Subject: Re: Platforms and ramps for railML 2.2
Posted by on Mon, 26 Mar 2012 10:49:35 GMT
View Forum Message <> Reply to Message

Hallo Christian and all others,

your suggestion about platforms sounds good. Especially about
> - The associated OCP can be de-referenced from the OCP's <trackGroup>
I think this is very important and not to be missed.

Concerning
> - name: the name contains the platform number (e.g. "3")
we have to take into account that, in some countries like Czech Republic,
to distinguish between a platform name and a 'track at platform' name (in
German we would say "Bahnsteigkante"). I am not aware whether the 'track
at platform' name is always the same as the operational track number
(which would make it easier). But since you want to assign the
<serviceSection> to a certain track, the Czech principle means that there
will be two <serviceSections> sharing the same name (which then means that
they share the same physical platform at opposite sides).

Additionally, I want to remember to Susanne's post from 2011-11-04 where
she pleads for a stop markers (Haltetafeln) in RailML. I think that the
following has to be possible:
- to link a platform with an ocp (as you have already written),
- to link stop markers with the associated platform,
- to link a train's stop with a stop marker and therefore with a platform,
- to define the relation between a train position (front, end, first
door...) and the stop marker (and therefore the platform).
All these links shall be done by references (meaning syntactically forced
as long as these information are provided in a RailML file at all).

To make it more easier in RailML, I think that it is not important to have
a general base type for platforms, ramps and maintenance facilities. A
platform is (in my opinion) such a familiar phenomen to all railways that
we can provide a special type for it.

All in all, my recommendation is:
- to define an element called 'platform marker' with the attributes (you
suggested) 'length', 'direction', 'height', 'platformName',
'platformTrackName', 'validForTrainLenghs'. The marker may be in the
middle of the platform so that we may have two lengths and two directions.
With 'validForTrainLenghs' I mean the typical additional plates with "100
m" or so written on it.
- to define an optional reference from an ocpTT to a 'platform marker'
with the relation of a position in the train (front, end, first door...)..

This leaves the ramps, loading- and maintenance facilities for other elements so far.

We still have the problem Susanne wrote, where several coupled train parts
may have inconsistent 'platform relations' (e. g. The first train part
says 'I want to stop with my front at the end of the platform' and the
last train part says 'I want to stop with my end at the beginning of the
platform'. This may be functional if the platform length is the same as
the train length but if not... it creates at least a gap in the train or,
more worst, a gap in the space-time-continuum.)

However, we shall consider that train parts may have different operating
days. So it may definitely make sense if all train parts want to stop at
the end of the platform because there may be days when each of the train
parts is the first one in the train. Considering this, we shall leave the
inconsistent cases to the reading software - we cannot save us from
everything and there are more worst possible inconsistencies in RailML.

With best regards,
Dirk.

Am 22.03.2012, 22:19 Uhr, schrieb Christian Rahmig
<coord@infrastructure.railml.org>:

> Hello everyone,
>
> considering the Trac ticket #122 [1], we are thinking about the
> introduction of a datatype for platforms and ramps with the next release
> 2.2. So, this is how it may look like:
>
> 1. <trackElements> will be extended with a new element named
> <serviceSection> for modelling platforms, ramps and related facilities..
>
> 2. The new element <serviceSection> describes the part ("section") of a
> track, which can be used for the exchange of passengers, goods or
> similar.
>
> 3. Attributes for this new element include:
>   - position information: defines the starting position and direction
> of the serviceSection
>   - length: the section length which is defined as the actually usable
> length (NOT the physically existing platform/ramp which can be longer)
>   - height: the objects's height in mm above rails (for feasible types
> only)
>   - type: enumeration of
>   "platform" (passanger platform)
>   "ramp" (ramp for loading / unloading goods)

```
>         "maintenance" (maintenance facilities e. g. in a depot)
>         "loadingFacility" (Goods can be (un-)loaded from the wagon's
>  top / underfloor)
>         "cleaning" (washing facility for cars and engines)
>         "fueling" (facility for re-fuelling of engines)
>         "parking" (section for parking of rolling stock)
>         "preheating" (electricity or steam for pre-heating purposes)
>         "other"
>     - side: right, left (seen in positive mileage direction)
>
>  4. Each <serviceSection> can have multiple types. Tracks with platforms
>  on both sides need two <serviceSection> elements to force the definition
>  of different platform names.
>
>  5. Additional semantics:
>     - name: the name contains the platform number (e.g. "3")
>     - The associated OCP can be de-referenced from the OCP's <trackGroup>
>
>  It would be nice to hear your opinion about this model approach. Are
>  there any important parameters missing?
>
>  Best regards.
>
>  [1] https://trac.assembla.com/railML/ticket/122
>
>  ---
>  Christian Rahmig
>  railML.infrastructure coordinator
```

--
Erstellt mit Operas revolutionärem E-Mail-Modul: http://www.opera.com/mail/