">
...
  <sectionTT>
    <trackRef ref="t1" dir="up"/>
...
```

```
<trainPart id="tp2">
...
  <sectionTT>
    <trackRef ref="t2" dir="up"/>
...
<trainPart id="tp3">
...
  <sectionTT>
    <trackRef ref="t3" dir="up"/>
```

Now let's have a look at my approach:

```
<speedProfile id="sp1" description="Default" influence="increasing"/>
<speedProfile id="sp2" description="temporary restriction"
                                        influence="reducing"/>
...
<track id="t1">
  <trackTopology>...(pos=0..345)
  <trackElements>
    <speedChange id="sc1" pos="0" dir="up" profileRef="sp1" vMax="80"/>
    <speedChange id="sc2" pos="150" dir="up" profileRef="sp2" vMax="40"/>
  </trackElements>
</track>
<track id="t2">
  <trackTopology>...(pos=0..123)
  <trackElements>
    <speedChange id="sc3" pos="0" dir="up" profileRef="sp1" vMax="80"/>
    <speedChange id="sc4" pos="0" dir="up" profileRef="sp2" vMax="40"/>
  </trackElements>
</track>
<track id="t3">
  <trackTopology>...(pos="0..567)
  <trackElements>
    <speedChange id="sc5" pos="0" dir="up" profileRef="sp1" vMax="80"/>
    <speedChange id="sc6" pos="0" dir="up" profileRef="sp2" vMax="40"/>
    <speedChange id="sc7" pos="300" dir="up" profileRef="sp2" vMax="end"/>
  </trackElements>
</track>
```

Suppose the following reference from within the timetable.

```
<trainPartSequence>
  <trainPartRef ref="tp1"/>
  <trainPartRef ref="tp2"/>
  <trainPartRef ref="tp3"/>
  <speedProfileRef ref="sp1"/>
  <speedProfileRef ref="sp2"/>
</trainPartSequence>
```

The <trainPart>... definitions are the same as in your approach.

>> In order to calculate the maximum speed aspect for a train along a track
>> section you have to follow the speedChange elements along the track
>> (referenced in the sectionTT) and take only the elements refering to the
>> specified speed profiles. If you get "overlapping" speed aspect you take
>> the highest "increasing" minus the deepest "decreasing" speed aspect.
>>
>> This way you don't need "both". "unknown" would be a fixed value that
>> can't be used for calculating with other overlapping speed aspects.
>>
> Ok. So what should we do if the keyword unknown appears?

Ok. Let's drop the "unknown"! ;-)

>>>> > <xs:element name="tiltingAngle" minOccurs="0" maxOccurs="1">
>>>> > <xs:annotation>
>>>> > <xs:documentation>Tilting parameters for which this speed profile
>>>> > is
>>>> > calculated.</xs:documentation>
>>>> > </xs:annotation>
>>>> > </xs:element>
>>>>
>>>> Good idea. I would spend a child element <tilting> for this and the
>>>> following kind of information. The attribute "angle" could be bound to
>>>> the railML type "tAngleDegQuadrant" for allowing 0° to 90°.
>>>>
>>>> Does a speedProfile covers only one tilting angle value or a range of
>>>> values or some single values?
>>>>
>>> It is a question of definition. In my sight this value means: your
>>> vehicle
>>> has to be able to reach a tilting angle of ... (or more).
>>
>> If your vehicle tilts up to 3.5° it has to pass the curve slower than
>> at an tilting angle of 8°. Maybe you have a speed aspect for a range of
>> tilting angle like:
>>
>> 0° .. 3.4° 80 km/h
>> 3.5° .. 5.4° 90 km/h
>> 5.5° .. 8° 100 km/h
>>
> Yes. So you would have to define 3 different speed profiles. A speed profile
> without any data to tiliting or a tilting angle of "0", a second one with
> tilting angles higher than 3.5° and a third one for tilting angles higher
> than 5.5°.

If that definition satifies all needs of railML users - let's do it this
way. Prior to fixing this issue to your clarification I would like to
ask for some Swiss, Austrian... experiences with tilting speed profiles.

>>>> >   <xs:element name="tiltingSpeed" minOccurs="0" maxOccurs="1">
>>>> >   <xs:annotation>
>>>> >   <xs:documentation>Tilting parameters for which this speed profile
>>>> > is
>>>> > calculated.</xs:documentation>
>>>> >   </xs:annotation>
>>>> >   </xs:element>
>>>>
>>>> The terminus "speed" may be a bit misleading. I suppose, that is not
>>>> related to the "train speed" but to the "rate/speed of tilting", that
>>>> means the value of tilting degrees per second. I would call this
>>>> attribute "rate". Are there any other ideas?
>>>>
>>>> This attribute may be bound to the railML type "tSpeedDegreesPerSecond".
>>>>
>>>> There is another kind of information related to the tilting that comes
>>>> to my mind: the method of tilting. It could go into an attribute
>>>> "method" that is bound to an enumeration of "active", "passive",
>>>> "rollCompensation", "unknown", "other:anything".
>>>>
>>> I know this. But is it really important here? Do I need to know which way
>>> the vehicle tilts?
>>
>> I thought that it is another type of speed restriction. If you go with
>> "roll compensation" you may increase your speed, if you support "active
>> tilting", you have to switch the system off and go with default speed
>> aspect. This may be a Swiss use case, I mean.
>>
> Yes and no. So you need a speed profile for tilting angles up to 2° (or
> something like that) and you may have a second one for higher tilting
> angles. So if your vehicle is only able to do a tilting up to 2° you can
> only run the "slow" profile. The way you reach the tilting angle is not
> really important. It is more a question of the tilting speed but it is also
> not important there.

Are there any different speed profiles for the same tilting angle
depending on the tilting method?

>>> By the way: We need to define an "endOfSpeedList". So for example if you
>>> have to run slow with a train e.g. above a bridge the allowed speed goes
>>> back to track speed after you left the bridge. So you need an option to
>>> say:
>>> the speed profile ends here. Other speed profiles are now in progress. So
>>> you can use speed profiles along the whole train run but they have one

>>> one
>>> speed limitation at a short bridge or something like this.
>>
>> No, that use case is covered by allowing multiple speed profiles that
>> overlap each other. The software export has to ensure that there are
>> valid speed definitions for all sections of the track.
>>
> This maybe a missunderstanding.
> For example the bridge with there speed limit. So you define a speedProfile
> for a high axleLoad or something like this and say at the beginning of the
> bridge: maxSpeed="60". So what do you write at the end of the bridge where
> the speed raises up to a minimum of all the other speedProfiles?

+1

I really missed this issue. We need some "end of speed restriction"
indication. That may be defined as additional fixed string allowed for
the "vMax" attribute, e.g.

```
<speedChange id="sc7" pos="300" dir="up" profileRef="sp2" vMax="end"/>
```

Thanks for the ongoing lively discussion.

Kind regards...
Susanne

--
Susanne Wunsch
Schema Coordinator: railML.common

Subject: Re: More detailed 'speed change' definitions
Posted by Carsten Weber on Mon, 23 Jan 2012 16:02:04 GMT
View Forum Message <> Reply to Message

Hi,

"Susanne Wunsch" <coord@common.railml.org> schrieb im Newsbeitrag
news:bb2ty3ztd6e.fsf@remi.heep.sax.de...
> Hi,
>
> The current thread evolves to some really fundamental discussion about
> future infrastructure (track layout) definitions in railML.
>
> Anybody who already uses railML infrastructure or plans to implement
> it, please feel personally invited for taking part in this
> discussion. Any questions, comments, opinions are highly appreciated.
>

It seems to be a dialque and not a discussion. :/

> "Carsten Weber" <weber@irfp.de> writes:
>> "Susanne Wunsch" <coord@common.railml.org> schrieb im Newsbeitrag
>> news:bb28vlg7rym.fsf@remi.heep.sax.de...
>>> Hi Carsten, and all who are interested,
>>>
>>> "Carsten Weber" <weber@irfp.de> writes:
>>>> "Susanne Wunsch" <coord@common.railml.org> schrieb im Newsbeitrag
>>>> news:bb2ty4a74xn.fsf@remi.heep.sax.de...
>>>> > "Carsten Weber" <weber@irfp.de> writes:
>>>
>>> If your track may be operated in both directions - yes. I try to put a
>>> small example here, hope it helps.
>>>
>>> <speedChange pos="10" dir="up" vMax="40"/>
>>> <speedChange pos="10" dir="down" vMax="60"/>
>>> <speedChange pos="200" dir="both" vMax="80"/>
>>>
>>> pos 10 200
>>> track dir ---------- >
>>> vMax -> 40-> 80->
>>> vMax <- <-? <-60 <-80
>>>
>>> The <speedChange> information defines the maximum speed aspect for the
>>> next track section, means up to the next <speedChange> element in the
>>> track definition direction. A <speedChange> for both directions means,
>>> that the speed aspect at the next track section is the same for both
>>> train running directions. It is _not meant_ to be the same speed aspect
>>> from this point in both directions - that is really misleading!
>>>
>> So it becomes tricky to use this structure.
>
> Yes. That may be current practice.
>
> Nevertheless I asked some current railML IS user for their current
> practice and opinion about the above issue. Most of them appreciated a
> clearer structure without the possibility to define speed aspects for
> both directions changing at a certain "point". That's really too much
> confusing.
>
> If nobody disagrees with good reasons I would file a Track ticket for
> "deprecating" the "both" enumeration value from the "dir" attribute in
> all "*Change" elements.
