
Subject: separation of <ocpTT> from <trainPart> - meeting 11.11.

Posted by on Tue, 19 Nov 2013 12:07:05 GMT

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

- English summary below -

Hallo allerseits,

Beim Treffen am 11.11. wurde ebenfalls die Frage diskutiert, inwieweit in Zukunft in RailML die Betriebsstellen- und Zeitenabfolge vom <trainPart> getrennt werden sollten. Hintergrund ist insbesondere die erhebliche Redundanz, die sich wiederholende Zeitenabfolge *_gekuppelter_* <trainParts> derzeit bedeutet.

Konkret würde <ocpsTT> mit allen Unter-Elementen vom <trainPart> in eine neue Struktur umziehen. Als Arbeitstitel für die neue Struktur wurde <itinerary> vorgeschlagen.

Beim <trainPart> verblieben die elementaren Alleinstellungsmerkmale wie <formationTT> mit /load/, /length/, <passengerUsage> usw. und <operatingPeriodRef>.

<itinerary> bzw. <itineraries> würde eine neue Struktur außerhalb der Hierarchie <train> <trainPartSequence> <trainPart> werden, d. h. ein Unter-Element von <timetable>. Auf ein <itinerary>-Element würde von einem <trainPartSequence> aus verwiesen (/itineraryRef/). Der Verweis soll außerdem einen optionalen Zeitversatz beinhalten können (<itineraryRef Ref=... timeShift=... />).

Dieser Vorschlag (von Andreas Tanner) fand allgemeine Zustimmung.

Es sei ausdrücklich darauf hingewiesen, dass durch diesen Ansatz zunächst „nur“ die Redundanz in parallel verketteten (gekuppelt verkehrenden) <trainParts> reduziert wird. In keiner Weise wird hierdurch die sequentielle Verkettung (Laufwegabschnitte – <trainPartSequences>) vermieden. Es wurde ebenfalls der Vorschlag geäußert, den gesamten Laufweg eines Zuges über alle bisherigen <trainPartSequences> hinweg in einer geschlossenen Betriebsstellen- und Zeitenabfolge abzubilden. Hierzu wurden keine Lösungsansätze diskutiert.

Ebenfalls wurde die Frage diskutiert, ob und wie „Mustertrassen“ ohne konkrete Zeitenabfolge abgebildet werden können. Es bestand Einigkeit, dass dies ohne Änderungen in RailML möglich ist. Es wären die Attribute /arrival/ und /departure/ zu vermeiden, sehr wohl aber die Elemente <ocpTT> (mit /ocpRef/) und <sectionTT> <runTimes> (z. B. mit /minimalTime/) anzugeben.

Beispiel:

```
<ocpTT ocpRef='ocp_DRAG' ocpType='pass'>
  <sectionTT section='DRAG-DAF' >
    <runTimes minimalTime='PT145S' operationalReserve='PT9S' />
  </sectionTT>
</ocpTT>

<ocpTT ocpRef='ocp_DAF' ocpType='stop'>
  <sectionTT section='DAF-DBZ' >
    <runTimes minimalTime='PT422S' operationalReserve='PT37S' />
  </sectionTT>
  <stopDescription>
    <stopTimes minimalTime='PT2M0S' />
  </stopDescription>
</ocpTT>
```

English summary

Dear all,

there was a discussion on a possible separation of the route and times from a <trainPart> into a new structure, possibly <itinerary>. This is mainly to avoid the current redundancy when two or more <trainParts> run coupled in one <trainPartSection>.

All <ocpTT> and its sub-elements would move from <trainPart> to new <itinerary>.

The new structure <itineraries> would be a outside the current hierarchy <train> <trainPartSection> <trainPart>, so it would be a sub-structure of <timetable>.

The cross-link to a <itinerary> would come from <trainPartSection>.

The suggestion (by Andreas Tanner) was mainly appreciated.

If you prefer a full English conversation on this topic, please do not hesitate to tell us.

Dirk.

Subject: addendum: separation of <ocpTT> from <trainPart> - meeting 11.11.

