Subject: [railML3] Validities and holidays
Posted by Milan Wölke on Tue, 19 Apr 2022 09:12:59 GMT
View Forum Message <> Reply to Message

Hi all,

in unserer letzten railML3 Telko haben wir uns auf ein neues, gegenüber railML 2.x vereinfachtes, Konzept zur Abbildung von Feiertagsregelungen geeinigt. Dieses möchte ich Ihnen hier gern vorstellen. Es ist zwar nicht mehr viel Zeit bis zum Release, aber falls Ihr Hinweise oder Anmerkungen habt, werde ich gerne versuchen diese noch zu berücksichtigen.

Also folgendes Konzept: Wie bisher auch ist es möglich Feiertage - auch solche, die sich über mehrere Tage erstrecken - an der <timetablePeriod> (oder wie sie nun heißt <timePeriod>) zu hinterlegen. Damit ist spezifiziert, wann im betrachteten Zeitraum welche Feiertage liegen. An jeder Validity (analog zur <operatingPeriod> aus railML 2.x) können Wochenpattern angegeben werden, um damit zu kommunizieren, auf welcher Basis die Bitmaske, die ja beschreibt an welchen Tagen etwas stattfindet und wann nicht, entstanden ist. An dieser Stelle ist es nun möglich, über ein einfaches Enum anzugeben, ob Züge, an Feiertagen, die in diese Bitmaske fallen, fahren, oder nicht. Mit ein paar Beispielen wird das vermutlich klarer:

```
<validity id="validity-examle-1">
 <name name="Daily, even holidays" language="en"/>
 <name name="Täglich, selbst an Feiertagen" language="de"/>
 <bitmaskValidity bitmask="11111111111111111111111111111" fromDate="2022-04-19">
  <weekPatterns>
   <weekPattern pattern="1111111" publicHolidayMode="runsOnIndicatedHolidays">
   </weekPattern>
  </weekPatterns>
 </bitmaskValidity>
</validity>
<validity id="validity-examle-2">
 <name name="Daily, except on holidays" language="en"/>
 <name name="Täglich, außer an Feiertagen" language="de"/>
 <bitmaskValidity bitmask="11111111111111111111111111111" fromDate="2022-04-19">
  <weekPatterns>
   <weekPattern pattern="1111111" publicHolidayMode="neverRunsOnHolidays">
   </weekPattern>
  </weekPatterns>
 </bitmaskValidity>
</validity>
<validity id="validity-examle-3">
 <name name="Only on holidays" language="en"/>
 <name name="Nur an Feiertagen" language="de"/>
 <bitmaskValidity bitmask="00000000000000000000000000000" fromDate="2022-04-19">
  <weekPatterns>
   <weekPattern pattern="0000000" publicHolidayMode="runsOnAllHolidays">
   </weekPattern>
```

```
  </weekPatterns>
 </bitmaskValidity>
</validity>
```

Es gibt 3 mögliche Werte für das Enum 'publicHolidayMode'; runsOnAllHolidays, runsOnIndicatedHolidays und neverRunsOnHolidays.

Mit runsOnAllHolidays kann angegeben werden, dass ein Zug, unabhängig von der 7-Bit-Bitmaske, an allen in der <timePeriod> angegebenen Feiertagen fährt (validity-example-3).

Mit runsOnIndicatedHolidays kann mitgeteilt werden, dass ein Zug gemäß der 7-Bit-Bitmaske fährt, unabhängig davon, ob es sich um einen Feiertag handelt oder nicht (validity-example-1).

Und mit neverRunsOnHolidays kann man sagen, dass ein Zug an allen Tagen fährt, die durch die 7-Bit-Bitmaske angegeben sind, außer an denen, die in der <timePeriod> als Feiertag gekennzeichnet sind (validity-example-2).

In einer späteren Version von railML 3 wollen wir auch mehrere Listen von Feiertagen unterstützen, um regionale Feiertage besser zu unterstützen.

Ich hoffe diese kurze Einführung in die Abbildung von Feiertagsregeln war soweit verständlich, wenn nicht, antworte ich gern auf Fragen. Lassen Sie mich wissen, was Sie davon halten.

At our last railML3 meeting, we agreed on a new concept for modelling public holiday regulations that is simpler than railML 2.x. I would like to present it to you here. I would like to present it to you here. There is not much time left until the release, but if you have any hints or comments, I will gladly also take them into account.

The concept is as follows: As before, it is possible to store holidays - even those that extend over several days - at the <timetablePeriod> (or as it is now called <timePeriod>). This specifies which public holidays take place at which dates in the period under consideration. At each validity (analogous to <operatingPeriod> from railML 2.x), week patterns can be specified in order to communicate on which basis the bit mask, which describes on which days something takes place and when not, was created. At this point, it is now possible to specify via a simple enum whether trains run on holidays that fall within this bitmask or not. With a few examples, this will probably be more clear:

```
<validity id="validity-examle-1">
 <name name="Daily, even holidays" language="en"/>
 <name name="Täglich, selbst an Feiertagen" language="de"/>
 <bitmaskValidity bitmask="11111111111111111111111111111" fromDate="2022-04-19">
  <weekPatterns>
   <weekPattern pattern="1111111" publicHolidayMode="runsOnIndicatedHolidays">
   </weekPattern>
```

```
  </weekPatterns>
 </bitmaskValidity>
</validity>
<validity id="validity-examle-2">
 <name name="Daily, except on holidays" language="en"/>
 <name name="Täglich, außer an Feiertagen" language="de"/>
 <bitmaskValidity bitmask="11111111111111111111111111111" fromDate="2022-04-19">
  <weekPatterns>
   <weekPattern pattern="1111111" publicHolidayMode="neverRunsOnHolidays">
   </weekPattern>
  </weekPatterns>
 </bitmaskValidity>
</validity>
<validity id="validity-examle-3">
 <name name="Only on holidays" language="en"/>
 <name name="Nur an Feiertagen" language="de"/>
 <bitmaskValidity bitmask="0000000000000000000000000000" fromDate="2022-04-19">
  <weekPatterns>
   <weekPattern pattern="0000000" publicHolidayMode="runsOnAllHolidays">
   </weekPattern>
  </weekPatterns>
 </bitmaskValidity>
</validity>
```

There are 3 possible values for the enum 'publicHolidayMode'; runsOnAllHolidays, runsOnIndicatedHolidays and neverRunsOnHolidays.

With runsOnAllHolidays it can be specified that a train runs on all holidays specified in the <timePeriod>, regardless of the 7-bit bitmask (validity-example-3).

With runsOnIndicatedHolidays it can be indicated that a train runs according to the 7-bit bitmask, regardless of whether it is a holiday or not (validity-example-1).

And neverRunsOnHolidays can be used to say that a train will run on all days specified by the 7-bit bitmask, except those marked as a holiday in the <timePeriod> (validity-example-2).

In a later version of railML 3, we also intend to support multiple lists of holidays to better support regional holidays.

I hope this short introduction to the modelling of public holiday rules was understandable so far, if not, I am happy to answer questions. Let me know what you think of it.

Best regards, Milan