>
> New or already used practice is to define seperate elements for each
> running direction. The above example would be:
>

> <speedChange pos="10" dir="up" vMax="40"/>
> <speedChange pos="200" dir="up" vMax="80"/>
> <speedChange pos="2xx" dir="down" vMax="80"/>
> <speedChange pos="200" dir="down" vMax="60"/>
>
> That means that the semantic for the same XML content changes. That is a
> very hard cut that can't be recognized by any parser!
>
It is a question of your position. Do I have to combine two speed
restrictions which are shown at the same position to differnt dircetions
into one element? I do not think so. So I can keep both of them in two
elements and do not do any mistake in RailML-useage. So for me the cut looks
not so heavy.

[...]

>>> If you have a speed restriction along a bridge, you may define different
>>> <speedChange> elements in each direction refering to the same speed
>>> profile with different speed aspects. The <speedChange> elements already
>>> need the direction attribute. Why do we need to duplicate it?
>>>
>>
>> Because of more clearance.
>> In my sight a speedChange should be a child of a speedProfile.
>> <speedProfile direction="up" ...>
>> <speedChange position="0" speed="60" .../>
>> <speedChange position="5" speed="120" .../>
>> <speedChange position="123" speed="100" .../>
>> <.../>
>> </speedProfile>
>>
>> So you can see: the speedChanges if you run the track in one direction in
>> a
>> line.
>> But this structure requires a break in downward compatibility which is
>> not
>> leagl at the moment.
>
> The compatibility break is only one side effect of the above mentioned
> example. Another aspect is that speed profiles are much more general
> definitions than the detailed speed changes along the tracks.
>
> You may define some speed profiles for different axle weights that are
> referenced at multiple tracks. If you would define these speed profiles
> at each track you get heavy redundancies and may "lose track of the big
> picture".
>
> What's the difference between these two approaches from the timetable

> point of view?
>
> Let's define a train (part sequence) that goes along three tracks
> deploying two speed profiles.
>
> (pos) 0 150 345 468 768 1035
> track t1-> t2-> t3->
> vMax sp1 80-> 80-> 80->
> vMax sp2 40-> 40-> 40-> end
>
> That means the train (part sequence) deploying both speed profiles is
> restricted to the following speed profile:
>
> (pos) 0 150 345 468 768 1035
> vMax 80->40-> 40-> 40-> 80->
>
> Now let's have a look at your approach (how I understood):
>
> <track id="t1">
> <trackTopology>...(pos=0..345)
> <trackElements>
> <speedProfile id="sp1" description="Default" dir="up"
> influence="increasing">
> <speedChange id="sc1" pos="0" vMax="80"/>
> </speedProfile>
> <speedProfile id="sp2" description="temporary restriction" dir="up"
> influence="reducing">
> <speedChange id="sc2" pos="150" vMax="40"/>
> </speedProfile>
> </trackElements>
> </track>
> <track id="t2">
> <trackTopology>...(pos=0..123)
> <trackElements>
> <speedProfile id="sp3" description="Default" dir="up"
> influence="increasing">
> <speedChange id="sc3" pos="0" vMax="80"/>
> </speedProfile>
> <speedProfile id="sp4" description="temporary restriction" dir="up"
> influence="reducing">
> <speedChange id="sc4" pos="0" vMax="40"/>
> </speedProfile>
> </trackElements>
> </track>
> <track id="t3">
> <trackTopology>...(pos="0..567)
> <trackElements>
> <speedProfile id="sp5" description="Default" dir="up"

> influence="increasing">
> <speedChange id="sc5" pos="0" vMax="80"/>
> </speedProfile>
> <speedProfile id="sp6" description="temporary restriction" dir="up"
> influence="reducing">
> <speedChange id="sc6" pos="0" vMax="40"/>
> <speedChange id="sc7" pos="300" vMax="end"/>
> </speedProfile>
> </trackElements>
> </track>
>
> Reference from within the timetable:
>
> <trainPartSequence>
> <trainPartRef ref="tp1"/>
> <trainPartRef ref="tp2"/>
> <trainPartRef ref="tp3"/>
> <speedProfileRef ref="sp1"/>
> <speedProfileRef ref="sp2"/>
> <speedProfileRef ref="sp3"/>
> <speedProfileRef ref="sp4"/>
> <speedProfileRef ref="sp5"/>
> <speedProfileRef ref="sp6"/>
> </trainPartSequence>
>
> <trainPart id="tp1">
> ...
> <sectionTT>
> <trackRef ref="t1" dir="up"/>
> ...
> <trainPart id="tp2">
> ...
> <sectionTT>
> <trackRef ref="t2" dir="up"/>
> ...
> <trainPart id="tp3">
> ...
> <sectionTT>
> <trackRef ref="t3" dir="up"/>
>
> Now let's have a look at my approach:
>
> <speedProfile id="sp1" description="Default" influence="increasing"/>
> <speedProfile id="sp2" description="temporary restriction"
> influence="reducing"/>
> ...
> <track id="t1">
> <trackTopology>...(pos=0..345)

>   <trackElements>
>   <speedChange id="sc1" pos="0" dir="up" profileRef="sp1" vMax="80"/>
>   <speedChange id="sc2" pos="150" dir="up" profileRef="sp2" vMax="40"/>
>   </trackElements>
> </track>
> <track id="t2">
>   <trackTopology>...(pos=0..123)
>   <trackElements>
>   <speedChange id="sc3" pos="0" dir="up" profileRef="sp1" vMax="80"/>
>   <speedChange id="sc4" pos="0" dir="up" profileRef="sp2" vMax="40"/>
>   </trackElements>
> </track>
> <track id="t3">
>   <trackTopology>...(pos="0..567)
>   <trackElements>
>   <speedChange id="sc5" pos="0" dir="up" profileRef="sp1" vMax="80"/>
>   <speedChange id="sc6" pos="0" dir="up" profileRef="sp2" vMax="40"/>
>   <speedChange id="sc7" pos="300" dir="up" profileRef="sp2"
> vMax="end"/>
>   </trackElements>
> </track>
>
> Suppose the following reference from within the timetable.
>
> <trainPartSequence>
>   <trainPartRef ref="tp1"/>
>   <trainPartRef ref="tp2"/>
>   <trainPartRef ref="tp3"/>
>   <speedProfileRef ref="sp1"/>
>   <speedProfileRef ref="sp2"/>
> </trainPartSequence>
>
> The <trainPart>... definitions are the same as in your approach.
>
Ok. Now I understand. Your approach looks quite heavy. So the exclusion of
common information is quite good if you have a lot of information the
exclude but I think this does not appear in this situation.

[...]

>>>
>> Yes. So you would have to define 3 different speed profiles. A speed
>> profile
>> without any data to tiliting or a tilting angle of "0", a second one with

>
> If that definition satifies all needs of railML users - let's do it this
> way. Prior to fixing this issue to your clarification I would like to
> ask for some Swiss, Austrian... experiences with tilting speed profiles.
>
Ok. Let me know the answer(s).

>>>> >> <xs:element name="tiltingSpeed" minOccurs="0" maxOccurs="1">
>>>> >> <xs:annotation>
>>>> >> <xs:documentation>Tilting parameters for which this speed
>>>> >> profile
>>>> >> is
>>>> >> calculated.</xs:documentation>
>>>> >> </xs:annotation>
>>>> >> </xs:element>
>>>> >
>>>> > The terminus "speed" may be a bit misleading. I suppose, that is not
>>>> > related to the "train speed" but to the "rate/speed of tilting", that
>>>> > means the value of tilting degrees per second. I would call this
>>>> > attribute "rate". Are there any other ideas?
>>>> >
>>>> > This attribute may be bound to the railML type
>>>> > "tSpeedDegreesPerSecond".
>>>> >
>>>> > There is another kind of information related to the tilting that comes
>>>> > to my mind: the method of tilting. It could go into an attribute
>>>> > "method" that is bound to an enumeration of "active", "passive",
>>>> > "rollCompensation", "unknown", "other:anything".
>>>> >
>>>> I know this. But is it really important here? Do I need to know which
>>>> way
>>>> the vehicle tilts?
>>>
>>> I thought that it is another type of speed restriction. If you go with
>>> "roll compensation" you may increase your speed, if you support "active
>>> tilting", you have to switch the system off and go with default speed
>>> aspect. This may be a Swiss use case, I mean.
>>>

>> something like that) and you may have a second one for higher tilting

>> only run the "slow" profile. The way you reach the tilting angle is not
>> really important. It is more a question of the tilting speed but it is
>> also
>> not important there.
>
> Are there any different speed profiles for the same tilting angle

> depending on the tilting method?
>
I do not think so. But maybe anybody else has an opposite example.

>>>> By the way: We need to define an "endOfSpeedList". So for example if
>>>> you
>>>> have to run slow with a train e.g. above a bridge the allowed speed
>>>> goes
>>>> back to track speed after you left the bridge. So you need an option to
>>>> say:
>>>> the speed profile ends here. Other speed profiles are now in progress.
>>>> So
>>>> you can use speed profiles along the whole train run but they have one
>>>> one
>>>> speed limitation at a short bridge or something like this.
>>>
>>> No, that use case is covered by allowing multiple speed profiles that
>>> overlap each other. The software export has to ensure that there are
>>> valid speed definitions for all sections of the track.
>>>
>> This maybe a missunderstanding.
>> For example the bridge with there speed limit. So you define a
>> speedProfile
>> for a high axleLoad or something like this and say at the beginning of
>> the
>> bridge: maxSpeed="60". So what do you write at the end of the bridge
>> where
>> the speed raises up to a minimum of all the other speedProfiles?
>
> +1
>
> I really missed this issue. We need some "end of speed restriction"
> indication. That may be defined as additional fixed string allowed for
> the "vMax" attribute, e.g.