Posted by on Tue, 19 Nov 2013 18:45:55 GMT

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

Hallo allerseits,

ergänzend zu den Notizen vom 11.11. würde ich gern ein paar Gedanken aus meiner Sicht zu diesem Thema beisteuern:

Für das „Gruppieren von Zügen“ – also z. B. für ein neues Element <trainGroup> – wurden m. E. folgende Instanzen des (bisherigen) Elements <train> in Betracht gezogen:

1a) Instanzen des (bisherigen) Elements <train> mit identischem Attribut /trainNumber/, die sich hinsichtlich der Verkehrstage unterscheiden: Sie haben disjunkte Verkehrstage. Man spricht von unterschiedlichen „Hauptläufen“ eines Zuges. Sie wären bisher nur an dem Attribut /scope/ = „primary“ mit unterschiedlicher /additionalTrainNumber/ erkennbar.

1b) Instanzen des (bisherigen) Elements <train> mit identischem Attribut /trainNumber/, die unter den Oberbegriff „Nebenläufe“ oder „Ergänzungsfahrpläne“ (DB-Terminologie) fallen.. Sie wären bisher nur an dem Attribut /scope/ = „secondary...“ erkennbar, ggf. mit unterschiedlicher /additionalTrainNumber/.

2) Instanzen des (bisherigen) Elements <train> mit identischem Attribut /trainNumber/, die unterschiedliche Planungszustände repräsentieren und sich daher gegenseitig überschreiben (neuere überschreiben ältere). Unter diese Kategorie fallen ebenfalls Ausfälle und Wieder-Einlegungen von Zügen. Im Allgemeinen kann das als „Zug-Historie“ bezeichnet werden. Ebenso fallen hierunter solche Planungszustände wie „Rahmenvertragsperiode“, „Jahresfahrplan“, „Gelegenheitsverkehr“. Für diese Kategorie gibt es m. E. bisher keine vorgesehene Abbildungsmöglichkeiten in RailML. Es ist aber bereits mehrfach danach gefragt worden, zuletzt

- durch Christian Wermelinger am 26.02.2013,
- durch Andreas Tanner am 26.06.2013,
- durch Philip Wobst am 19.11.2013.

All diesen Fällen ist gemein: Es geht immer um Instanzen des (bisherigen) Elements <train> mit identischem Attribut /trainNumber/.

Aus meiner Sicht handelt es sich dabei um zwei unterschiedliche (Teil-)Problematiken: 1 „Mehrfachzugläufe“ und 2 „Zug-Historie“. Sie sollten (oder zumindest können) m. E. mit unterschiedlichen Ansätzen gelöst / umgesetzt werden. Wie bereits im vorherigen Beitrag (Notizen vom 11.11.) geschrieben, betrifft 2 „Zug-Historie“ unter Umständen auch weitere RailML-Elemente (Zugarten, Infrastruktur, Fahrzeuge), so dass hierfür u. U. ein allgemeingültigerer Ansatz in Frage kommt.

Es wäre sicherlich hilfreich, wenn wir zunächst klären könnten, ob hinsichtlich der Trennung und der Berührungspunkte dieser beiden Teilkomplexe Einigkeit besteht. Ich möchte vorschlagen, dass wir die weitere Diskussion im Bewusstsein der Trennung beider Teilkomplexe führen.

Viele Grüße,
Dirk.

Subject: Re: addendum: separation of <ocpsTT> from <trainPart> - meeting 11.11.
Posted by [Joachim Rubröder railML](#) on Sun, 09 Mar 2014 22:59:56 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Dirk,

Dirk Brähler wrote:

> Konkret würde <ocpsTT> mit allen Unter-Elementen vom <trainPart> in
> eine neue Struktur umziehen. Als Arbeitstitel für die neue Struktur
> wurde <itinerary> vorgeschlagen.
>

> Hierarchie <train> <trainPartSequence> <trainPart> werden, d. h. ein
> Unter-Element von <timetable>.

Ich habe diesen Vorschlag in das Trac Ticket #240 überführt.
<https://trac.railml.org/ticket/240>

> Auf ein <itinerary>-Element würde von einem <trainPartSequence> aus
> verwiesen (/itineraryRef/).

Ich würde die Referenz auf "itinerary" wie bisher im Element "trainPart"
belassen und damit lediglich das jetzige Element "ocpsTT" ersetzen. Die
Verschiebung der "Laufweg-Referenz" nach "train/trainPartSequence" hätte
den Nachteil, dass leer mitgeführten Zugteile nicht mehr als solche
beschrieben werden könnten.

> können (<itineraryRef Ref=... timeShift=... />).

