

Mitternachtsübergänge

Es gibt mehrere Möglichkeiten, Mitternachtsübergänge in RailML korrekt abzubilden:

1. Mit Hilfe der Attribute **arrivalDay** und **departureDay**.
2. Durch Aufteilen des Zuglaufs in Zugteile vor und nach Mitternacht sowie Wechsel der *operatingPeriod*-Referenz zwischen diesen Zugteilen
 - a) und Verwenden des Attributs **dayOffset** der nach Mitternacht gültigen *operatingPeriod*³,
 - b) und durch um einen Tag verschobene Angabe der Verkehrstage der nach Mitternacht gültigen *operatingPeriod*.

Möglichkeit 1 ist die regulär vorgesehene für Mitternachtsübergänge innerhalb eines in RailML abgebildeten Zuglaufs. Es ist ausdrücklich nicht vorgesehen, einen Zuglauf *nur wegen des Mitternachtsübergangs* in Zugteile aufzuteilen. Möglichkeit 2a kommt dennoch in besonderen des Mitternachtsübergangs vor einem (hier betrachteten) Zuglauf in Betracht. Möglichkeit 2b soll vermieden werden.

Mitternachtsübergänge innerhalb des abgebildeten Zuglaufs

Für Mitternachtsübergänge *innerhalb eines abgebildeten Zuglaufs* sind in RailML die Attribute **arrivalDay** und **departureDay** vorgesehen. Sie stellen eine Zählung der Anzahl Mitternachtsübergänge relativ zu einem Bezugspunkt (z. B. Abfahrtsort; s. u.) dar. Diese Attribute sind optional mit Default-Wert 0, d. h. sie müssen solange nicht angegeben werden, solange ein Zug noch nicht über Mitternacht gefahren ist. Nach dem ersten Mitternachtsübergang, d. h. sobald ihr Wert >0 ist, sind sie jedoch anzugeben:

Beispiel 1: Mitternachtsübergang während eines Zwischenaufenthalts

```

<ocpTT ocpRef='ocp_DOLB' ocpType='stop'
  <times scope='scheduled' arrival='23:59:49' departure='00:00:19' departureDay='1' />
  <sectionTT section='DOLB-DN E' timeref='ln_80_6212' trackRef='tr_80_6212_2' trackInfo='1'>
    <runTimes minimalTime='PT48S' operationalReserve='PT1S' />
  </sectionTT>
  <stopDescription commercial='true' stopOnRequest='false'>
    <stopTimes minimalTime='PT30S' />
  </stopDescription>
</ocpTT>

```

hier Mitternachtsübergang

Alle weiteren Ankunfts-, Abfahrts- und Durchfahrtszeiten des Zuges bis zu seinem Laufwegende sind mit *arrivalDay=1* und *departureDay=1* gekennzeichnet.

Beispiel 2: Mitternachtsübergang während der Fahrt

```

<ocpTT ocpRef='ocp_DNKW' ocpType='pass'>
  <times scope='scheduled' departure='23:55:00' />
</ocpTT>
<ocpTT ocpRef='ocp_DNKW_A' ocpType='pass'>
  <times scope='scheduled' departure='23:55:35' />
</ocpTT>
<ocpTT ocpRef='ocp_DNKO' ocpType='stop'>
  <times scope='scheduled' arrival='23:57:53' departure='23:58:23' />
</ocpTT>
<ocpTT ocpRef='ocp_DWT_N' ocpType='pass'>
  <times scope='scheduled' departure='00:01:25' departureDay='1' />
</ocpTT>
<ocpTT ocpRef='ocp_DWT' ocpType='stop'>
  <times scope='scheduled' arrival='00:02:17' arrivalDay='1' departure='00:03:00'
    departureDay='1' />
</ocpTT>

