
Subject: Facultative trains / Bedarfszüge

Posted by [Philip Wobst](#) on Fri, 30 Aug 2019 09:47:14 GMT

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

Hello all,

there is a need to export facultative trains with railML. A facultative train is planned the same way as any other train. However, as a facultative train it is not yet decided whether or not the train will actually run or not.

In the railML file there needs to be a way to distinguish the facultative train from one that is planned to actually run/operate.

The use cases for this might be:

- a train runs Mo-Fr on a regular basis and on Sa only on request
- a freight train is scheduled to run Mo-Fr on request and the train only actually runs/operates on the days that are eventually "needed" by the freight operator

Currently there is - as far as I know - no clear way to model facultative trains in railML. Merely the description of the @onRequest attribute of the <operatingDay> element suggests a potential solution for the first use case above.

If the use of @onRequest is the intended way to model facultative trains in railML, I see several drawbacks:

For the second use case with deviating operating days this will result in numerous <operatingDay> elements with start/end dates that need to be interpreted correctly.

The bitmask of the <operatingPeriod> is supposed to be the main attribute to define the actual running days of a train. The reading system would need to analyse all <operatingDay> elements to distinguish between operating and facultative days.

The identification of <train>s/<trainPart>s that are facultative is not easily possible - as is the case for cancellations.

I would like to discuss the common understanding and interpretation of facultative trains and find an agreed solution for railML 2.5.

Please provide some feedback in the forum regarding:

1. Do you have the need to model facultative trains in railML (yes/no)
2. Solutions for facultative trains in existing railML export/import scenarios
3. Ideas for a solution in railML 2.5

Thank you and best regards,
Philip Wobst

Hallo zusammen,

es besteht die Notwendigkeit, fakultative Züge mit railML zu exportieren. Ein fakultativer Zug wird wie jeder andere Zug geplant. Als fakultativer Zug ist jedoch noch nicht entschieden, ob der

Zug tatsächlich fährt oder nicht.

In der railML-Datei muss es eine Möglichkeit geben, den fakultativen Zug von einem zu unterscheiden, der tatsächlich fahren wird.

Die Anwendungsfälle dafür könnten sein:

ein Zug verkehrt Mo-Fr. regelmäßig und am Sa nur auf Anfrage.

ein Güterzug ist auf Anfrage für Mo-Fr vorgesehen und der Zug fährt nur an den Tagen, die der Frachtführer schließlich "benötigt".

Derzeit gibt es meines Wissens keine klare Möglichkeit, fakultative Züge in railML zu modellieren. Lediglich die Beschreibung des @onRequest-Attributs des Elements <operatingDay> schlägt eine mögliche Lösung für den ersten obigen Anwendungsfall vor.

Wenn die Verwendung von @onRequest der beabsichtigte Weg ist, fakultative Züge in railML zu modellieren, sehe ich einige Nachteile:

Für den zweiten Anwendungsfall mit abweichenden Betriebstagen ergeben sich daraus zahlreiche <operatingDay> Elemente mit Start-/Enddaten, die korrekt interpretiert werden müssen.

Die Bitmaske der <operatingPeriod> soll das Hauptattribut sein, um die tatsächlichen Fahrtage eines Zuges zu definieren. Das Importsystem müsste alle <operatingDay> Elemente analysieren, um zwischen Betriebs- und Fakultativtagen zu unterscheiden.

Die Identifizierung von <train>s/<trainPart>s, die fakultativ sind, ist nicht ohne weiteres möglich - z.B. wie bei Stornierungen.

Ich möchte das gemeinsame Verständnis und die Interpretation von fakultativen Zügen diskutieren und eine gemeinsame Lösung für railML 2.5 finden.

Bitte geben Sie im Forum ein Feedback zu:

1. Haben Sie die Notwendigkeit, fakultative Züge in railML zu modellieren (ja/nein)?
2. Gibt es bereits Lösungen für fakultative Züge in bestehenden railML-Export-/Importszenarien
3. Lösungsideen in railML 2.5

Danke und Gruß,
Philip Wobst

Posted by [Tobias Bregulla](#) on Tue, 03 Sep 2019 12:48:49 GMT

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

Hello / Hallo!

Am 30.08.2019 um 11:47 schrieb Philip Wobst:

- > In the railML file there needs to be a way to distinguish
- > the facultative train from one that is planned to actually
- > run/operate.

I would like to continue the discussion in German, so here is only a short version in English: We consider the model of the "facultative train" (runs on demand/request only) to be outdated in today's IT world. But if there is a need in the community, then gladly. Very important would be a very clear and sharp definition and demarcation.