Gekuppelte Zugteile haben exakt gleiche Halte- und Durchfahrzeiten.
Diese Redundanz soll durch das "Auslagern und Referenzieren" vermieden
werden. Wofür soll dann der Zeitversatz helfen?

Mir fallen dafür "Mustertrassen" bzw. Taktzüge ein. "Mustertrassen"
können wir gern noch separat von der Auslagerung diskutieren. Taktzüge
werden als Einzelzüge definiert, die mittels "trainGroup/trainRef" und
dem Attribut 'type="interval"' zusammengefasst werden.

> English summary

>

> there was a discussion on a possible separation of the route and times

- > from a <trainPart> into a new structure, possibly <itinerary>. This is
- > mainly to avoid the current redundancy when two or more <trainParts>
- > run coupled in one <trainPartSection>.
- >
- > All <ocpTT> and its sub-elements would move from <trainPart> to new
- > <itinerary>.
- >
- > The new structure <itineraries> would be a outside the current
- > hierarchy <train> <trainPartSection> <trainPart>, so it would be a
- > sub-structure of <timetable>.

See Trac Ticket #240:

<https://trac.railml.org/ticket/240>

- > The cross-link to a <itinerary> would come from <trainPartSection>.

As first step, I would prefer to keep the current structure and replace the element "ocpsTT" by the new "itineraryRef" element. Moving the reference into "train/trainPartSequence" would deprecate the possibility to define coupled empty train parts.

Kind regards,
Joachim

Joachim Rubröder
Schema Coordinator: railML.timetable

--

----- posted via PHP Headliner -----

Subject: Re: addendum: separation of <ocpTT> from <trainPart> - meeting 11.11.
Posted by [Joachim Rubröder railML](#) on Fri, 28 Mar 2014 14:56:13 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hallo Dirk,

Vielen Dank für Deine Zusammenfassung und die hier geordneten Gedanken.
Hier noch ein Kommentar zu Deinem Zusatz.

~~~~~

Short English summary:

The 'trainGroup' element may be used for 'interval' train grouping with 'trainRef' elements. It further helps for grouping of trains with different 'processStatus' ('planned', 'actual', 'calculated', 'toBeChecked', 'changed', 'imported' und 'other:xxx'). This

'processStatus' attribute is optional for 'trainPart', 'train' and 'trainGroup' elements.

A new attribute 'planningStatus' may be introduced for topics like

Which procedures are used by your customers from the very beginning of timetable construction up to the ad hoc timetable adjustments? Which intermediate states are transferred to other tools and may require a clear status definition in railML? This question is regarding the same train on a certain day.

The same train number is often used for trains, that run on several days at the same times along the same stations/stops. On some days the train runs shorter or even longer, but mostly the same stations/stops at the same times.

How are such trains handled by your customers? Do they get the same train number? How about the 'additionalTrainNumber'? The concept of Deutsche Bahn is already integrated with the 'scope' attribute.

The topic "change management" should be discussed separately. The idea is to export only changed data, not to export the whole train with its dependencies.

~~~~~

Dirk Br uer wrote:

Element

> Elements <train> in Betracht gezogen:

railML hat bereits ein Element 'trainGroup', in dem Z ge per 'trainRef' gruppiert werden k nnen.

<http://wiki.railml.org/index.php?title=TT:trainGroup>

M glicherweise k nnte die Typisierung ('type' Attribut) erweitert werden. Derzeit k nnen Taktz ge (type="interval") oder verschiedene Planungszust nde eines Zuges ('processStatus' Attribut) beschrieben werden.

> 1a) Instanzen des (bisherigen) Elements <train> mit identischem
> Attribut /trainNumber/, die sich hinsichtlich der Verkehrstage
> unterscheiden: Sie haben disjunkte Verkehrstage. Man spricht von

> /additionalTrainNumber/ erkennbar.

>
> 1b) Instanzen des (bisherigen) Elements <train> mit identischem

nur

> unterschiedlicher /additionalTrainNumber/.

Hier stellt sich die Frage, wie das in anderen Ländern gehandhabt wird, die nicht zwischen Hauptlauf und Nebenlauf eines Zuges unterscheiden, wie es die DB in Deutschland tut.

