Subject: [railML3] Semantics of the attributes arrivalDay and departureDay
Posted by on Tue, 12 Jan 2021 18:28:32 GMT
View Forum Message <> Reply to Message

Hello All,

In railML2, the attributes 'arrivalDay' and 'departureDay' at the element <ocpTT><times> are used to define the offset between the train's operating days at the departure station and at the current operating point. This view makes it possible, among other things, to determine the exact days on which a train passes a particular operating point, even if the train runs beyond midnight. There is the semantic restriction that the value for arrivalDay/departureDay at the first operating point of the train run should always be 0.

However, during the discussion within the railML 3 TT Developer Group we came to the conclusion that this restriction is too strict. In addition to the view described above, the operating days of a train will also be used in some systems to form a primary key for a train (together with the train number). In order to create a uniqueness of trains with the identical train number, in an annual timetable with a duration of 364 days, for example, each individual train of a day must be assigned to a different operating day. In practice, however, it happens that two trains with the same train number start their journey on the same day (see examples below). If the requirement that each train run must start with an arrivalDay/departureDay of 0 were upheld, the two trains would be assigned to the same operating day. Therefore, since certain situations cannot be mapped in railML with the current regulation, we would like to allow for railML3 that values other than 0 (especially positive integers) may be specified at the first operating point of a train.

We are interested in feedback if there are positive or negative opinions and arguments about this change.

Best regards
Christian

Examples:

Example 1:

Let there be a train run Hamburg-Berlin-Munich, which runs between Hamburg and Berlin over midnight. On Monday-Friday evenings, the train is to start in Hamburg and pass through Berlin on the morning of the following day (i.e. Tuesday-Saturday). On the other days of the week, the train is to run under the same train number only from Berlin (departures there on Sunday and Monday). Since this timetable has two train runs starting on Mondays (one in Berlin in the morning, one in Hamburg in the evening), it is not possible to uniquely identify a train on the basis of the operating day at the departure station. To solve this problem, the trains starting in Berlin would have to be assigned to the day before their actual departure, i.e. Saturday and Sunday. In order for these trains to keep their correct departure operating days in Berlin, these journeys must start with a departureDay of 1.

Example 2:

A train that regularly (daily) runs late in the evening is scheduled so far later on one day that it does not start until the following day, e.g. due to construction work. If the primary key of the train is composed of the train number and the operating day at the departure station, it would (unintentionally) change its primary key due to this late running. If the train is to keep its primary key even after the time shift, its departure on the following day must be defined by a departureDay of 1 at the departure station.

German Version:
=====================

Hallo zusammen,

In railML2 werden die Attribute 'arrivalDay' und 'departureDay' am Element <ocpTT><times> benutzt, um den Versatz zwischen den Verkehrstagen des Zuges am Abgangsbahnhof und der aktuellen Betriebsstelle zu definieren. Diese Sichtweise ermöglicht es u.a., die genauen Tage zu ermitteln, an denen ein Zug eine bestimmte Betriebsstelle passiert, auch wenn der Zug über Mitternacht verkehrt. Für die Verwendung existiert die semantische Einschränkung, dass der Wert für arrivalDay/departureDay an der ersten Betriebsstelle des Zuglaufs immer 0 sein sollte.

Im Rahmen der Diskussion innerhalb der railML 3 TT Developer Group kamen wir jedoch zu dem Ergebnis, dass diese Restriktion zu streng ist. Neben der oben beschriebenen Sichtweise werden die Verkehrstage eines Zuges (zusätzlich zur Zugnummer) in manchen Systemen auch zur Bildung eines Primärschlüssels für einen Zug herangezogen werden. Um eine Eindeutigkeit von Zügen mit der selben Zugnummer herzustellen, muss dabei beispielsweise in einem Jahresfahrplan mit einer Dauer von 364 Tagen jeder Einzelzug eines Tages einem unterschiedlichen Verkehrstag zugeordnet werden. Praktisch kommt es aber vor, dass zwei Züge der selben Zugnummer am selben Tag ihre Fahrt beginnen (siehe Beispiele unten). Würde die Forderung aufrecht erhalten, dass jeder Zuglauf mit einem arrivalDay/departureDay von 0 beginnen muss, so würden die beiden Züge demselben Verkehrstag zugeordnet werden. Da sich also gewisse Situationen mit der derzeitigen Regelung nicht in railML abbilden lassen, möchten wir für railML3 zulassen, dass an der ersten Betriebsstelle eines Zuges auch andere Werte als 0 (insbesondere positive Ganzzahlen) angegeben werden dürfen.

Wir sind an Rückmeldungen interessiert, ob es positive oder negative Meinungen und Argumente zu dieser Änderung gibt.

Viele Grüße
Christian

Beispiele:

Beispiel 1:

Gegeben sei ein Zuglauf Hamburg-Berlin-München, der zwischen Hamburg und Berlin über Mitternacht verkehrt. Montag-Freitag abends soll der Zug in Hamburg beginnen und Berlin am Morgen des Folgetags passieren (also Dienstag-Samstag). An den übrigen Wochentagen soll der Zug unter der gleichen Zugnummer erst ab Berlin verkehren (Abfahrten dort Sonntag und Montag). Da bei diesem Fahrplan montags zwei Zugläufe beginnen (1x in Berlin morgens, 1x in Hamburg abends) ist somit eine eindeutige Identifikation eines Zuges anhand des Verkehrstags am Abgangsbahnhof nicht möglich. Um dieses Problem zu lösen müßten die in Berlin beginnenden Züge dem Vortag ihrer tatsächlichen Abfahrt zugeordnet werden, also Samstag und Sonntag. Damit diese Züge ihre korrekten Abfahrts-Verkehrstage in Berlin behalten, müssen diese Fahrten mit einem departureDay von 1 beginnen.