- > In der railML-Datei muss es eine Möglichkeit geben, den
- > fakultativen Zug von einem zu unterscheiden, der
- > tatsächlich fahren wird.

Wir halten das Modell des Bedarfszuges in der heutigen Welt der (europäischen) Eisenbahnen mit getrennten Verantwortlichkeiten zwischen EVU/RU und EIU/IM und einer fast durchgängigen Verfügbarkeit von vernetzten Daten und Informationen für nicht mehr zeitgemäß. Für eine Datenhaltung historischer Fahrpläne und Anwendungsfälle sehe ich railML nicht so recht im Einsatz.

Aus der Diskussion bei uns kam der Hinweis auf die (vermutete) Historie des Bedarfszuges: Durch die Kennzeichnung als Bedarfszug konnten beteiligte Stellen des Bahnbetriebs ohne Anbindung an die Fahrplanwelt (z.B. Schrankenwärter Laumann; siehe <https://www.youtube.com/watch?v=CTpS2Oz5Q3o>) erfahren, dass diese Züge ohne weitere Information ausfallen können. In einer frühen Version der deutschen Fahrdienstvorschrift stand sogar, dass sich das Betriebspersonal nach verspäteten und evtl. ausgefallenen Zügen zu erkundigen hat (beim Bedarfszug dann wohl nicht). Ob das beim heutigen Bahnbetrieb noch eine Relevanz hat?

Bei der DB Netz wurden (wohl auch aus Gründen der Abrechnung und diskriminierungsfreien Trassenvergabe) die Bedarfszüge vor einigen Jahren abgeschafft. Bei welcher Bahn besteht denn der konkrete Bedarf für Bedarfszüge?

Wenn es in der railML-Community aber einen Bedarf für Bedarfszüge geben sollte, dann kann es gerne in railML 2.5 eingeführt werden. Interessant wäre da auch die Meinung der einsetzenden Bahnen wie SBB, BaneNOR, ÖBB und der schreibenden und lesenden Programme.

Extrem wichtig wäre aber eine sehr klare Definition der Anwendung, Beschreibung der semantischen Einschränkungen und Abgrenzung zur bestehenden Modellierung. Könnt ihr da bitte einen Vorschlag machen?

--

Best regards,

Tobias Bregulla
Bahnkonzept Dresden/Germany

Posted by [Milan Wölke](#) on Mon, 16 Sep 2019 15:07:45 GMT

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

Hallo zusammen, hi guys,

following tobias approach; first a short English summary: In general I think, although it may seem outdated in our modern interconnected railway world, however, apart from the use cases Philip is describing, I know of at least one other major railway company who very much relies on planning facultative trains and exchanging these between systems for their operation. Therefore I agree with the approach and would support the motion of adding this to railML with version 2.5.

Im Allgemeinen kann ich Tobias Einwand nachvollziehen, dass heutzutage die Planung von Fakultativtrassen und der Austausch der selben etwas veraltet wirkt. Allerdings kenne ich neben dem Anwendungsfall, über den Philip berichtet auch ein anderes großes EVU, bei dem die Planung und der Austausch von Fakultativtrassen eine entscheidende Rolle im betrieblichen Prozess spielen. Insofern würde ich mich dem Wunsch anschließen, eine entsprechende Modellierung in railML 2.5 TT zu ermöglichen. Bisher kenne ich neben der von Philip erwähnten Nutzung von @onRequest am <operatingDay> eine Abbildung, bei der solche Trassen als besondere Zugkategorie (/railML/timetable/categories/category) erfasst werden. Diese Lösung halte ich allerdings für ungünstig, da es erfordert, dass Produzent und Konsument sich über diese spezielle Zugkategorie außerhalb des Standards einigen müssten.

Soviel erstmal zu den ersten 2 Punkten aus Philips Post. Zum dritten Punkt muss ich mir auch erst nochmal Gedanken machen.

Best regards, Milan

Subject: Re: Facultative trains / Bedarfszüge

Posted by [Philip Wobst](#) on Wed, 18 Sep 2019 20:47:19 GMT

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

Hello all,

I will continue with only a German version. We have several customers across Europe and also elsewhere, that use facultative trains for planning trains. This ranges from the yearly timetable to very short term planning for extra trains.

Let me describe one process for a customer that uses railML already:

1. In the yearly timetable the planner has to consider facultative trains in the planning process and these actually occupy capacity and create conflicts with other trains that need to be resolved before the timetable can be published. E.g. a facultative train has to be planned with all crossings with other trains just as if it is running.
2. The yearly timetable is published and moves into short term planning (STP). In this world the facultative train is visible in all train graphs and overviews but it does not occupy capacity or create conflicts any longer. The STP planner needs to consider the facultative train for all

replanning or extra trains according to defined rules.