>
> <speedChange id="sc7" pos="300" dir="up" profileRef="sp2" vMax="end"/>
>
Yes, sSomething like this.
So it can be mixed up with another speed information.

When will the preview be available?

Best regards.

Subject: Re: More detailed 'speed change' definitions
Posted by Christian Rahmig on Wed, 25 Jan 2012 21:08:36 GMT

Hello Susanne, hello Carsten,

>> The current thread evolves to some really fundamental discussion about
>> future infrastructure (track layout) definitions in railML.
>>
>> Anybody who already uses railML infrastructure or plans to implement
>> it, please feel personally invited for taking part in this
>> discussion. Any questions, comments, opinions are highly appreciated.
>>
> It seems to be a dialque and not a discussion. :/

I am sorry for entering your discussion so late. However, it is very
interesting to see that although the infrastructure schema is not very
complex at the moment, it may be interpreted in different ways.
Therefore, we really need to think about the infrastructure
re-structuring as part of a major release.

>>>> If your track may be operated in both directions - yes. I try to put a
>>>> small example here, hope it helps.
>>>>
>>>> <speedChange pos="10" dir="up" vMax="40"/>
>>>> <speedChange pos="10" dir="down" vMax="60"/>
>>>> <speedChange pos="200" dir="both" vMax="80"/>
>>>>
>>>> pos 10 200
>>>> track dir ---------- >
>>>> vMax -> 40-> 80->
>>>> vMax<-<-?<-60<-80
>>>>
>>>> The<speedChange> information defines the maximum speed aspect for the
>>>> next track section, means up to the next<speedChange> element in the
>>>> track definition direction. A<speedChange> for both directions means,
>>>> that the speed aspect at the next track section is the same for both
>>>> train running directions. It is _not meant_ to be the same speed aspect
>>>> from this point in both directions - that is really misleading!
>>>>
>>> So it becomes tricky to use this structure.
>>
>> Yes. That may be current practice.
>>
>> Nevertheless I asked some current railML IS user for their current
>> practice and opinion about the above issue. Most of them appreciated a
>> clearer structure without the possibility to define speed aspects for
>> both directions changing at a certain "point". That's really too much
>> confusing.
>>
>> If nobody disagrees with good reasons I would file a Track ticket for

>> "deprecating" the "both" enumeration value from the "dir" attribute in
>> all "*Change" elements.
>>
>> New or already used practice is to define seperate elements for each
>> running direction. The above example would be:
>>
>> <speedChange pos="10" dir="up" vMax="40"/>
>> <speedChange pos="200" dir="up" vMax="80"/>
>> <speedChange pos="2xx" dir="down" vMax="80"/>
>> <speedChange pos="200" dir="down" vMax="60"/>
>>
>> That means that the semantic for the same XML content changes. That is a
>> very hard cut that can't be recognized by any parser!
>>
> It is a question of your position. Do I have to combine two speed
> restrictions which are shown at the same position to differnt dircetions
> into one element? I do not think so. So I can keep both of them in two
> elements and do not do any mistake in RailML-useage. So for me the cut looks
> not so heavy.

I agree with you, Carsten. The two interpretations of the usage of speedChange elements are not a cut within the railML infrastructure syntax. Nevertheless, I support the idea of you, Susanne, to mark the "both" value of the direction attribute as being deprecated. From the user's point of view, speedChange points need to be referenced with a specified direction (up or down) on the track segment.

>>>> If you have a speed restriction along a bridge, you may define different
>>>> <speedChange> elements in each direction refering to the same speed
>>>> profile with different speed aspects. The<speedChange> elements already
>>>> need the direction attribute. Why do we need to duplicate it?
>>>>
>>>
>>> Because of more clearance.
>>> In my sight a speedChange should be a child of a speedProfile.
>>> <speedProfile direction="up" ...>
>>> <speedChange position="0" speed="60" .../>
>>> <speedChange position="5" speed="120" .../>
>>> <speedChange position="123" speed="100" .../>
>>> <.../>
>>> </speedProfile>
>>>
>>> So you can see: the speedChanges if you run the track in one direction in
>>> a
>>> line.
>>> But this structure requires a break in downward compatibility which is
>>> not
>>> leagl at the moment.

>>
>> The compatibility break is only one side effect of the above mentioned
>> example. Another aspect is that speed profiles are much more general
>> definitions than the detailed speed changes along the tracks.
>>
>> You may define some speed profiles for different axle weights that are
>> referenced at multiple tracks. If you would define these speed profiles
>> at each track you get heavy redundancies and may "lose track of the big
>> picture".

I really appreciate the discussion about a possible speedProfile element, because it questions the current track-based structure of the whole infrastructure schema.

Focusing on the speedProfile idea, I see the same problem like Susanne: a direction attribute within a speedProfile is a redundany to the direction attribute of the speedChange. Furthermore, it is not even unique in usage, e.g. when a speed profile contains speedChange points of several tracks, which are oriented differently. In other words: while the speedChange points can refer to the direction of the track they are placed on, a speedProfile cannot do likewise.

Defining speedChange points as "children" of a speedProfile brings up the question of the dimension of a speedProfile: Where does it begin and where does it end? From a user's point of view, this aspect depends on the application and by interpreting the speedChange points, a speedProfile that matches the special requirements of the application can be generated.

Kind regards

Christian Rahmig
railML.infrastructure coordinator

Subject: Re: More detailed 'speed change' definitions
Posted by Susanne Wunsch railML on Tue, 24 Apr 2012 21:31:07 GMT
View Forum Message <> Reply to Message

Hello to all who are interested,

We discussed some open issues of the upcoming speed change / speed profile enhancements (for railML version 2.2) afterwards the todays' railML meeting in Dresden. I conclude the outcomes herein.

[directions in speed changes]

<speedChange> elements offer the "dir" attribute with the enumeration values "up" and "down".

(See also Trac ticket #145 [1] for renaming the enumeration values with next major release.)

The current enumeration value "both" is marked deprecated for more clear semantics.

[position of speed changes in the XML tree]

<speedChange> elements stay where they are currently defined.

<speedProfile> elements are seperated from the <tracks> and (with respect to its container element <speedProfiles>) a direct child of the <infrastructure> element.

[different run history]

The actual speed aspect depends not only on the rollingstock characteristics as mentioned in the previous postings. It sometimes depends on the route through a "branching station" from a macroscopic point of view.

Given the route between the neighbouring stops/stations (ocps) the different speed aspects at the same track for the same rollingstock characteristics may be defined.

So far we would need two attributes for refering to <ocp id=""> elements at the <speedProfile> element. "from" and "to" don't help in this case because they also apply to the other running direction which would be confusing.

How about the attributes "ocpRef1" and "ocpRef2"? Or "neighbour1" and "neighbour2"? Or "neighbourOcpRef1" and "neighbourOcpRef2"?

Any other (even better) naming suggestions?

[train relation]

What is a real use case for the enumeration value "midOfTrain"? Are there any speed aspects that are valid since half of the train passed its defined position?

If it is not the case, we would suggest not to define it.

[minimum percentage of brake power]

At some railway infrastructure companies the minimum percentage of brake power can't be directly calculated by means of physics. It is somehow defined by some legal body.

Therefore we would suggest an additional attribute "minimumBrakePercentage" for this value in the <speedProfile> element.

Any comments, remarks, questions are highly appreciated.

[1] http://trac.assembla.com/railML/ticket/145

--
Kind regards...
Susanne

Subject: Re: More detailed 'speed change' definitions
Posted by Christian Rahmig on Sat, 16 Jun 2012 16:59:33 GMT
View Forum Message <> Reply to Message

Hello Susanne et.al.,

> We discussed some open issues of the upcoming speed change / speed
> profile enhancements (for railML version 2.2) afterwards the todays'
> railML meeting in Dresden. I conclude the outcomes herein.
>
> [directions in speed changes]
>
> <speedChange> elements offer the "dir" attribute with the enumeration
> values "up" and "down".
>
> (See also Trac ticket #145 [1] for renaming the enumeration values with
> next major release.)
>
> The current enumeration value "both" is marked deprecated for more
> clear semantics.

agreed.

> [position of speed changes in the XML tree]
>
> <speedChange> elements stay where they are currently defined.
>
> <speedProfile> elements are seperated from the<tracks> and (with
> respect to its container element<speedProfiles>) a direct
> child of the<infrastructure> element.

agreed.

> [different run history]
>
> The actual speed aspect depends not only on the rollingstock
> characteristics as mentioned in the previous postings. It sometimes
> depends on the route through a "branching station" from a macroscopic
> point of view.
>
> Given the route between the neighbouring stops/stations (ocps) the
> different speed aspects at the same track for the same rollingstock
> characteristics may be defined.
>
> So far we would need two attributes for refering to<ocp id="">
> elements at the<speedProfile> element. "from" and "to" don't help in
> this case because they also apply to the other running direction which
> would be confusing.
>
> How about the attributes "ocpRef1" and "ocpRef2"? Or "neighbour1" and
> "neighbour2"? Or "neighbourOcpRef1" and "neighbourOcpRef2"?
>
> Any other (even better) naming suggestions?

c.f. ongoing discussion in thread "speed profiles for generl directions" [2]

> [train relation]
>
> What is a real use case for the enumeration value "midOfTrain"? Are
> there any speed aspects that are valid since half of the train passed
> its defined position?
>
> If it is not the case, we would suggest not to define it.

