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