```

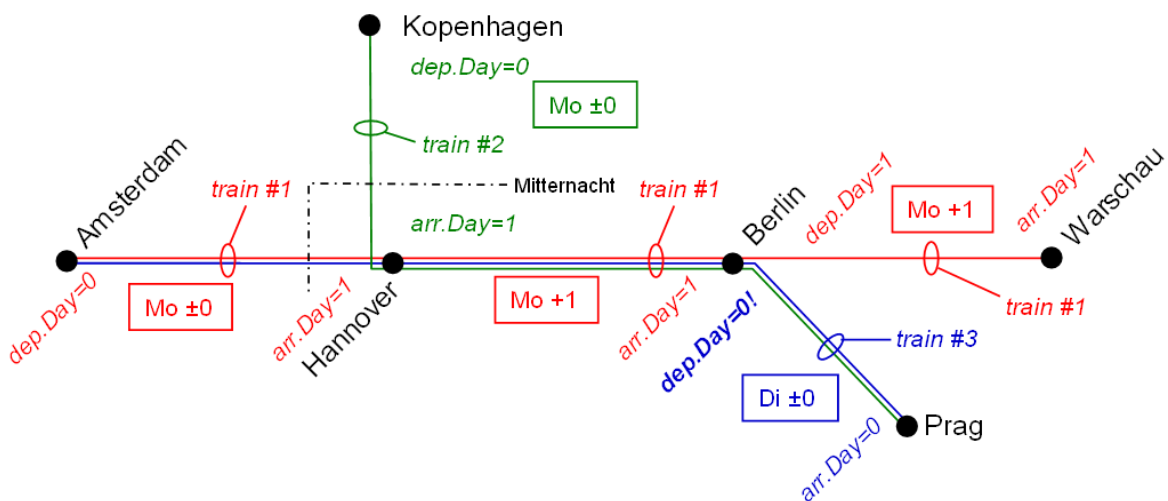
hier Mitternachtsübergang

Alle weiteren Ankunfts-, Abfahrts- und Durchfahrtszeiten des Zuges bis zu seinem Laufwegende sind mit *arrivalDay=1* und *departureDay=1* gekennzeichnet.

³ Das Attribut *dayOffset* wird mit RailML 2.2 eingeführt.

Referenzort der Tages-/Mitternachtszählung

- Es ist vorgesehen, dass die Tages- bzw. Mitternachtszählung (arrival/departureDay) **an der ersten Abfahrt eines Zuges mit 0 beginnt**.
- Demzufolge kann der Wert -1 vorkommen bei der ersten Ankunft im seltenen Fall, dass der Zug am ersten Bahnhof über Mitternacht steht und zuvor „von außerhalb“ kommt.
- Wenn ein Zug aus mehreren (zu aufeinanderfolgenden Abschnitten verketteten) Zugteilen besteht, bezieht sich die Tageszählung i. d. R. auf den ersten Abfahrtsort **des Gesamtzuglaufs**. Bei einzelnen Zugteilen kann es also vorkommen, dass diese bereits mit Tageszählung >0 beginnen.
- Wenn Zugteilläufe (in RailML: *commercialTrains*) von einem Zug auf einen anderen übergehen, kann es zu einem „Rückschritt“ der Tageszählung kommen. (Dies bedeutet, dass Zugteillauf von einem Zug, der bereits über Mitternacht gefahren ist, übergeht auf einen Zug, der noch nicht über Mitternacht gefahren ist.) Auch dies ist kein Fehler.



Im Zugteillauf der Zugteile **grün** und **blau** kommt es zum „Rücksprung“ des Tagesindex (arrival/departureDay) von 1 auf 0. Der Rücksprung geht i. d. R. mit einem scheinbaren Verkehrstageswechsel einher (operatingPeriod wechselt, effektive Verkehrstage wechseln nicht).

Bedeutung der Tages-/Mitternachtszählung

- Die Verkehrstage-Angabe (*operatingPeriod*) ist allein nicht interpretierbar, sondern immer nur im Zusammenhang mit dem jeweiligen Tagesindex (*arrival/departureDay*). Um festzustellen, wann ein Zug tatsächlich fährt, muss man z. B. die Bitmaske um die Anzahl Stellen aus Tagesindex verschieben.
- Ob ein effektiver Wechsel der Verkehrstage innerhalb des Zuglaufs stattfindet, kann nur durch Vergleichen der verschobenen Bitmasken herausgefunden werden. Ein Wechsel der *operatingPeriod* allein ist nicht zwingend ein effektiver Verkehrstageswechsel.
- Es kann beliebig viele Kombinationen aus Verkehrstage-Angabe (*operatingPeriod*) und Tagesindex (*arrival/departureDay*) geben, die inhaltlich das gleiche bedeuten.
- Das schreibende Programm ist frei in der Wahl der Kombination, diese kann auch „unterwegs“ wechseln.