c.f. ongoing discussion in thread "train relation" [1]

> [minimum percentage of brake power]
>
> At some railway infrastructure companies the minimum percentage of
> brake power can't be directly calculated by means of physics. It is
> somehow defined by some legal body.
>
> Therefore we would suggest an additional attribute
> "minimumBrakePercentage" for this value in the<speedProfile> element.

c.f. ongoing discussion in thread "speed profiles and braking
percentages" [3]

[1]
http://www.railml.org/forum/ro/index.php?group=1&offset= 0&thread=45&id=121

[2]
http://www.railml.org/forum/ro/index.php?group=1&offset= 0&thread=46&id=122

[3]
http://www.railml.org/forum/ro/index.php?group=1&offset= 0&thread=44&id=120

Christian Rahmig
railML.infrastructure coordinator

Subject: Re: More detailed 'speed change' definitions
Posted by Susanne Wunsch railML on Fri, 29 Jun 2012 08:40:41 GMT
View Forum Message <> Reply to Message

Hello Carsten, Christian, Dirk and all interested

I tried to conclude the discussion in this and some neighbouring threads updating the Trac ticket #41. [1] For ease of following I copied the content herein.

Please check the new proposal for your needs and give us appropriate feedback. (We are also pleased for positive feedback like "All is fine for us" ;-)

Some "open points" left, please take a look at this section at the end of this posting.

--- Trac ticket #41 comment #4 ----------

I want to conclude the current discussion of multiple forum threads:
http://www.railml.org/forum/ro/index.php?group=1&id=89
http://www.railml.org/forum/ro/index.php?group=1&id=120
http://www.railml.org/forum/ro/index.php?group=1&id=121
http://www.railml.org/forum/ro/index.php?group=1&id=122

```
<infrastructure>
  <infraAttrGroups>...
  <tracks>
    <track id="t1">
      <trackTopology>
        <trackBegin ...
 <trackEnd ...
      </trackTopology>
      <trackElements>
        <speedChanges>
   <speedChange id="sc1" pos="0" vMax="100"  dir="up"
```

```
    profileRef="sp1" trainRelation="endOfTrain"/>
   <speedChange id="sc2" pos="0" vMax="60"  dir="up"
    profileRef="sp2" trainRelation="headOfTrain"/>
   <speedChange id="sc3" pos="0" vMax="40"  dir="up"
    profileRef="sp4" trainRelation="headOfTrain"/>
   <speedChange id="sc4" pos="0" vMax="120"  dir="up"
    profileRef="sp6" trainRelation="endOfTrain"/>
        <speedChange id="sc5" pos="1234.5" vMax="80" dir="up"
    profileRef="sp1" trainRelation="headOfTrain"/>
   <speedChange id="sc6" pos="234.5" vMax="60" dir="up"
    profileRef="sp3" trainRelation="headOfTrain"
         signalised="true"/>
   <speed Change id="sc7" pos="256.7" vMax="end" dir="up"
    profileRef="sp3" trainRelation="endOfTrain"/>
 </speedChanges>
     </trackElements>
   </track>
 </tracks>
 <trackGroups>...
 <operationControlPoints>
   <ocp id="o1"/>
   <ocp id="o2"/>
   <ocp id="o3"/>
 </operationControlPoints>
 <controllers>...
 <speedProfiles>
   <speedProfile id="sp1" code="A12" name="Basis" influence="increasing"
    description="Speed profile as general basis for other overlaying profiles"/>
   <speedProfile id="sp2" code="B3" name="Heavy Freight" influence="decreasing"
    description="Speed profile for heavy freight trains"
    axleLoad="40"/>
   <speedProfile id="sp3" code="C1" name="Bridge" influence="increasing"
    description="Speed profile for typical bridges"
    meterLoad="8"/>
   <speedProfile id="sp4" code="T" name="Temporary" influence="decreasing"
    description="Temporary speed profile due to maintenance work"
    operatingPeriodRef="op_1"/>
   <speedProfile id="sp5" code="T5" name="Tilting" influence="increasing"
    description="Speed profile for tilting trains">
    <tilting maxTiltingAngle="8" actuation="active" tiltingSpeed="1"/>
   </speedProfile>
   <speedProfile id="sp6" code="HS" name="LZB" influence="increasing"
    description="Speed profile for trains with activated LZB train protection system"
    nationalSystem="LZB"/>
   <speedProfile id="sp7" code="R" name="Route" influence="decreasing"
    description="Speed profile for trains on a certain route">
    <route>
     <ocpRef ref="o1"/>
```

```
 <ocpRef ref="o3"/>
      </route>
    </speedProfile>
    <speed Profile id="sp8" code="B" name="Breaking" influence="decreasing"
     description="Speed profile for trains with certain brake capabilities"
     minimumBrakePercentage="123"/>
  </speedProfiles>
</infrastructure>
<timetable>
  <operatingPeriods>
    <operatingPeriod id="op_1" name="Maintenance window"
     startDate="2012-03-01" endDate="2012-03-02">
     <specialService type="include" startTime="23:05:00Z" endTime="04:15:00Z"/>
     <specialService type="exclude" startTime="02:30:00Z" endTime="03:00:00Z"
      singleDate="2012-03-02"/>
    </operatingPeriod>
  </operatingPeriods>
  <trainParts>
    <trainPart id="tp1" name="Passenger train with tilting system">
     <ocpsTT>
      <ocpTT ocpRef="o2">
   <sectionTT>
    <trackRef ref="t1">
     <speedRef ref="sp1"/>
           <speedRef ref="sp3"/>
           <speedRef ref="sp5"/>
     <speedRef ref="sp7"/>
         </trackRef>
       </sectionTT>
      </ocpTT>
     </ocpsTT>
    </trainPart>
    <trainPart id="tp2" name="Freight train">
     <ocpsTT>
      <ocpTT ocpRef="o2">
       <sectionTT>
         <trackRef ref="t1">
          <speedRef ref="sp1"/>
          <speedRef ref="sp2"/>
     <speedRef ref="sp3"/>
     <speedRef ref="sp4"/>
         </trackRef>
       </sectionTT>
    </ocpTT>
     </ocpsTT>
    </trainPart>
  </trainParts>
</timetable>
```

Documentation of changes

* speedChange element

** The profileRef attribute is of the railML type tGenericRef with an additional xs:key-binding. It can be made required with next major release only.

** The trainRelation attribute provides one of the enumeration values headOfTrain, midOfTrain and endOfTrain. It can be made required with next major release only. Mostly the endOfTrain value is used for increasing speed aspects and the headOfTrain value for decreasing speed aspects. In some special cases, e.g. level crossings in Germany, headOfTrain would be also used for increasing speed aspects.

** The vMax attribute provides an additional enumeration value for indicating the end of a local speed restriction, e.g. on a bridge.

** The dir attribute allows only for the enumeration values up and down. The current value both is marked deprecated. They will be renamed in next major release.

** The signalised attribute indicates whether the speed aspects is shown next to text with some light signal or a panel (true) or only in the "drivers timetable" (false).

* speedProfile element

** The code attribute in <speedProfile> element should be used for the companies' internal "abbreviation" in software tools or printed lists.

** If additional (official) codes for identifying a speed profile are needed, please request this. It is currently not foreseen.

** There is no direction attribute in the speed profile.

** The influence attribute allows for overlaying speed profiles. It is defined as enumeration list of increasing and decreasing. It should be required.

** The axleLoad attribute is of the railML type tWeightTons.

** The meterLoad atribute is of the railML type tMeterloadTonsPerMeter.

** If temporary speed restriction will be used the <operatingPeriods> element of the Timetable sub-schema must be applied.

** The maxTiltingAngle attribute is of the railML type tAngleDegQuadrant (0..90°). It is reused from the Rollingstock sub-schema type tTilting.

** The tiltingSpeed attribute is of the railML type tSpeedDegreesPerSecond. It is reused from the Rollingstock sub-schema type tTilting.

** The actuation attribute is of the railML type tTiltingActuationType allowing the enumeration values passive, active, none and rollCompensation. It is reused from the Rollingstock sub-schema type tTilting.

** The nationalSystem attribute is of the railML type tNationalSystemsType. It is provided as "railway base type" for all sub-schemas and provides an extensible enumeration list of train protection system identifications.

** The route element is for indicating a train run between neighbouring "ocp"s no matter which direction it goes.

** The minimumBrakePercentage attribute is of the railML type tBrakePercentage allowing any integer value between 6 and 225.

* speedRef element

** All referred speed profiles are taken into account along the referred track. For getting the resulting speed profile for a certain train the software should calculate according to the following algorithm:

** The track is "virtually" divided into sections with no speed aspect changes looking at each speedChange element of the referred speedProfiles.
1. For each "virtual" track section the most increasing speed aspect has to be found across all referred speedProfiles according to the running direction of the trackRef element.
2. For each "virtual" track section the most decreasing speed aspect has to be found across all referred speedProfiles according to the running direction of the trackRef element. If no decreasing value is found, the most increasing value from the prior step will be acquired.

Open Points

** What to do with the current trainCategory attribute?

** Marking as "deprecated" is an easy way, but there are nevertheless international train categories for lines, e.g. "NC_Train" of the ETCS

domain [1].

** What to do with the current status attribute?

** How to define the "blocking of a track"?

** Using a time restricted speed profile with vMax="0" in the speedChange elements?

** How to define an "obligational stop" where all or only certain trains have to stop prior going on with the same speed aspect as before?

** Could this be an attribute of the speedChange element?

** Do we need the midOfTrain value as trainRelation in the speedChange context?

** The relevance for timetabling (middle of a platform at a station) is not the question here, but the relevance for infrastructure-based speed aspects.

