

---

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:tNationalSystemsType"  
 >> 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>  
> referring 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 referring 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?file=tl\\_files/railML.org/documents/events/slides/2009-10-06\\_tud\\_resden\\_albrecht-interlocking.pdf](http://www.railml.org/web/index.php/previous-events.html?file=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

>