3. If on the day of operation the path for the facultativ train is still "free" the dispatcher an "activate" the train and let it run as planned. In order to do this the dispatch system needs to "receive" the facultative train though in the first place.

Other customers have facultative trains

a. for on request freight with fixed times

b. for pre-planned train "routes" (stations, runtimes, stops) but no fixed start time. On the day of operation such a train can be activated and given a start time without having to plan the actual train run up to the destination

c. but I do not know their use case

We would have two customers who would use facultative trains in a railML 2.5 export.

Thank you for your feedback and best regards,

Philip

Hallo zusammen,

Ich werde mit nur einer deutschen Version fortfahren. Wir haben mehrere Kunden in ganz Europa und auch anderswo, die fakultative Züge für die Zugplanung nutzen. Dies reicht vom Jahresfahrplan bis hin zur sehr kurzfristigen Planung von Sonderzügen.

Lassen Sie mich einen Prozess für einen Kunden beschreiben, der railML bereits einsetzt:

1. Im Jahresfahrplan muss der Planer fakultative Züge im Planungsprozess berücksichtigen, die tatsächlich Kapazität belegen und Konflikte mit anderen Zügen erzeugen, die vor der Veröffentlichung des Fahrplans gelöst werden müssen. So muss z.B. ein fakultativer Zug mit allen Kreuzungen mit anderen Zügen so geplant werden, als ob er fährt.

2. Der Jahresplan wird veröffentlicht und geht in die Kurzfristplanung (STP) über. In dieser Welt ist der fakultative Zug in allen Zuggrafiken und Übersichten sichtbar, belegt aber keine Kapazität mehr und erzeugt keine Konflikte mehr. Der STP-Planer muss den fakultativen Zug für alle Um- oder Zusatzzüge nach definierten Regeln berücksichtigen.

3. Wenn am Tag der Inbetriebnahme die Trasse für den fakultativen Zug noch "frei" ist, aktiviert der Dispatcher den Zug und lässt ihn wie geplant fahren. Dazu muss das Dispositionssystem den fakultativen Zug zunächst "importieren".

Drei andere Kunden haben auch fakultative Züge

a. für auf Bedarfsgüterzüge mit festen Zeiten

b. für vorgeplante Zug "Strecken" (Betriebstellen, Laufzeiten, Halte), aber keine feste Startzeit.

Am Tag des Betriebs kann ein solcher Zug aktiviert und mit einer Startzeit versehen werden, ohne dass die tatsächliche Zugfahrt bis zum Ziel geplant werden muss.

c. ein Kunde hat Züge mit fakultativen Verkehrstagen aber ich kenne ihren Anwendungsfall nicht.

Wir hätten zwei Kunden, die fakultative Züge in einem railML 2.5-Export verwenden würden.

Vielen Dank für dein Feedback und beste Grüße,

Posted by _____ on Wed, 13 Nov 2019 08:15:43 GMT

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

Hello,

I agree that it is currently not practicable to represent facultative trains. From my point of view, two solutions are possible:

1) The differentiation between regular and facultative operating days takes place within the <operatingPeriod>. Since a combination of regular and facultative operating days can occur (see Philips example "Mo-Fr runs regularly and on Sa on demand"), all elements of the <operatingPeriod> would have to be duplicated. In addition, there is a rule that more than one <operatingDay> element may occur in one <operatingPeriod>, but only if "startDate" / "endDate" ensures that they do not overlap. This rule would also have to be abandoned.

2) Separate <operatingPeriods> and correspondingly separate <trainPart>s are used for regular and facultative operating days. In this case the <operatingPeriod> would have to get a new attribute "onRequest", the attribute of the same name on <operatingDay> would be dropped. Disadvantage would be a certain redundancy, since additional <trainPart>s have to be created, which will often be identical except for the <operatingPeriodRef>. But the change effort is clearly smaller, likewise the comprehensibility of the element <operatingPeriod> becomes clearly better from my point of view.

Due to the described advantages and disadvantages we would plead for variant 2.