So, what else? ;-)

[1] ETCS Variables v1.2.pdf

---------- ----------

Kind regards...
Susanne

[1] https://trac.assembla.com/railML/ticket/41#comment:4

--
Susanne Wunsch
Schema Coordinator: railML.common

Subject: Re: More detailed 'speed change' definitions
Posted by on Fri, 29 Jun 2012 17:50:09 GMT
View Forum Message <> Reply to Message

Dear Susanne and all others,

thank you for the good conclusion on speed changes!

> Please check the new proposal for your needs and give us appropriate
> feedback. (We are also pleased for positive feedback like "All is fine
> for us"

This would be nice… But it is unfortunately not so easy. I think, for iRFP, we can implement a programming based on the changes you introduced.. If this works well - which I would expect - we could send you an “All is fine for us” from iRFP. As you know, sometimes one only recognises the wishes when implementing the solution.

So far, I wrote some notes below. Most are minor issues, some are possibly important. I think we should consider now the question of the relations (see below). Depending on this, we could start implementing possibly in some weeks.

I did not reply to all of your issues - they are welcome and understandable. So no objection from our side. And thank you again for the good work.

<speedChange id="sc3"...>

Can we have the ‚id’ of a speed change as an optional attribute? Since we have many, many speed changes and seldom the need to refer to them I do (optionally) want to save some space and effort.

> axleLoad="40"

would be interesting to know the unit of ‘axleLoad’ ;-)

> ** The axleLoad attribute is of the railML type tWeightTons.

Oh, thank you. One would not expect it with this value, even not in Australia… ;-)

If they talk about ’40-Tonner’ on streets, I guess they mean _total_ weight - not axle weight (I hope)... ;-)

> minimumBrakePercentage="123"

There are at least two minimum brake percentages for a speed profile: One for brake switch G and another for the other brake switches (P, R). This applies to the middle-European (UIC/RIC) practice of two-valued braking tables. (The other braking switches result in different braking percentages of the vehicles but not in different braking tables.)

In General, we should consider to make the speed profile (section) depending on the triplet
”brakeType”+“minimumBrakePercentage”+”minimumBrake Switch” (see news

entry
from 26.04.2012).

> minimumBrakePercentage="123"

This does not take into account the aspects written on 26.04.2012: There is a "minimumBrakePercentage" for each section of a speed profile between two places where trains can start or end (i. e. between two stations). So, I would prefer to add the above named triplet as attributes of a speed change: They are valid until the next speed change with such attributes.

> axleLoad="40"
> meterLoad="8"

For non-Boolean conditions I would prefer to clarify the relation they are meant with their name: maximumAxleLoad or minimumAxleLoad?

Is the speed profile necessary for trains with an axle load of more than ‘40’ (probably declaring speed restrictions) or for trains with an axle load of less than ‘40’ (probably permitting raised speeds)?

We have done so with ‘minimumBrakePercentage’ so we should also do it with the others.

We can also describe the relation explicitly, defining an (obligatory) attribute with the enumerations ‘lessThan’, ‘greaterThan’, ‘lessOrEqual’, and ‘GreaterOrEqual’. However, we should somehow clarify that, either implicitly in the name or explicitly. We should not leave it to the documentation alone.

There are some attributes of <speedProfile> which act as conditions: The <speedProfile> is valid if the train fulfils the condition (axleLoad, meterLoad, operatingPeriodRef, nationalSystem). At least if there are more than one such conditions, there is the question: Are they linked with logical AND or with logical OR?

Is the speed profile valid for the train if it has an axleLoad of more than xxx tons AND a meter load of more than yyy tons per meter? Or for if the train has an axleLoad of more than xxx tons OR a meter load of more than yyy tons per meter?

This may be left to the documentation but it is not very easy. It depends at least on the attribute influence=decreasing/increasing: Typically, the conditions of an increasing speed profile are necessary (AND), and the conditions of an decreasing speed profile are optional (OR). To clarify this, I would prefer an attribute for the logical link if there is more

than one condition.

> ** The trainRelation attribute provides one of the enumeration values
> headOfTrain, midOfTrain and endOfTrain. It can be made required with
> next major release only. Mostly the endOfTrain value is used for
> increasing speed aspects and the headOfTrain value for decreasing speed
> aspects. In some special cases, e.g. level crossings in Germany,
> headOfTrain would be also used for increasing speed aspects.

<DOCUMENTATION HINT>

We should note, in the documentation, that all speed changes are to be understood to act independently from each other: A speed change at the front of a train does not effect the speed changes still ‘under’ the train. The lowest of all speed changes currently valid for a train is relevant.

This explains why we do not need a trainRelation="headOfTrainIfDecreasingAndEndOfTrainIfIncreasing " enumeration value… If the speed changes come closer than the length of the train (which may always be the case), the train has to maintain several speed restrictions. If the train passes an increasing speed change (valid for head) with its front but has not jet passed a lower increasing speed change (valid for end) with its end, the lower speed still has to be maintained - _in_spite_of_ the new speed change being valid for head of train. But anyway, the train has also to remember the new speed, so it cannot ignore the new speed change.

This is only another (but less obvious) variant of the very common case where a train is in one speed restriction with it’s end and already in a next speed restriction with it’s front, so never enjoying the raised speed between both restrictions… A case which all drivers of the world must handle.

So, no reason to worry about the trainRelation="headOfTrain” aspect.

<\END OF DOCUMENTATION HINT>

** The signalised attribute indicates whether the speed aspects is shown
next to text with some light signal or a panel (true) or only in the
"drivers timetable" (false).

We would need the same Boolean value for the fact whether the speed change is pre-signalised. Normally, the question of the pre-signalisation is more important (for decreasing speed changes) than the question of the in-place signalisation. The in-place signalisation is relevant for increasing speed

changes, so we do need both attributes.

<DOCUMENTATION HINT>

To anticipate possible questions: A speed change is treated as not pre-signalised if there is at least one route at which a train can pass the speed change decreasing and did not pass any pre-signalisation for the speed reduction a proper braking distance before.

<\END OF DOCUMENTATION HINT>

> ** What to do with the current trainCategory attribute?

With the discussions earlier this year, nobody seams to need it. As far as I know, with the typical usage of train categories nowadays, they have nothing to do with technical (speed) matters. It would need a trainCategory with much more technical background than the categories of RailML.

I would prefer to declare it deprecated.

> ** How to define the "blocking of a track"?** Using a time restricted
> speed profile with vMax="0" in the
> speedChange elements?

From my experience, blockings have nothing to do with speed restrictions. We should consider a new ‘list of blockings’ in future, probably rather in TimeTable subschema. If we mix blockings and speed restrictions, it becomes much more difficult here…

> ** How to define an "obligational stop" where all or only certain trains
> have to stop prior going on with the same speed aspect as before?
> ** Could this be an attribute of the speedChange element?

That is an issue I would like to see solved with RailML 2.2. I would prefer either such an attribute (obligational stop of head/end of train).. The other, possible “cleaner” but also more complex solution would be to define an infrastructure element “obligationalStop” (independently from speed changes).

I think the attribute at a speedChange will be ok.

** Do we need the midOfTrain value as trainRelation in the speedChange context?

I think so (see news entry from 26.04.2012 on ‘train relation’):
Especially in very ‘rough’ infrastructure modelling, with stations being only a deterministic point at a line and trains being only a ‘point of mass’, the default understanding is that the mid of a train arrives/departs at the mid of a station. If the train arrives from a line with a different speed than that of the line it departs, consequently the speed change lies exactly at the station place and is valid for the mid of the train.

(Additionally, but less general, in Germany we have speed changes officially valid “until scheduled stopping place” (planmäßiger Halteplatz). If a stopping place is “mid of train = mid of platform”, you consequently need a speed change valid for mid of train and placed at the mid of the platform.)

Best regards,
Dirk.

Subject: Required 'id' attributes (was: More detailed 'speed change' definitions)
Posted by Susanne Wunsch railML on Wed, 07 Nov 2012 13:25:11 GMT
View Forum Message <> Reply to Message

Hello Dirk and others,

The following aspect was quite overseen, sorry for that.

I changed the follow-up to railML.misc because of the more general topic.

Dirk Bräuer <dirk.braeuer@irfp.de> writes:
> <speedChange id="sc3"...>
>
> Can we have the ‚id’ of a speed change as an optional attribute? Since
> we have many, many speed changes and seldom the need to refer to them
> I do (optionally) want to save some space and effort.

* The "id" attribute provides good facilities to fetch single elements (with its attributes and sub-elements) from an XML file irrespective of an existing reference in the XML file.

* There are other often-used XML standards that use the same mechanism:

e.g. GML (Geographic Markup Language)

<attribute name="id" type="ID">
<annotation>

```
    <documentation>The attribute gml:id supports provision of a handle
    for the XML element representing a GML Object. Its use is
    mandatory for all GML objects. It is of XML type ID, so is
    constrained to be unique in the XML document within which it
    occurs.</documentation>
   </annotation>
  </attribute>

  <complexType name="AbstractGMLType" abstract="true">
   <sequence>
    <group ref="gml:StandardObjectProperties"/>
   </sequence>
   <attribute ref="gml:id" use="required"/>
  </complexType>
```

This attribute is required for all GML objects irrespective any reference.

A huge amount of XML Schemas in the Geographic field of application base on the above mentioned XML Schema.

* It is already a very smart implementation for identification.