Beispiel 2:

Ein Zug der regulär (täglich) spät abends verkehrt, wird, z.B. aufgrund einer Baumaßnahme, an einem Tag soweit später gelegt, dass er erst am Folgetag beginnt. Wenn sich der Primärschlüssel des Zuges aus Zugnummer und Verkehrstag am Abgangsbahnhof zusammensetzt, so würde er durch diese Später-Legung (ungewollt) seinen Primärschlüssel ändern. Wenn der Zug seinen Primärschlüssel auch nach der zeitlichen Verschiebung behalten soll, muss seine Abfahrt am Folgetag durch einen departureDay von 1 am Abgangsbahnhof definiert werden.

--
iRFP e. K. · Institut für Regional- und Fernverkehrsplanung
Hochschulstr. 45, 01069 Dresden
Tel. +49 351 4706819 · Fax. +49 351 4768190 · www.irfp.de
Registergericht: Amtsgericht Dresden, HRA 9347

Subject: Re: [railML3] Semantics of the attributes arrivalDay and departureDay
Posted by David Lichti on Mon, 18 Jan 2021 06:52:34 GMT

View Forum Message <> Reply to Message

On one hand, I think trains should be identified by explicit identifiers, not implicitly by their schedule.

But appart from identification, there is also the problem that many train line operations do not align with the 00:00-24:00 cycle. Actual operating times may rather resemble 05:00-01:00 or 04:00-02:00, where the latest trains of a given operating day actually run on the next calendar day. Rather than defining custom day cycles (which might differ between different lines in the same file, anyways), such trains could be modeled by having their first departure shifted by one day.

In example 2, it could also be necessary to have the train leave earlier at its first station, moving the departure from shortly after midnight to shortly before midnight. In this case, it would be desirable to even have negative integers as day offsets.

Best regards

David

Subject: Re: [railML3] Semantics of the attributes arrivalDay and departureDay
Posted by Roman Naumann on Tue, 09 Mar 2021 16:47:05 GMT

View Forum Message <> Reply to Message

David Lichti wrote on Mon, 18 January 2021 07:52But appart from identification, there is also the problem that many train line operations do not align with the 00:00-24:00 cycle. Actual operating times may rather resemble 05:00-01:00 or 04:00-02:00, where the latest trains of a given operating day actually run on the next calendar day. Rather than defining custom day cycles (which might differ between different lines in the same file, anyways), such trains could be modeled by having their first departure shifted by one day.

I fully agree with David's remark. In addition, I believe that being less restrictive with the departureDay/arrivalDay offsets does not make the status quo notably more complex, so I do not see anything speaking against the proposed change.

For example, a part of a split train as descripted in Figure 1 in Red here [1] currently (RailML 2) has two valid representations: a shifted operating period with a "negative jump" of departureDay between train parts or, alternatively, a positive departureDay for the beginning of the second train. Likewise, the first example ("route of the train is not completely mapped in the railML file") also explicitly allows for positive departureDays at the beginning of RailML 2 trains.

These corner cases in RailML 2, where we currently allow for positive departureDays at the beginning of trains, show, that any import implementation already has to implement mapping logic from shifted operating periods to calendaric validities with RailML 2.

Last, I would like to add that in example 2, if we consider named operating periods ("Mo-Fr") for timetable exchange, disallowing a positive departureDay at the beginning results in oddities or inconsistencies: We would have to export trains that are planned in the same operating period in the source system with two operating periods ("Mo-Fr" and "Tue-Sat") and shifted bitmasks - except if all trains shortly after midnight are part of a split train where the other part starts before midnight. The forced "splitting" of operating periods seems odd and the exception to it inconsistent (why should the operating period's naming of one train be affected by another, after all?).

This point was even made in the RailML 2 documentation, despite the restriction in place [2]: "Why not use shifted/rotated <operatingPeriod>s" -> "For several <operatingPeriod>s which are often used for passenger information it may be very difficult to find (understandable) text expressions if they would be shifted by one or more days."

So in summary, the "less restrictive" version described by Christian would in my opinion not introduce new complexity, but reduce inconsistencies and improve expressiveness.

[1] https://wiki2.railml.org/wiki/Dev:Examples_for_a_non-zero_operating_day_offset_at_the_first_ocpTT_of_a_train_run
[2] https://wiki2.railml.org/wiki/Dev:Midnight_overrun

Subject: Re: [railML3] Semantics of the attributes arrivalDay and departureDay
Posted by Milan Wölke on Sun, 25 Apr 2021 15:14:06 GMT
View Forum Message <> Reply to Message

Hi,

to keep everyone up to date..

As there are no opinions disagreeing the developer group for railML 3 has opted for allowing the less restrictive approach.

Best regards, Milan