Es geht also um Züge, die an verschiedenen Verkehrstagen die gleiche Betriebsstellenabfolge zu gleichen Zeiten absolvieren, jedoch an ausgewählten Tagen einen Teil der Betriebsstellen weglassen oder darüber

Deutschen Bahn unterschieden?

> 2) Instanzen des (bisherigen) Elements <train> mit identischem
> Attribut /trainNumber/, die unterschiedliche Planungszustände
> repräsentieren und sich daher gegenseitig überschreiben (neuere
> überschreiben ältere).

Verschiedene Planungszustände sind mit dem 'trainGroup' Element und 'processStatus' Attribut abbildbar. Jedoch kann damit kein

Werte von 'processStatus' sollten ggf. weiter ergänzt werden. Derzeit sind 'planned', 'actual', 'calculated', 'toBeChecked', 'changed', 'imported' und 'other:xxx' vorgesehen. Deren gewünschte Verwendung sollte detaillierter beschrieben werden.

> Unter diese Kategorie fallen ebenfalls Ausfälle und Wieder-Einlegungen
> werden.

vorgesehen. Dieses Thema könnte separat diskutiert werden. Dass dafür Bedarf vorhanden ist, wurde schon häufiger signalisiert.

> Ebenso fallen hierunter solche Planungszustände wie

einigen kann, könnten diese im Attribut "processStatus" ergänzt werden. Vielleicht ist dafür auch ein neues Attribut "planningStatus" sinnvoll. Es ist sowohl im 'trainPart' als auch in 'train' und 'trainGroup' verfügbar.

Als Voraussetzung dafür wäre es hilfreich, die üblichen Planungsabläufe mit Zwischenzuständen aufzuzeigen, jedoch nur mit dem railML-Export-Gedanken im Hintergrund. Zustände, die nur in einem Planungstool bleiben und nicht exportiert werden, sollten nicht beachtet werden.

Kind regards,
Joachim

Joachim Rubner
Schema Coordinator: railML.timetable

--

----- posted via PHP Headliner -----

Subject: Re: addendum: separation of <ocpTT> from <trainPart> - meeting 11.11.
Posted by on Mon, 08 Dec 2014 12:09:37 GMT
[View Forum Message](#) <> [Reply to Message](#)

Lieber Joachim und liebe Mitlesende,

ich möchte zunächst nur zu Protokoll geben, dass der einleitende Forumseintrag von mir nur eine Art Protokoll des Treffens vom 11.11.2013 sein sollte. Nicht alles davon kann ich auch inhaltlich vertreten.

Daher sehe ich hier auch die Vertreter gefragt, die diese Anträge inhaltlich eingebracht haben. Inzwischen steht das Thema aber über ein Jahr ohne weitere Beiträge. Wenn wir die Gelegenheit eines "refactorings" für 3.0 nutzen wollen, müssen wir uns mal positionieren.

Joachim, ich stimme Dir voll zu, dass vieles von dem Beantragten auch ohne großes "refactoring" quasi auf dem kleinen Dienstweg gelöst werden könnte, wie beispielsweise durch Erweiterung von <trainGroup>.type. Einzig die eingangs (meines ursprünglichen Beitrags) erwähnte Redundanz der Zeitenabfolge `_gekuppelter_` Zugteile würde dadurch nicht beseitigt.

Es könnte aber ein durchaus in Frage kommendes (und meiner persönlichen Meinung nach auch zu begrüßendes) "Minimal-refactoring" sein, wenn man die Zeitenabfolge etwa in die von Andreas Tanner schon mal vorgeschlagene <itineraries> auslagerte und den Rest ließe wie es ist.

zu /trainNumber/, /scope/, /additionalTrainNumber/:

- > Hier stellt sich die Frage, wie das in anderen Ländern gehandhabt wird,
- > die nicht zwischen Hauptlauf und Nebenlauf eines Zuges unterscheiden,
- > wie es die DB in Deutschland tut.

....

- > Wie werden diese Szenarien außerhalb der Deutschen Bahn unterschieden?

Meiner bescheidenen, natürlich nicht vollständigen Erfahrung nach gibt es diese Phänomene in anderen Ländern so nicht.

