
Subject: Re: Steckenunterbruch/line blocking

Posted by [Susanne Wunsch railML](#) on Thu, 17 Jan 2013 17:20:16 GMT

[View Forum Message](#) <> [Reply to Message](#)

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

>> for getting a connection to another forum, it is better to use the
>> followup-tag.

>

> I don't know what a follow-up tag is.

Better to call it "Followup-To":

<http://en.wikipedia.org/wiki/Crossposting>

<http://de.wikipedia.org/wiki/Crossposting>

>> However, since the "disabled" sub-element will be available for all
>> relevant tracks, there is no need to define a length of the blocking
>> section in form of a "from-to" attribute group.

>

> But then there would be no possibility do "block a part of an
> <element>" such as part of a track (sub-section of a track).

I just implemented the new element <state> with an attribute 'disabled' of type xs:boolean. It may be constrained with the attribute 'operatingPeriodRef' referring to an operatingPeriod/@id from the timetable subschema. Furthermore it may be constrained to a part of a <track> by relative positions.

Example (assume 'operatingPeriod' are defined in the same file):

```
<track id="t1">
  <states>
    <state disabled="true" operatingPeriodRef="op1" remarks="blah">
      <from pos="250"/>
      <to pos="1340"/>
    </state>
    <state disabled="true" operatingPeriodRef="op2">
      <from pos="8900"/>
    </state>
  </states>
  <trackTopology>
    <trackBegin id="tb1" pos="0">
      <macroscopicNode ocpRef="o1"/>
    </trackBegin>
    <trackEnd id="te1" pos="10000">
      <macroscopicNode ocpRef="o2"/>
    </trackEnd>
  </trackTopology>
</track>
```

```
</trackTopology>
</track>
```

There are some assumptions, that should be documented in the wiki:

- * If no "from" is defined, it begins at the "trackBegin".
- * If no "to" is defined, it begins at the "trackEnd".
- * If no "operatingPeriodRef" is defined, it is valid for all data of the railML file.
- * For all other times (out of the referred operating period) the defined track (section) is enabled/disabled depending on the "disabled" value.
- * If no "state" element is defined, the "track" is usable. That means 'disabled="false"'.

For further constraints use the xs:any element or the anyAttribute.

Currently the <from> and <to> elements may additionally refer to an 'ocp' via an 'ocpRef' attribute. Maybe that should be dropped because of redundancy reasons.

But otherwise that may be helpful if the exact blocking locations (relative positions) are known but differ from the ocp locations. The ocp references may be used as a hint, not overwriting the exact locations.

See ticket [1] and last issue-related implementation [2].

Please test the new structure for your needs and give us a feedback.

- > A typical problem (to solve here) is "line/track is blocked from
- > ... to ..." with from/to being mileages or stations.

The 'line blocking' has to be defined through the 'track blocking'.

- >> Here, I suggest to just implement a reference from the <disabled>
- >> element to an operating period.
- >
- > Such a reference would - as far as I know - the first time we would
- > create such a "forward-reference". Forward in the meaning of "from
- > infrastructure to timetable". We already have many references from
- > timetable to infrastructure which are "natural" since one `_first_`
- > needs infrastructure `_before_` one can operate a train.

We already put such a "forward-reference" into the infrastructure subschema with the implementation of speed profiles.

The TSR (temporary speed restrictions, de:Langsamfahrstellen) also

refer to an 'operatingPeriod' from the timetable subschema.

In most use cases a software has to multiply selectively import the railML file in order to semantically validate the data. Or the other way around, a software imports the whole railML file at once and cross-checks the internal data structures afterwards.

From this railML version on, the sequential selectively import does not work anymore, because of both reference directions (from TT to IS and from IS to TT).

> To have both directions, I would think is like "circular references"
> which sometimes can become problematic in informatics. In my opinion,
> a software first has to import <infrastructure> before it can import
> <timetable>. So it wouldn't be able to dissolve the references from
> infrastructure to timetable when importing.

For a future version we would like to move the operating periods to the common part of railML, in order to provide a better "straight-forward" structure. But this is a change for the next major release. ;-)

I created a ticket for this issue. [3]

[1] <http://trac.assembla.com/railML/ticket/156>

[2] <http://trac.assembla.com/railML/changeset/528>

[3] <http://trac.assembla.com/railML/ticket/215>

Kind regards...

Susanne

By the way, this need was already required three years ago. ;-)

<http://trac.assembla.com/railML/ticket/34>

--

Susanne Wunsch

Schema Coordinator: railML.common