  The following example shows up a typical alternative from the banking sector:

  <act:id type="guid">aa671349a0befafcfff1773e42eb8a49</act:id>

* There are algorithms for different programming languages for the ID-generation in general.

  e.g. generate-id() in XPath (staying at the XML-world) ;-)

* The _required_ attribute may only be changed into "optional" for the next major release.

I hope it clarifies the issue a bit.

Kind regards...
Susanne

Crosspost & Followup-To: railML.misc

--
Susanne Wunsch
Schema Coordinator: railML.common

Subject: Re: Required 'id' attributes (was: More detailed 'speed change' definitions)
Posted by on Thu, 08 Nov 2012 20:06:11 GMT
View Forum Message <> Reply to Message

Dear Susanne,

thank you for the reply.

Some of the aspects on 'id's in RailML

* There are other often-used XML standards that use the same mechanism:

Yes, of course, all strict-relational data models do so.
(Strict-relational data models need to have Id's.)

In RailML, we also have some hierarchy (so RailML is not strict-relational) and, more important, we do not know about the data model of the reading/writing software. So I would say it is bad to force it to strict-relational data models.

Additionally, we do not have 'id's at each <elements>, so the selection of elements with id (out of all elements) is a rather arbitrary one. If your arguments would be accepted: Wouldn't we need to provide 'id's at all <elements>, would we?

So to come out of this current "arbitrariness" I thought about the rule "no 'id' without a 'ref' to it". So: If we need a reference, we have to provide both 'id' and 'ref'.

That would lead us to a rather logic structure: Hierarchy (<elements>) combined with non-hierarchic references with 'id's.

The question on the meaning of 'id's in RailML (as a primary key for relational data models) is a more general one - not specially on speed changes, so I will end this arguing here and open a new topic at "misc".

Best regards,
Dirk.

Subject: Logical constraints for speed profile characteristics (was: More detailed 'speed change' definitions)
Posted by Susanne Wunsch railML on Mon, 12 Nov 2012 12:07:12 GMT
View Forum Message <> Reply to Message

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

> There are some attributes of <speedProfile> which act as conditions:

> The <speedProfile> is valid if the train fulfils the condition
> (axleLoad, meterLoad, operatingPeriodRef, nationalSystem). At least
> if there are more than one such conditions, there is the question:
> Are they linked with logical AND or with logical OR?
>
> Is the speed profile valid for the train if it has an axleLoad of more
> than xxx tons AND a meter load of more than yyy tons per meter?
> Or for if the train has an axleLoad of more than xxx tons OR a meter
> load of more than yyy tons per meter?
>
> This may be left to the documentation but it is not very easy. It
> depends at least on the attribute influence=decreasing/increasing:
> Typically, the conditions of an increasing speed profile are
> necessary (AND), and the conditions of an decreasing speed profile
> are optional (OR). To clarify this, I would prefer an attribute for
> the logical link if there is more than one condition.

I filed a Trac ticket for this issue in order to clarify it within
railML 2.2:

http://trac.assembla.com/railML/ticket/194

Kind regards...
Susanne

--
Susanne Wunsch
Schema Coordinator: railML.common

Subject: Blocking of infrastructure elements (was: More detailed 'speed change' definitions)
Posted by Susanne Wunsch railML on Mon, 12 Nov 2012 12:11:44 GMT
View Forum Message <> Reply to Message

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

>> ** How to define the "blocking of a track"?** Using a time
>> restricted speed profile with vMax="0" in the
>> speedChange elements?
>
> From my experience, blockings have nothing to do with speed
> restrictions. We should consider a new ‘list of blockings’ in future,
> probably rather in TimeTable subschema. If we mix blockings and speed
> restrictions, it becomes much more difficult here…

We already filed a Trac ticket for this issue in order to clarify it
with some later release:

http://trac.assembla.com/railML/ticket/156

Kind regards...
Susanne

--
Susanne Wunsch
Schema Coordinator: railML.common

Subject: Obligational stop (was: More detailed 'speed change' definitions)
Posted by Susanne Wunsch railML on Mon, 12 Nov 2012 12:17:03 GMT
View Forum Message <> Reply to Message

Dirk Bräuer <dirk.braeuer@irfp.de> writes:
>> ** How to define an "obligational stop" where all or only certain trains
>> have to stop prior going on with the same speed aspect as before?
>> ** Could this be an attribute of the speedChange element?
>
> That is an issue I would like to see solved with RailML 2.2. I would
> prefer either such an attribute (obligational stop of head/end of
> train). The other, possible “cleaner” but also more complex solution
> would be to define an infrastructure element “obligationalStop”
> (independently from speed changes).
>
> I think the attribute at a speedChange will be ok.

This issue is implemented with the following commit:

http://trac.assembla.com/railML/changeset/456

The following attributes are added to the 'speedChange' element.

```
<xs:attribute name="mandatoryStop" type="xs:boolean">
  <xs:annotation>
    <xs:documentation>specifying the speedChange as a mandatory stop point, e.g. in front of a level crossing</xs:documentation>
  </xs:annotation>
</xs:attribute>
<xs:attribute name="mandatoryBraking" type="xs:boolean">
  <xs:annotation>
    <xs:documentation>specifying the speedChange as a mandatory braking point, e.g. in front of a descending slope</xs:documentation>
  </xs:annotation>
</xs:attribute>
```

Kind regards...

Susanne

--
Susanne Wunsch
Schema Coordinator: railML.common

Subject: Re: Obligational stop
Posted by Susanne Wunsch railML on Mon, 11 Mar 2013 13:38:56 GMT
View Forum Message <> Reply to Message

Dear railML users,

During our railML conference in Berlin a discussion about the issue of
this thread started. (Look at the end of this posting.)

Susanne Wunsch <coord@common.railml.org> writes:
> Dirk Bräuer <dirk.braeuer@irfp.de> writes:
>>> ** How to define an "obligational stop" where all or only certain
>>> trains have to stop prior going on with the same speed aspect as
>>> before?
>>> ** Could this be an attribute of the speedChange element?
>>
>> That is an issue I would like to see solved with RailML 2.2. I would
>> prefer either such an attribute (obligational stop of head/end of
>> train). The other, possible “cleaner” but also more complex solution
>> would be to define an infrastructure element “obligationalStop”
>> (independently from speed changes).
>>
>> I think the attribute at a speedChange will be ok.
>
> This issue is implemented with the following commit:
>
> http://trac.assembla.com/railML/changeset/456
>
> The following attributes are added to the 'speedChange' element.
>
> <xs:attribute name="mandatoryStop" type="xs:boolean">
> <xs:annotation>
> <xs:documentation>specifying the speedChange as a mandatory stop point, e.g. in front of a level crossing</xs:documentation>
> </xs:annotation>
> </xs:attribute>

I summarized the new arguments in the appropriate Trac ticket. [1]

During the last railML conference (2013-03-06) in Berlin the use case
for "mandatory stop" was questioned.

The described scenario is very specific to the German railway operations. In order to avoid country-specific operation rules in the context of the more general railML data exchange, the current implementation should be removed.

As a short-term solution, the partners may use any-Attributes instead.

An implementation following the idea of an operational stop would be preferred to the current implementation as a "speedChange".

Any comments* appreciated.

[1] http://trac.assembla.com/railML/ticket/100#comment:4
* +1, -1, hints, questions...

Kind regards...
Susanne

--
Susanne Wunsch
Schema Coordinator: railML.common

Subject: Re: Obligational stop
Posted by on Tue, 12 Mar 2013 19:35:49 GMT
View Forum Message <> Reply to Message

Dear Susanne and all others,

> The described scenario is very specific to the German railway
> operations. In order to avoid country-specific operation rules in the
> context of the more general railML data exchange, the current
> implementation should be removed.

You want to decline a possibility for a mandatory stop post because you think it's "too German" but you want to keep the place of "Betriebsbremsung"?

I am very much wondering. It's a joke, isn't it?

A mandatory stop post is by far not German. As I already explained in this thread earlier, they appear for instance also in many other countries. RETB stop posts in UK, for instance.

On the contrary, I do not know any other country which has something like "Betriebsbremsung". Despite being from Germany, I do not even really know what to do at these places as a driver. The background is very old and

out-dated as so many other rules in Germany... But we have it in RailML!
Very good!

I am wondering. And I am learning. Thank you for that lesson.

Didn't we allow speed profiles depending on the greater "route" (or path)
of a train despite this is not very general and physical and happens in
Switzerland only? I can remember a discussion at a RailML conference where
the Swiss members said "ok but what should we do"? So why not using
"anyAttributes" for that? I know more countries with mandatory stops by
infrastructure than with path-dependent speeds.

Well, we have to be careful not to get much of the other's countries
specialties... I will keep an eye on it!

Dirk.

Subject: Re: Obligational stop
Posted by Susanne Wunsch railML on Tue, 12 Mar 2013 21:57:47 GMT
View Forum Message <> Reply to Message

Dear Dirk and others interested,

Dirk Bräuer <dirk.braeuer@irfp.de> writes:
>> The described scenario is very specific to the German railway
>> operations. In order to avoid country-specific operation rules in the
>> context of the more general railML data exchange, the current
>> implementation should be removed.
>
> You want to decline a possibility for a mandatory stop post because
> you think it's "too German" but you want to keep the place of
> "Betriebsbremsung"?

We want to remove both attributes (mandatoryStop and mandatoryBraking)
from the "speedChange" element for the upcoming 2.2 version. The members
of the conference considered them bad placed. And indeed both scenarios
are some kind of operational-rule-driven. The consensus was to
re-investigate the use case and find a more appropriate place in the XML
tree for such kind of information.