Mitternachtsübergänge außerhalb des abgebildeten Zuglaufs

I. d. R. ist es beabsichtigt – bei einigen Anwendungen auch von außen her erforderlich – dass ein Zug mit Tageszählung =0 beginnt, d. h. dass sich die Verkehrstage, mit denen er identifiziert wird, auf den ersten *abgebildeten* Abfahrtsbahnhof beziehen.

Sollte der Zug bereits vor dem ersten *abgebildeten* Abfahrtsbahnhof – quasi „im Ausland“ – über Mitternacht gefahren sein, besteht damit nicht mehr die Möglichkeit, mit Tagesindex (*arrival/departureDay*) >0 am ersten abgebildeten Bahnhof zu beginnen.

Damit bliebe dann nur noch die Möglichkeit (eingangs dieses Kapitels mit 2b bezeichnet), die Verkehrstage um die Anzahl Tage versetzt anzugeben, wie oft der Zug zuvor über Mitternacht gefahren ist. Diese Lösung ist jedoch ausdrücklich nicht gewollt u. a. aus folgenden Gründen:

- Fahrplanperioden sind i. d. R. nicht geschlossen, d. h. auf den letzten Tag einer Fahrplanperiode folgt nicht wieder der erste Tag der gleichen Periode. Ein Zug, der in einer Fahrplanperiode *täglich* beginnt und über Mitternacht fährt, fährt danach genau genommen nicht mehr täglich (nämlich nicht am ersten Tag der Fahrplanperiode, dafür aber zusätzlich am ersten Tag der nächsten Fahrplanperiode). Die Bitmaske seiner Verkehrstage (*operatingPeriod.bitMask*) ist gegenüber vor Mitternacht um ein Bit in Richtung aufsteigenden Datums verschoben, d. h. selbst die zuvor nur aus Einsen bestandene Täglich-Bitmaske beginnt danach mit einer Null.
- Das um Tage verschobene Angeben der Verkehrstagereregungen (*operatingPeriods*) ist ausdrücklich zu vermeiden, da wie bereits im Kapitel *Abbildung spezieller Verkehrstagereregungen ohne Datumsbezug* erwähnt, für viele praktisch vorkommende Verkehrstagereregungen keine exakte Folgetagsentsprechung existiert (z. B. nicht für die in Deutschland recht verbreiteten „werktags außer Samstags“ und „Sonn- und Feiertags“).

Aus diesen Gründen wird mit RailML 2.2 neu das Attribut **dayOffset** im Element *operatingPeriod* eingeführt:

```
<operatingPeriods>
  <operatingPeriod id='opp_1' name='täglich' description='verkehrt täglich, auch an Feiertagen' timetablePeriodRef='ttp_
    bitMask='111..111'>
    <operatingDay operatingCode='1111111' startDate='2020-12-13' endDate='2021-12-11' />
  </operatingPeriod>
  <operatingPeriod id='opp_2' name='täglich +1' description='verkehrt täglich nach Mitternacht' timetablePeriodRef='ttp_
    bitMask='111..101' dayOffset='1'>
    <operatingDay operatingCode='1111111' startDate='2020-12-13' endDate='2021-12-11' />
  </operatingPeriod>
```

Die Bitmaske der mit *dayOffset*≠0 anzugebenden *operatingPeriods* ist nicht verschoben darzustellen, d. h. die Bitmaske ist identisch wie beim Fall *dayOffset*=0.

Grundsätzlich besteht die an sich gleichwertige Möglichkeit, in den betreffenden Fällen anstatt *dayOffset*>0 weiterhin mit *arrival/departureDay*>0 auch am ersten Bahnhof zu beginnen. Ein lesendes Programm sollte beide Varianten verarbeiten können. Die Variante mit *dayOffset*>0 ist vorgesehen für Fälle, in denen *departureDay*=0 am ersten Bahnhof aus externen Gründen erzwungen werden muss. Letzteres ist allerdings auch die Empfehlung des RailML-Konsortiums.