Traditionell wird sowieso die Verkehrstagerregelung (und die Fahrplanperiode) in den Primärschlüssel eines Zuges einbezogen (also zum eindeutigen Identifizieren eines Zuges mitbenutzt). Das hat zwar Nachteile in der Planungsphase, wenn sich die Verkehrstagerregelung eines Zuges noch ändern kann. Diese Nachteile würden aber railML-seitig durch eindeutige /code/s oder (zukunftig eventuell) /UUID/s leicht beherrschbar sein. Nach der Planungsphase (bei Fahrplan-Veröffentlichungen usw.) ist die Identifikation über Verkehrstage aber m. E. überall gelebte Praxis.

Ansonsten: Die anderen mir bekannten Länder/Staatsbahnen behaupten, in vergleichbaren Fällen einfach eine neue Zugnummer zu vergeben anstatt sowas wie Hauptlauf und Nebenlauf derselben Zugnummer zu unterscheiden.

Hier kommen wir zu dem allerdings nicht ganz von der Hand zu weisen Argument, das die DB Netz AG Europas größtes EIU ist, der damit von allen als erster die (fünfstelligen) Zugnummern ausgehen. Warum keine mehr als fünfstelligen Zugnummern verwendet werden... hat wohl mit viel "alter" Software zu tun, die das nicht verarbeiten und nicht von heute auf morgen geändert werden kann.

Es ist Fakt, dass die DB Netz AG derzeit noch die Mehrfachverwendung von Zugnummern praktiziert und von ihren EVUs auch verlangt. Es ist wohl nicht am railML-Konsortium, das zu werten. Wenn wir es für railML ausschließen wollen, dann verschließen wir uns de facto vor Anwendungen mit Kompatibilität zu DB Netz. Wollen wir das?

Gedacht war, dass /trainNumber/, /scope/ und /additionalTrainNumber/ für sich immer eindeutig sein müssen, was dem DB-Netz-Fall genügt, aber auch jede Teilmenge zulässt - also keine Verwendung von /scope/ und /additionalTrainNumber/, womit die /trainNumber/ allein eindeutig zu sein braucht. Das sollte damit m. E. allen bekannten Fällen genügen. Hat jemand damit ein Problem?

- > Es geht also um Züge, die an verschiedenen Verkehrstagen die gleiche
- > Betriebsstellenabfolge zu gleichen Zeiten absolvieren, jedoch an
- > ausgewählten Tagen einen Teil der Betriebsstellen weglassen oder darüber
- > hinaus etwas weiter verkehren.

Nicht nur. Es geht vor allem auch um Züge, die an allen Tagen die gleiche Betriebsstellenabfolge absolvieren, aber nicht an allen Tagen im Detail zu den gleichen Zeiten. Guckst Du hier: [1].

- > Ein typisches "Änderungsmanagement" ist mit railML derzeit nicht
- > vorgesehen. Dieses Thema könnte separat diskutiert werden. Dass dafür
- > Bedarf vorhanden ist, wurde schon häufiger signalisiert.

Ich stimme Dir voll zu.

- > Verschiedene Planungszustände sind mit dem 'trainGroup' Element und
- > 'processStatus' Attribut abbildbar. Jedoch kann damit kein
- > "Überschreiben" also Ändern eines Zuges bedient werden.

Ich denke, es sollte eine allgemeingültigere Möglichkeit geschaffen werden, auch sich potentiell überschreibende Planungszustände abzubilden - oder weiterhin die Finger davon gelassen werden. Ich bin kein Freund halber Sachen.

- > Als Voraussetzung dafür wäre es hilfreich, die üblichen Planungsabläufe
- > mit Zwischenzuständen aufzuzeigen, jedoch nur mit dem
- > railML-Export-Gedanken im Hintergrund. Zustände, die nur in einem
- > Planungstool bleiben und nicht exportiert werden, sollten nicht beachtet
- > werden.

Auch hier stimme ich Dir voll zu: Nicht alles mögliche unterstützen, sondern nur das Gebrauchte. Ich sehe in den Planungsphasen (Jahresfahrplan, Gelegenheitsverkehr usw.) prinzipiell das gleiche wie in sich kurzfristig gegenseitig überschreibenden Phasen (Stichwort Ein- und Auslegen von Zügen).

Ich würde all das mit "Änderungsmanagement" im weitesten Sinne überschreiben und mich zum diesem Thema eher zurückhalten, da es vordringlich von anderer Seite eingebracht wurde.

Danke und viele Grüße,
Dirk.

[1] http://www.irfp.de/download/railml_beispiel_mzl.pdf