The "mandatoryBraking" attribute, which is the topic of this thread, may
be modelled as an operational stop with a reference to its level
crossing. But this idea is also not fully checked and far from "ready to
implement".

> A mandatory stop post is by far not German. As I already explained in
> this thread earlier, they appear for instance also in many other

> countries. RETB stop posts in UK, for instance.

Thanks for remembering. The implementation for this scenario is postponed, not "deprecated". We should find another place (element/attribute) in the XML tree and evaluate the pros and cons of all found possibilities (current "speedChange" or future "stop" ...)

> On the contrary, I do not know any other country which has something
> like "Betriebsbremsung". Despite being from Germany, I do not even
> really know what to do at these places as a driver. The background is
> very old and out-dated as so many other rules in Germany... But we
> have it in RailML! Very good!

That's the reason for the other thread "Mandatory braking in front of a steep gradient". We got the feedback to look for a better place in the model for such kind of information.

Kind regards...
Susanne

--
Susanne Wunsch
Schema Coordinator: railML.common

Subject: Re: Obligational stop
Posted by on Wed, 13 Mar 2013 20:09:09 GMT
View Forum Message <> Reply to Message

Dear Susanne,

Am 12.03.2013, 22:57 Uhr, schrieb Susanne Wunsch <coord@common.railml.org>:
> We want to remove both attributes (mandatoryStop and mandatoryBraking)
> from the "speedChange" element for the upcoming 2.2 version.

Ah, I understand.

> And indeed both scenarios
> are some kind of operational-rule-driven.

The "Betriebsbremsung" more than the "mandatory stop".

So I agree to remove "Betriebsbremsung" to somewhere else, may be away from <infrastructure> to <rules> or such.

I do not agree concerning "mandatory stops". Their reason is clearly infrastructure. In the case of level crossings (the case you always quote) the reason is "bad sight" from street to railway line due to an obstacle

in the triangle between a car, a train, and the level crossing. This "obstacle" - possibly a house - is clearly infrastructure - somebody has built it there. May be it's not railway property, but rather infrastructure in general than rule.

Other examples for "mandatory stops" are at least the same "infrastructure-like": RETB stop markers are a kind of starter signal, or H-Tafel or Trapeztafel in Germany.

Of course all these also have a touch of rule: The reason for a starter signal is a rule (just one train in one section). Despite this, I guess you would treat starter signals, H-Tafel, and Trapeztafel as infrastructure, too. So you should do the same with mandatory stop marker boards.

Another example would be Ra10 / Rangierhalttafel from Germany (limit of shunting marker board in English). Is it infrastructure or rule? Some of both, of course. There is no physical need to stop there, as there is no physical need to stop at any other main signal or marker board.

However, following the rule Christian once said: At least if you can touch it, it is infrastructure. You can touch a main signal, a Ra10, as well as a "mandatory stop" marker or these "0 km/h" speed signals at German level crossings with "bad sight".

Convinced?

> The "mandatoryBraking" attribute, which is the topic of this thread, may
> be modelled as an operational stop with a reference to its level
> crossing. But this idea is also not fully checked and far from "ready to
> implement".

I guess there is a mistake in your writing: You do not mean "mandatoryBraking" but "mandatoryStop".

The "mandatoryStop" has another character than an operational stop. Operational stops are by far not mandatory - on the contrary. They can be skipped (the train is allowed to run through) under certain conditions, which are pure of "timetabling" matter.

Currently, you cannot create an operational stop in RailML referencing a level crossing - stops can only reference OCPs, and a level crossing is no OCP. It would be necessary to additionally create an OCP at the place of the level crossing to model the operational stop.

Anyway, with this technology you cannot express that stops are regularly necessary forced by the infrastructure manager (or some other authority) at this place. I think it should be possible to create infrastructure-only

RaiLML file (a RailML file with just infrastructure, no trains). If this
is given to anybody who wants to create a timetable, it should tell him as
much as he could see "in nature". It should spare him to go outside and
look at each sign. If you agree with this, the "mandatory braking" marker
boards should be infrastructure.

If you do not want to put them as an attribute of <speedChange>, then
please allow a cross-reference from/to <speedChange> to keep background
information.

Best regards,
Dirk.

Subject: Re: Obligational stop
Posted by thomas.kauer on Fri, 15 Mar 2013 14:38:59 GMT
View Forum Message <> Reply to Message

Dirk BrÃ¤uer wrote:
>
> Dear Susanne,
>
> Am 12.03.2013, 22:57 Uhr, schrieb Susanne Wunsch <coord@common.railml.org>:
>> We want to remove both attributes (mandatoryStop and mandatoryBraking)
>> from the "speedChange" element for the upcoming 2.2 version.
>
> Ah, I understand.
>
>> And indeed both scenarios
>> are some kind of operational-rule-driven.
>
> The "Betriebsbremsung" more than the "mandatory stop".
>
> So I agree to remove "Betriebsbremsung" to somewhere else, may be away
> from <infrastructure> to <rules> or such.
>
> I do not agree concerning "mandatory stops". Their reason is clearly
> infrastructure. In the case of level crossings (the case you always quote)
> the reason is "bad sight" from street to railway line due to an obstacle
> in the triangle between a car, a train, and the level crossing. This
> "obstacle" - possibly a house - is clearly infrastructure - somebody has
> built it there. May be it's not railway property, but rather
> infrastructure in general than rule.
>
> Other examples for "mandatory stops" are at least the same
> "infrastructure-like": RETB stop markers are a kind of starter signal, or
> H-Tafel or Trapeztafel in Germany.
>

> Of course all these also have a touch of rule: The reason for a starter
> signal is a rule (just one train in one section). Despite this, I guess
> you would treat starter signals, H-Tafel, and Trapeztafel as
> infrastructure, too. So you should do the same with mandatory stop marker
> boards.
>
> Another example would be Ra10 / Rangierhalttafel from Germany (limit of
> shunting marker board in English). Is it infrastructure or rule? Some of
> both, of course. There is no physical need to stop there, as there is no
> physical need to stop at any other main signal or marker board.
>
> However, following the rule Christian once said: At least if you can touch
> it, it is infrastructure. You can touch a main signal, a Ra10, as well as
> a "mandatory stop" marker or these "0 km/h" speed signals at German level
> crossings with "bad sight".
>
> Convinced?
>
>> The "mandatoryBraking" attribute, which is the topic of this thread, may
>> be modelled as an operational stop with a reference to its level
>> crossing. But this idea is also not fully checked and far from "ready to
>> implement".
>
> I guess there is a mistake in your writing: You do not mean
> "mandatoryBraking" but "mandatoryStop".
>
> The "mandatoryStop" has another character than an operational stop.
> Operational stops are by far not mandatory - on the contrary. They can be
> skipped (the train is allowed to run through) under certain conditions,
> which are pure of "timetabling" matter.
>
> Currently, you cannot create an operational stop in RailML referencing a
> level crossing - stops can only reference OCPs, and a level crossing is no
> OCP. It would be necessary to additionally create an OCP at the place of
> the level crossing to model the operational stop.
>
> Anyway, with this technology you cannot express that stops are regularly
> necessary forced by the infrastructure manager (or some other authority)
> at this place. I think it should be possible to create infrastructure-only
> RaiLML file (a RailML file with just infrastructure, no trains). If this
> is given to anybody who wants to create a timetable, it should tell him as
> much as he could see "in nature". It should spare him to go outside and
> look at each sign. If you agree with this, the "mandatory braking" marker
> boards should be infrastructure.
>
> If you do not want to put them as an attribute of <speedChange>, then
> please allow a cross-reference from/to <speedChange> to keep background
> information.

>
> Best regards,
> Dirk.
>
>
Dear Dirk

I agree that if there is a "mandatory braking marker" this should be part of the infrastructure. So it should be treated as a marker (a kind of signal). A lot of speed changes have their origins in a amrker or some other kind of signal - a cross-reference would very well fit for that need.
If the "mandatory braking" has no marker but is only written somewhere in operational rules you would have to make a difference between "general" rules for all trains and "timetable specific" rules that may only be applied by some railway companies running there.
But I don't think you need a <speedChange> for a "mandatory braking marker" since the resulting speed is depending on the exact breaking rules and train properties, so you normally won't be able to give any conrete speeds at so a <speedChance>.

Best regards,
Thomas

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

Subject: Re: Blocking of infrastructure elements
Posted by Christian Rahmig on Mon, 02 Dec 2013 10:43:20 GMT
View Forum Message <> Reply to Message

Hello everybody,

just for completeness, please refer to [1] for the further discussion of the topic.

[1]
http://www.railml.org/forum/ro/index.php?group=1&offset= 0&thread=70&id=324

Best regards

--
Christian Rahmig
railML.infrastructure coordinator

Am 12.11.2012 13:11, schrieb Susanne Wunsch:
> Dirk Bräuer <dirk.braeuer@irfp.de> writes:

>
>>> ** How to define the "blocking of a track"?** Using a time
>>> restricted speed profile with vMax="0" in the
>>>   speedChange elements?
>>
>>   From my experience, blockings have nothing to do with speed
>> restrictions. We should consider a new ‘list of blockings’ in future,
>> probably rather in TimeTable subschema. If we mix blockings and speed
>> restrictions, it becomes much more difficult here…
>
> We already filed a Trac ticket for this issue in order to clarify it
> with some later release:
>
> http://trac.assembla.com/railML/ticket/156
>
> Kind regards...
