

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 referring 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

---