

Hallo,

meine Name ist Patrik Thoma von der Rhätischen Bahn (RhB) in der Schweiz und bin zuständig für Betrieb und Weiterentwicklung der Applikationen im Bahnbetrieb bei uns.

Bei der Realisierung unseres Projektes «Flügelzugbetrieb» vernetzten wir unsere Applikationen für Planung und Disposition im Bereich Fahrplan und Formationen neu.

Für den Austausch der Fahrplandaten ist das Format railML 2.5 vorgesehen bzw. bereits in Umsetzung. Für die Formationen/Umläufe war bis anhin geplant, ein spezifisches Format zu verwenden.

Bei der Umsetzung sind nun aber diverse Probleme aufgetaucht, vor allem das Referenzieren der Formationen (aus Formations-/Umlaufexport) zu den Zügen (Fahrplanexport) bereitet mit all den zeitlichen Abweichungen grosse Schwierigkeiten.

Aus diesem Grund prüfen wir die Integration der Formations- und Umlaufdaten in den railML-Fahrplan und wollen hier die zur Verfügung stehenden Elemente/Attribute nutzen. Bei einem Thema stehen wir allerdings vor einer Herausforderung: unsere Züge bestehen sehr oft aus einer Lok und Einzelwagen und die Formation verändert sich sehr oft auf einem Zuglauf. Mit dem gem. railML-Wiki definierten semantic constrains für das Element Blockpart (siehe Auszug unten) würde dies bedeuten, dass wir für jede veränderte Formation einen separaten trainPart definieren/generieren müssten, was erhebliche Auswirkungen auf Komplexität und Datenmenge hat.

Aus diesem Grund wollen wir anfragen, ob dieses Semantic Constraints so angepasst werden kann, dass ein trainPart von mehreren blockParts (mit verschiedenen Formationen) referenziert werden kann, immer unter Berücksichtigung der Anforderung, dass der gesamte trainPart von blockParts abgedeckt wird.

Die Idee wäre also den semantic constraint TT:003 so anzupassen, dass bei Referenzierung eines trainPart durch einen blockPart durchaus startOcpRef und endOcpRef abweichend zum referenzierten trainPart angegeben werden können, unter der Voraussetzung, dass erstens startOcpRef und endOcpRef auf Ocps verweisen, die im Zuglauf des trainParts enthalten sind, und zweitens andere blockParts auf den verbleibenden Teil des trainPart ohne Überlagerung referenzieren. Dies würde für unsere Betriebsfälle, also sehr häufige Formationswechsel auf der Strecke, die sich auch zeitlich nicht systematisieren lassen (also unregelmäßig auftreten) dazu führen, dass sich die Redundanz bezüglich der trainParts im railML Fahrplan massiv reduzieren würde. Wir könnten uns vorstellen, dass dies auch anderen Bahnen mit ähnlichen Bedürfnissen (abweichende Formationen) sehr weiterhelfen würde.

Für eine Prüfung danken wir herzlich!

Patrik Thoma

siehe Semantic Constraint "TT:003":

<https://wiki2.railml.org/wiki/TT:blockPart>

---

---

Subject: Re: TT:blockPart: Semantic Constraint  
Posted by \_\_\_\_\_ on Fri, 28 Jan 2022 14:10:45 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hallo Patrik,

willkommen im railML-<Timetable>-Forum!

Als traditionell umlaufaffiner Mitwirkender fühle ich mich mit dieser Frage einmal angesprochen.

Zunächst: Das Anliegen und die Beschreibung des Bedarfs sind im Wesentlichen nachvollziehbar, danke für die gute Beschreibung.

- > Dies würde für unsere Betriebsfälle, also
- > sehr häufige Formationswechsel auf der Strecke, die sich
- > auch zeitlich nicht systematisieren lassen (also
- > unregelmäßig auftreten)...