Best regards
Christian Rößiger

German version:

Hallo,

ich stimme zu, dass die Abbildung von Bedarfszügen derzeit nicht praktikabel möglich ist. Aus meiner Sicht wären zwei Lösungen denkbar:

1) Die Differenzierung nach regulären und Bedarfs-Verkehrstagen erfolgt innerhalb der <operatingPeriod>. Da eine Mischung von regulären und

Bedarfs-Verkehrstagen vorkommen kann (siehe Philips Beispiel "verkehrt Mo-Fr regelmäßig und an Sa nach Bedarf"), müßten alle Elemente der <operatingPeriod> dupliziert werden. Außerdem besteht bisher die Regel, dass mehrere <operatingDay> - Elemente in einer <operatingPeriod> auftreten dürfen, aber nur, wenn via "startDate" / "endDate" sicher gestellt ist, dass sie sich nicht überlappen. Diese Regel müßte ebenfalls aufgegeben werden.

2) Es werden für reguläre und Bedarfs-Verkehrstage jeweils eigene <operatingPeriods> und dementsprechend auch separate <trainPart>s verwendet. In diesem Fall müßte die <operatingPeriod> ein neues Attribut "onRequest" erhalten, das gleichnamige Attribut am <operatingDay> würde entfallen. Nachteil wäre eine gewisse Redundanz, da zusätzliche Zugteile angelegt werden müssen, die außer der <operatingPeriodRef> oft identisch sein werden. Dafür ist der Änderungsaufwand deutlich geringer, ebenso wird aus meiner Sicht die Verständlichkeit des Elements <operatingPeriod> deutlich besser.

Aufgrund der beschriebenen Vor- und Nachteile würden wir daher für Variante 2 plädieren.

Viele Grüße
Christian Rößiger

--

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

Posted by [Milan Wölke](#) on Tue, 26 Nov 2019 16:47:54 GMT
[View Forum Message](#) <> [Reply to Message](#)

I would agree with the 2nd approach of Christian. I think its a more clear approach.

Ich würde Christian zustimmen und mich für den 2. Ansatz aussprechen. Aus meiner Sicht ist der Ansatz erheblich klarer.

Posted by [Milan Wölke](#) on Thu, 28 Nov 2019 13:48:50 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

in der heutigen Entwickler Telco wurde dieses Thema diskutiert. Aus der Diskussion hat sich ergeben, dass eine Abbildung nach Vorschlag 2 von Christian Rößiger auf breite Zustimmung trifft, mit einer Änderung. Das neue Flag für onRequest soll nunmehr nicht mehr an der operatingPeriod definiert werden, sondern am trainPart. Hintergrund ist, dass Operating Periods auch in anderen Kontexten verwendet wird, in denen ein onRequest Flag keinen Sinn macht und daher über Semantic Constraints ausgeschlossen werden müsste. Um das zu vermeiden und eine klare Modellierung zu haben, wird nun der trainPart als "Bei Bedarf" gekennzeichnet.

Ausserdem wurde sich in der heutigen Telco dafür ausgesprochen, das Flag onRequest am operatingDay ab der Version 2.5 für deprecated zu erklären. Dazu ist im Forum ein separater Post angelegt worden um eine Diskussion zu ermöglichen.

in today's developer Telco this topic was discussed. The discussion it resulted in a broad approval of a modelling based on proposal 2 of Christian Rößiger, with one change. The new flag for onRequest should no longer be defined at the operatingPeriod, but at the trainPart. The background is that operating periods are also used in other contexts in which an onRequest flag makes no sense and would therefore have to be excluded via semantic constraints. To avoid this and to have a clear modeling, the trainPart is now marked as "On Request".

Furthermore, in today's telco, it was decided to declare the flag onRequest on operatingDay deprecated from version 2.5 onwards. For this purpose a separate post has been created in the forum to allow a discussion.

Best regards, Milan

Posted by [Philip Wobst](#) on Mon, 02 Dec 2019 07:40:49 GMT

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

Milan Wölke wrote on Thu, 28 November 2019 14:48

... Aus der Diskussion hat sich ergeben, dass eine Abbildung nach Vorschlag 2 von Christian Rößiger auf breite Zustimmung trifft, mit einer Änderung. Das neue Flag für onRequest soll nunmehr nicht mehr an der operatingPeriod definiert werden, sondern am trainPart.

Die von Milan beschriebene Umsetzung deckt unsere Anforderungen für den Anwendungsfall ab. Dank an alle, die zur Diskussion beigetragen haben.

Gruß aus Hannover,

Philip Wobst

Posted by [Milan Wölke](#) on Mon, 02 Dec 2019 16:55:56 GMT
[View Forum Message](#) <> [Reply to Message](#)

Trac ticket for this issue has been created in order to incorporate the necessary change into version 2.5.

<https://trac.railml.org/ticket/372#ticket>