> Susanne
>

Subject: Mandatory stops e.g. before a level crossing (was: Obligational stop)
Posted by christian.rahmig on Wed, 05 Apr 2017 11:48:19 GMT
View Forum Message <> Reply to Message

Dear all,

as there has not been a final solution to the topic, I want to
re-animate the discussion here in the forum:

Am 11.03.2013 um 14:38 schrieb Susanne Wunsch:
> [...]
>
> I summarized the new arguments in the appropriate Trac ticket. [1]
>
>   During the last railML conference (2013-03-06) in Berlin the use case
>   for "mandatory stop" was questioned.
>
>   The described scenario is very specific to the German railway
>   operations. In order to avoid country-specific operation rules in the
>   context of the more general railML data exchange, the current
>   implementation should be removed.
>
>   As a short-term solution, the partners may use any-Attributes instead.
>
>   An implementation following the idea of an operational stop would be
>   preferred to the current implementation as a "speedChange".
>
> Any comments* appreciated.

>
> [1] http://trac.assembla.com/railML/ticket/100#comment:4
> * +1, -1, hints, questions...
>

The main question to be answered:
Shall a mandatory stop e.g in front of a level crossing be modelled
using the current solution with <speedChange>@mandatoryStop [boolean] or
shall we define an operational point there?

I would like to get this topic clarified with the upcoming version 2.4.

Best regards
Christian

--
Christian Rahmig - Infrastructure scheme coordinator
railML.org (Registry of Associations: VR 5750)
Phone Coordinator: +49 173 2714509; railML.org: +49 351 47582911
Altplauen 19h; 01187 Dresden; Germany www.railml.org

Subject: Re: Mandatory stops e.g. before a level crossing
Posted by on Thu, 18 May 2017 16:56:32 GMT
View Forum Message <> Reply to Message

Dear Christian,

why do you think that there is no final solution? Possibly all other
thought there is:

- Mandatory stops shall be regarded as infrastructure at least if there
is a sign at the place of the stop. (In every case I know, there is one.)

- Mandatory stops can be modelled with a <speedChange> with speed=0. We
already do so. Where is the problem?

- German "Betriebsbremsung" is not infrastructure, it's <rules> or such.
It can be closed here (as you wrote in your other post).

With best regards,
Dirk.

Subject: Re: Mandatory stops e.g. before a level crossing
Posted by christian.rahmig on Mon, 12 Feb 2018 12:39:17 GMT
View Forum Message <> Reply to Message

Dear Dirk,

thank you for your input. Following the whole discussion here in the forum and subject in Trac ticket #100 [1] for railML 2.4 I suggest the following solution:

If the mandatory stop is indicated by a physical panel/sign at the place of the stop, use:

<speedChange>@mandatoryStop="true"

The <speedChange> shall be referenced from the panel/sign modelled as <signal>:

<signal><speed><speedChangeRef>@ref

If the mandatory stop is not physically visible by a panel/sign, please model it as an OCP:

<ocp><propOperational>@operationalType="stoppingPoint"

For railML 3.x I suggest to have a more flexible solution with a separate element for marking stopping places for railway vehicles that can be referenced from different other objects.

I hope this proposed solution fits to everyone's needs and use cases.

[1] http://trac.assembla.com/railML/ticket/100

Best regards
Christian

Am 18.05.2017 um 18:56 schrieb Dirk Bräuer:
> Dear Christian,
>
> why do you think that there is no final solution? Possibly all other
> thought there is:
>
> - Mandatory stops shall be regarded as infrastructure at least if there
> is a sign at the place of the stop. (In every case I know, there is one.)
>
> - Mandatory stops can be modelled with a <speedChange> with speed=0. We
> already do so. Where is the problem?
>
> - German "Betriebsbremsung" is not infrastructure, it's <rules> or such.
> It can be closed here (as you wrote in your other post).
>

> With best regards,
> Dirk.
>

--
Christian Rahmig - Infrastructure scheme coordinator
railML.org (Registry of Associations: VR 5750)
Phone Coordinator: +49 173 2714509; railML.org: +49 351 47582911
Altplauen 19h; 01187 Dresden; Germany www.railml.org

Subject: Re: Mandatory stops e.g. before a level crossing
Posted by on Wed, 14 Feb 2018 09:38:59 GMT
View Forum Message <> Reply to Message

Dear Christian,

I agree with your summary with the exception of one big attention:

> If the mandatory stop is not physically visible by a panel/sign, please model it as an OCP:
> <ocp><propOperational>@operationalType="stoppingPoint"

This is clearly not the original intention of an <ocp> nor of "stoppingPoint". It could lead to several further problems and misunderstandings. So, I would best call it a "work-around".

An <ocp> is originally intended as a container for "timetable measuring places", e. g. stations and all other "aggregated" infrastructure places which are usually show in timetables. On the contrary, a non-physical mandatory stop will not necessarily be shown in timetables. In such cases, there will be no passing/stopping times and highly probably no "register" and all that. So why using an <ocp> then at all? A non-physical mandatory stop with no timetable relevance is purely a rule, and as long as we have no <railML><rules> (and no demand for such a modelling), it should not be modelled in railML.

However, we can agree that _if_ a non-physical mandatory stop is to be shown in timetables, an <ocp> is the right solution.

So, I want to ask you not to recommend the usage of "stoppingPoint". In cases where a modelling of non-physical mandatory stops in railML 2.x is unavoidable, it can be acceptable as a work-around but even then, I would recommend not to use "stoppingPoint" but rather a place-holder-only, an "empty" <ocp>.

Best regards,
Dirk.

Subject: Re: Mandatory stops e.g. before a level crossing

Dear Dirk,

Am 14.02.2018 um 10:38 schrieb Dirk Bräuer:
> Dear Christian,
>
> I agree with your summary with the exception of one big attention:
>
>> If the mandatory stop is not physically visible by a panel/sign, please model it as an OCP:
>> <ocp><propOperational>@operationalType="stoppingPoint"
>
> This is clearly not the original intention of an <ocp> nor of "stoppingPoint". It could lead to several further problems and misunderstandings. So, I would best call it a "work-around".
>
> An <ocp> is originally intended as a container for "timetable measuring places", e. g. stations and all other "aggregated" infrastructure places which are usually show in timetables. On the contrary, a non-physical mandatory stop will not necessarily be shown in timetables. In such cases, there will be no passing/stopping times and highly probably no "register" and all that. So why using an <ocp> then at all? A non-physical mandatory stop with no timetable relevance is purely a rule, and as long as we have no <railML><rules> (and no demand for such a modelling), it should not be modelled in railML.
>
> However, we can agree that _if_ a non-physical mandatory stop is to be shown in timetables, an <ocp> is the right solution.
>
> So, I want to ask you not to recommend the usage of "stoppingPoint". In cases where a modelling of non-physical mandatory stops in railML 2.x is unavoidable, it can be acceptable as a work-around but even then, I would recommend not to use "stoppingPoint" but rather a place-holder-only, an "empty" <ocp>.

Thank you very much for your feedback. My adapted proposal for solution would be the following:

If the mandatory stop is indicated by a physical panel/sign at the place of the stop, use:

<speedChange>@mandatoryStop="true" together with
<speedChange>@signalised="true"

The <speedChange> shall be referenced from the panel/sign modelled as <signal>:

<signal><speed><speedChangeRef>@ref

If the mandatory stop request is not physically visible by a panel/sign, use:

<speedChange>@mandatoryStop="true" together with
<speedChange>@signalised="false"

I hope you can agree with that solution?

Best regards
Christion

--
Christian Rahmig - Infrastructure scheme coordinator
railML.org (Registry of Associations: VR 5750)
Phone Coordinator: +49 173 2714509; railML.org: +49 351 47582911
Altplauen 19h; 01187 Dresden; Germany www.railml.org

Subject: Re: Mandatory stops e.g. before a level crossing
Posted by Thomas Nygreen JBD on Mon, 04 Jun 2018 10:41:53 GMT
View Forum Message <> Reply to Message

As this is not a feature in the Norwegian infrastructure, I just have a question out of curiosity: With a speed change of 0, how are trains able to move past it after stopping?

Subject: Re: Mandatory stops e.g. before a level crossing
Posted by christian.rahmig on Mon, 04 Jun 2018 11:00:58 GMT
View Forum Message <> Reply to Message

Dear Thomas,

the mandatory stop is considered being done immediately after the train stopped. Right afterwards, the train is allowed to proceed with line speed. So, in contrast to other speed changes that are usually valid until the next one, the mandatory stop shall be interpreted as a single occuring event.

Best regards
Christian

Am 04.06.2018 um 12:41 schrieb Thomas Nygreen:
> As this is not a feature in the Norwegian infrastructure, I
> just have a question out of curiosity: With a speed change
> of 0, how are trains able to move past it after stopping?

Subject: Re: Mandatory stops e.g. before a level crossing
Posted by on Mon, 03 Sep 2018 17:28:11 GMT

I must complement Christians answer a bit: It is not such a special exception as Christian writes.

Theoretically, a "speed change" which is a mandatory stop could be encoded as:
- one speed <speedChange> with @speed=0,
- one more <speedChange> with @speed>0 and @trainRelation='headOfTrain' at the same position.

This would be logical (at least as long as the order is out of question) and no exception. This is how we write it into driver's timetables.

But for railML, we did want to avoid having two <speedChange>s (of the same speed profile) at the same position.

So, we introduces <speedChange>@mandatoryStop="true". Such a <speedChange>@mandatoryStop="true" implies the two above mentioned <speedChange>s and avoids speed=0.

Best regards,
Dirk.