Gewünscht ist ein Beschränken des Laufwegs (startOcpRef und endOcpRef) auf einen Teil des Laufwegs des referenzierten <trainPart>s. Ich verstehe nicht, wie in dem Zusammenhang "zeitlich nicht systematisieren lassen" gemeint ist. Der referenzierte <trainPart> hätte an den jeweiligen <ocpTT>s immer planmäßige Ankunfts- und Abfahrtszeiten. Diese Zeiten wäre i. S. des Fahrplans einschl. Umlaufplans ausreichend Systematisierung - mehr ist gar nicht notwendig.

Um Missverständnisse zu vermeiden, daher nochmal die Rückfrage:

a) Wird hier ein Fall beschrieben, in dem der Umlaufplan eine detailliertere Abbildung umfasst als der allgemeine Fahrplan (<trainPart>s und dessen übergeordnete Elemente)? Das würde bedeuten, dass in der Formation des referenzierten <trainPart>s Fahrzeuge enthalten sind, die im Zug inzwischen gar nicht mehr oder noch nicht mitfahren.

b) Sind sich Umlaufplan und allgemeiner Fahrplan hinsichtlich Abbildungsgenauigkeit einig? Dann sollte das ganze ohne Mehraufwand an <trainPart>s abbildbar sein.

Wenn es sich um (b) handelt - wovon ich ausgehe -, muss tatsächlich jedem Zwischenbahnhof (<ocpTT>), an dem die Umlaufpläne ein An- oder Abhängen von Wagen vorsehen, ein neuer <trainPart> gebildet werden. Das ist logische Konsequenz der gleichen Abbildungsgenauigkeit:

Wo unter <rostering> was an- oder abgehängt wird, muss auch unter <trainParts> was an- oder abgehängt werden.

ABER: Es muss nicht für jedes einzelne an- oder abzuhängende Fahrzeug ein neuer <trainPart> gebildet werden! Daher ist es keine Redundanz und m. E. kein signifikanter Mehraufwand gegenüber lang durchgehenden <trainPart>s. Es wird einfach nur ein langer <trainPart> in mehrere kurze, sequentiell verkettete <trainPart>s aufgeteilt - ohne Überlappung, ohne Redundanz.

- > Aus diesem Grund wollen wir anfragen, ob dieses Semantic
- > Constraints so angepasst werden kann, dass ein trainPart von
- > mehreren blockParts (mit verschiedenen Formationen)
- > referenziert werden kann, immer unter Berücksichtigung der
- > Anforderung, dass der gesamte trainPart von blockParts
- > abgedeckt wird.

Dieser Formulierung würde ich voll zustimmen: Ein <trainPart> kann von mehreren <blockPart>s aus verschiedenen Umlaufplänen (<rostering>s) referenziert werden kann, unter den rein semantischen Bedingungen

- dass das oder die Fahrzeuge des Umlaufplans auch in der Formation des <trainPart>s enthalten sind,
- dass möglichst(?) jedes Fahrzeug der Formation des <trainPart>s durch genau einen Umlaufplan erfasst ist.

Wenn also ein <trainPart> folgende Formation hat: "Lok + 2x WagenA + 3x WagenB", dann darf es gern

- einen Umlaufplan für die Lok, einen anderen für "2x WagenA" und einen dritten für "3x WagenB" geben

oder

- einen Umlaufplan für die Lok und einen anderen für "2x WagenA + 3x WagenB" geben

oder

- je einen (insgesamt 6) Umlaufplan für jedes einzelne Fahrzeug im Zug

oder

viele andere mögliche Kombinationen.

Im nächsten Abschnitt (hier: <trainPart>) des Zuges kann die Formation anders aussehen (etwa nur noch "Lok + 2x WagenA + 2x WagenB" = 1x WagenB wurde abgekuppelt) und demzufolge auch die Anzahl Umlaufpläne anders sein, die diesen <trainPart> erfassen.

Hilft das zur Einschätzung der Situation aus Sicht railML weiter?

Falls ich etwas missverstanden oder mit meiner Interpretation der Anforderungen falsch liegen sollte, bitte nochmal melden.

Viele Grüße,  
Dirk Bräuer.

Subject: Re: TT:blockPart: Semantic Constraint  
Posted by [Milan Wölke](#) on Thu, 03 Feb 2022 13:23:54 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hallo zusammen,

zunächst mal, wie auch schon von Dirk, willkommen im railML-<timetable> Forum. Das Ziel von railML ist es Bahnen bei Austausch von bahnbezogenen Daten zu unterstützen, insofern prüfen wir gerne, ob der Semantic Constraint an der Stelle gelockert oder verändert werden könnte um einen reibungslosen Datenaustausch zu gewährleisten.

Um das vorliegende Problem besser verstehen und diskutieren zu können, würde ich anregen wollen ein kleines Beispiel anzubringen, damit man sich das besser vorstellen kann.

Mit freundlichen Grüßen, Milan

---

---

Subject: Re: TT:blockPart: Semantic Constraint  
Posted by [Wolfgang Nixdorf](#) on Mon, 07 Feb 2022 03:24:06 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hoi railML-Team,

mein Name ist Wolfgang Nixdorf von der Trapeze Group.  
Wir unterstützen die RhB bei der Umsetzung des Projektes 'Flügelzugbetrieb'.  
Dem Wunsch eines Beispiels möchte ich gerne nachkommen.  
Anbei ein Auszug mit einem Trainpart und zwei Blockparts, die jeweils auf eine Teilreferenz des Ortsverlaufs verweisen:

```
<trainPart categoryRef="cat-2" debitcode="9233" id="tp-64_1" line="RhB17" name="17_1 Takt  
LQ-DIS" timetablePeriodRef="ttp-1310" trainNumber="1725">
```

```
...  
<blockPart id="bp-1725-24496-2-1" begin="08:17:00" end="09:30:00" startOcpRef="ocp-149"  
endOcpRef="ocp-145" mission="timetable" trainPartRef="tp-64_1" runLength="1.000"  
formationRef="f-4882069-1" operatingPeriodRef="op-2"/>  
<blockPart id="bp-1725-24496-2-2" begin="09:33:00" end="10:11:00" startOcpRef="ocp-145"  
endOcpRef="ocp-138" mission="timetable" trainPartRef="tp-64_1" runLength="1.000"  
formationRef="f-4882069" operatingPeriodRef="op-2"/>
```

Mit hanseatischen Grüßen,  
Wolfgang Nixdorf

---

---

Subject: Re: TT:blockPart: Semantic Constraint

Posted by \_\_\_\_\_ on Mon, 07 Feb 2022 11:46:21 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hallo Herr Nixdorf,

danke für das Beispiel.

Wir können daraus allerdings nicht erkennen, worin der Unterschied zwischen Formation f-4882069-1 und Formation f-4882069 liegt. Wenn es zwischen beiden Formationen einen semantischen Unterschied gibt (unterschiedliche Fahrzeuge, nicht "nur" unterschiedliche Reihung), dann könnte man aus dem Umlauf nicht mehr verfolgen, welches Fahrzeug von wo nach wo fährt.

Auch hätte man dann den von mir schon im Fall (a) angesprochenen Widerspruch zwischen <trainPart> und <rostering>: Der <trainPart> - der von ocp-149 über ocp-145 bis ocp-138 fährt - kann auf diesem gesamten Weg nur entweder auf Formation f-4882069-1 oder auf Formation f-4882069 verweisen, aber nicht auf beide. Wenn beide also inhaltlich nicht äquivalent sind, würde der <trainPart> dem Umlaufplan widersprechen. Das wäre dann vermutlich kein Fall, den wir in railML standardisieren wollten.

Etwas weniger bedenklich wäre das Beispiel m. E. nur dann, wenn sich beide Formationen nur in der Reihung (Reihenfolge der Fahrzeuge) unterschieden, aber die gleichen Fahrzeuge beinhalteten, etwa wegen Kopfmachens des Zuges im Laufweg. Allerdings gäbe es auch dann meist eine alternative Abbildungsmöglichkeit (s. a. [1]).

Viele Grüße,  
Dirk Bräuer.

[1] [https://wiki2.railml.org/wiki/Dev:Reversing\\_trains\\_and\\_formations](https://wiki2.railml.org/wiki/Dev:Reversing_trains_and_formations)

---

---

Subject: [railML2] Re: TT:blockPart: Semantic Constraint  
Posted by [Milan Wölke](#) on Tue, 08 Feb 2022 14:29:39 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hallo zusammen,

Aus meiner Sicht muss vor allem sichergestellt werden, dass die Informationen am trainPart und die Informationen an dem/den blockPart(s) sich nicht widersprechen, da stimme ich dem Post von Dirk von iRFP absolut zu. Entsprechend müsste man entweder den trainPart teilen oder am trainPart keine Formationsinformationen referenzieren.

Insofern könnte ich mir vorstellen, den Semantic Constraint TT:003 unter dieser Vorbedingung so anzupassen:

redundant to referenced <trainPart>, must be empty or without contradiction to referenced <trainPart>. if the <blockPart> references only part of the <trainPart> all other parts of the <trainPart> need to be referenced by other <blockPart>s as well.

Ausserdem würde ich vorschlagen, zusätzlich einen neuen Semantic Constraint zu verfassen:

There shall not be a contradiction between the specified formations between  
<rostering>/<blockPart> and <trainPart>.

Best regards, Milan

---

---

Subject: Re: TT:blockPart: Semantic Constraint  
Posted by \_\_\_\_\_ on Wed, 09 Feb 2022 08:25:25 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hallo Milan und Mitlesende,

- > Insofern könnte ich mir vorstellen, den Semantic Constraint
- > TT:003 unter dieser Vorbedingung so anzupassen:
  
- > ...if the
- > <blockPart> references only part of the <trainPart> all
- > other parts of the <trainPart> need to be referenced by
- > other <blockPart>s as well.

Zwei Anmerkungen zu diesem Vorschlag:

- Ich bin dagegen. Meiner Meinung nach sollte es weiterhin zulässig bleiben, nur einen Teil der Fahrzeuge eines <trainPart>s mit einem Umlauf abzudecken, etwa weil nicht für alle Fahrzeuge überhaupt ein Umlaufplan gebildet wird (bei Güterzügen etwa nur für die Lok, nicht für die Güterwagen). In diesem Sinne ist "nur eine Teilmenge" kein Widerspruch zwischen <trainPart> und Umlauf. Wichtig im Sinne von "without contradiction" wäre, dass der Umlauf nicht behaupten würde, das Fahrzeug führe A-B, wenn der <trainPart> behauptet, es führe A-C oder umgekehrt.

- Wenn Du den zusätzlichen Satz trotzdem aufnimmst, dann bitte darin klarstellen, dass sich das auf Fahrzeuge der Formation des <trainPart>s bezieht. Die Formulierung "parts of the <trainPart>" könnte im Allgemeinen mehr als Fahrzeuge umfassen, etwa <ocpTT>s, was ja aber nun gerade wieder zu Widersprüchen führen würde.

Bitte bedenken, dass die Anfrage der RhB sich auf ein Lockern der Bedingungen bezog. Wäre es nicht über das Ziel hinausgeschossen, wenn Du sie jetzt daraufhin verschärfen würdest? Ich denke, wir sollten erst einmal die Rückmeldung von RhB und Trapeze Group zur eigentlichen Anfrage abwarten.

Viele Grüße,  
Dirk.

---

---

Subject: Re: TT:blockPart: Semantic Constraint

Posted by [Milan Wölke](#) on Wed, 09 Feb 2022 14:31:28 GMT

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

---

Hallo Dirk, hallo Patrik und Wolfgang,

da sehen wir mal wieder wie missverständlich Sprache sein kann. Was ich ausdrücken wollte, ist, dass wenn man schon im <trainPart> @startOcp oder @endOcp abweichend vom ersten bzw. letzten <ocpTT> des <trainPart> formuliert, dass dann doch bitte der Rest der <ocpTT> des <trainPart> von einem anderen <blockPart> abgedeckt sind, damit die Beschreibung nicht einfach mitten im <trainPart> abbricht.

Soweit ich das verstehe, wollen die Kollegen von RhB und Trapeze Triebzugumläufe abbilden aber trotzdem die volle Formation beschreiben, auch wenn diese sich an einem Ort eines <trainPart> ändert. Der <trainPart> beschreibt also den Zuglauf des Triebzugs (und soll nicht gebrochen werden, wenn sich etwas an den Wagen ändert). @Trapeze, @RhB: Ist meine Interpretation korrekt?

Best regards, Milan

---

---

Subject: Re: TT:blockPart: Semantic Constraint

Posted by [Wolfgang Nixdorf](#) on Wed, 23 Feb 2022 14:23:12 GMT

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

---

Hoi railML-Team,

lieber spät als nie ...

Noch eine Ergänzung zu den referenzierten Formationen.

Grundsätzlich werden bei einer 'Teilung' des Zuges innerhalb eines blockParts immer alle Abschnitte des trainParts geliefert.

Der trainPart wird somit komplett referenziert. Die im Beispiel genannten Formationen unterscheiden sich in der Anzahl der Wagen.

Bei einer Änderung des Triebfahrzeugs und damit verbundenen Umlaufänderung würde der trainPart dann von unterschiedlichen blockParts und Formationen referenziert werden.

---

---

Subject: Re: TT:blockPart: Semantic Constraint

Posted by \_\_\_\_\_ on Fri, 25 Feb 2022 10:16:41 GMT

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

---

Hallo Herr Nixdorf,

danke für die Rückmeldung. Allerdings haben wir dann Fall (a) mit dem Widerspruch zwischen <trainPart> und <rostering>: Der <trainPart> kann auf nur entweder die auf Formation mit mehr oder auf die Formation mit weniger Wagen verweisen, aber nicht auf beide. Auf irgendeinem Abschnitt des <trainPart>s weist dieser dann mehr oder weniger Wagen auf als der Umlaufplan. Dieser Widerspruch ist vermutlich kein Fall, den wir in railML standardisieren

---

wöllten.

Um das zu vereinheitlichen, wäre ja nur eine Teilung des ursprünglichen <trainPart>s in zwei <trainPart>s an der Stelle notwendig, an der der Umlaufplan den Wagenwechsel ausweist. Eine Redundanz entsteht dadurch nicht, da sich beide "<trainPart>-Hälften" nicht überlappen. Im Sinne einheitlicher Datenabbildung und Standardisierung sollte meiner Meinung nach dieser Weg gewählt werden.

Mit freundlichen Grüßen,  
Dirk Bräuer.

---

---

Subject: Re: TT:blockPart: Semantic Constraint  
Posted by [Vasco Paul Kolmorgen](#) on Tue, 08 Mar 2022 16:22:26 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hallo zusammen,

besten Dank für Eure konstruktiven Beiträge und die Fachdiskussion bislang. Es ist toll, wenn wir hier gemeinsam railML voranbringen.

Als allgemeiner Koordinator sehe ich hier vor allem einen Aspekt: Wir haben einen Produzenten und einen Konsumenten, die beide railML 2.5 verwenden wollen und werden. Insofern sollten wir hier konstruktiv eine Lösung suchen, wie beiden Kollegen geholfen werden kann. Wäre es denn nicht gangbar, die Widersprüche über eine kundenspezifische Erweiterung aufzulösen und somit trotzdem den allgemeingültigen Charakter des railML-Files zu erhalten?

Was denkt ihr darüber?

Freundliche Grüße,

--

Vasco Paul Kolmorgen - Governance Coordinator  
railML.org (Registry of Associations: VR 5750)  
Phone railML.org: +49 351 47582911  
Altplauen 19h; 01187 Dresden; Germany

---