
Subject: Wendevorgaben

Posted by [Christoph.Jobmann](#) on Wed, 16 Mar 2011 07:37:47 GMT

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

Hallo miteinander,

ich hatte im Rahmen des gestrigen Treffens in Innsbruck angemerkt, dass Interesse besteht, Wendevorgaben in RailML abbilden zu können. Zunächst möchte ich kurz ausführen, was ich mir darunter vorstelle:

In Einzelfällen kann es angebracht sein, für bestimmte Fahrten die jeweils nachfolgende Fahrt bereits als Bestandteil des Fahrplans (nicht des Umlaufplans) zu übergeben. Ein auf Basis solch eines Fahrplans erstellter Umlaufplan sollte solch eine Vorgabe dann natürlich auch berücksichtigen. In diesem Zusammenhang ist - je nach Situation - mal von einer Durchbindung, mal von einer Zwangsbindung oder auch einer Wendevorgabe die Rede.

Nach einer kurzen Diskussion kamen wir zu dem Schluss, dass dies am ehesten im Einklang mit dem timetable-Teilschema zu lösen ist, indem jene Elemente verwendet werden, die für die Abbildung von Umläufen vorgesehen sind - auch wenn de facto noch kein (vollständiger) Umlauf vorliegt.

Ein Kritikpunkt an dieser Vorgehensweise war zunächst noch, dass diese Vorgaben nicht mehr erkennbar wären, wenn für diesen Fahrplan ein Umlaufplan erstellt und wieder exportiert würde. Die "fixed"-Attribute der Elemente "blockPart" und "block" sehen zwar vielversprechend aus, können jedoch nicht sinnvoll genutzt werden, da die zu fixierenden

vorliegen.

und bitte hiermit um Rückmeldung, ob sie für brauchbar gehalten wird.

Es wird ein neuer Typ "tRosteringScope" eingeführt, welcher als Vorgabe die Werte "timetable" oder "schedule" annehmen kann.

Weiter erhält das Element "rostering" ein zusätzliches Attribut namens "scope", welches den Typ "tRosteringScope" hat.

Ein "rostering"-Element mit "scope=timetable" würde dann Wendevorgaben / Zwangs- / Durchbindungen beschreiben. Dabei würden vermutlich nicht alle Elemente zum Einsatz kommen: "circulations" und "circulation" würde beispielsweise nicht benötigt.

Ein "rostering"-Element mit "scope=schedule" würde - wie bisher - einen Umlaufplan abbilden.

Für Rückmeldungen jeglicher Art bin ich dankbar.

Folgende Punkte habe ich dabei noch bewusst offen gelassen:

- Wie sind Zwangsbindungen anzulegen, die über die Grenze eines "block"-Elements hinausgehen? [Vorschlag: für "scope=timetable" kann die Länge eines "block" von der Länge für "scope=schedule" abweichen.]
- Ist für jede Zwangsbindung ein eigenes "rostering"-Element anzulegen, oder können/sollten alle in einem untergebracht werden? [Vorschlag: Ein "rostering"-Element pro Zwangsbindung]

Christoph Jobmann

DB Mobility Logistics AG

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

Subject: Re: Wendevorgaben

Posted by [Joachim Rubröder railML](#) on Wed, 23 Mar 2011 10:02:11 GMT

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

Hallo Christoph,

die Idee, solche durch den Fahrplan bedingten Wendevorgaben als eine Form der Umlaufplanung anzusehen finde ich konsequent. Da gehören sie meiner Ansicht nach hin.

Die Unterscheidung beim Element "rostering" über ein zusätzliches Attribut namens

"scope" mit Typ "tRosteringScope" finde ich auch gut. Allerdings empfinde ich die Namenswahl "schedule"/"timetable" etwas unglücklich. Wir haben ohnehin schon zweierlei Auffassungen, was so ein "rostering" eigentlich ist. Mein Vorschlag wäre daher:

"timetable" - für im Fahrplan vorgesehene Wendevorgaben (Zugformationen, Verkehrstage)

"conceptual" - für konzeptionelle Umlaufpläne (Fahrzeuggruppen, Normwochen)

"operational" - für betriebliche Umlaufpläne (einzelne Fahrzeuge, Kalendertage)

Wenn für den neuen timetable-rostering die "circulations" nicht benötigt werden, haben wir allerdings ein kleines Problem, da es sich dabei um Pflichtelemente handelt.

Was spricht denn genau dagegen, je Wendevorgabe genau ein Element circulation zu genau einen "fixed" block mit genau zwei blockParts anzulegen? Damit entspräche jede Wendevorgabe einem eigenen kleinen

offenen Teil-Umlauf.

Viele GrÃ¼sse,
Joachim RubrÃ¶der

--

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

Subject: Re: Wendevorgaben
Posted by [Christoph Jobmann](#) on Thu, 24 Mar 2011 06:14:57 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hallo Joachim,

vielen Dank fÃ¼r die Antwort.

Die Anmerkungen und Bedenken kann ich nachvollziehen und bin mit den
Alternativ-VorschlÃ¤gen einverstanden.

Christoph Jobmann

--

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

Subject: Re: Wendevorgaben
Posted by [Joachim RubrÃ¶der railML](#) on Wed, 30 Mar 2011 07:19:40 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hallo Christoph,
das entsprechende scope-Feld habe ich aufgenommen.

Sind damit aus Deiner Sicht die dringenden Anforderungen fÃ¼r eine
Version 2.1 erfÃ¼llt, oder benÃ¶tigt Du noch weitere Felder?

Viele GrÃ¼sse,
Joachim RubrÃ¶der

--

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

Subject: Re: Wendevorgaben

Posted by [Joachim Rubröder railML](#) on Fri, 01 Apr 2011 07:38:12 GMT

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

Hallo Christoph,

als Kosequenz der jetzt gefundenen Modellierungsalternativen habe ich die alten Ausprägungsmöglichkeiten der connOperation entsprechend reduziert und die Werte 'join', 'split' und 'turnaround' als DEPRECATED deklariert (commit 383).

Viele Grüße,
Joachim Rubröder

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

Subject: Re: Wendevorgaben

Posted by on Mon, 04 Apr 2011 12:49:52 GMT

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

Hallo Herr Jobmann und hallo miteinander,

das Interesse an den Wendevorgaben ist für mich durchaus nachvollziehbar. Wir sollten nur klar unterscheiden (und das ist ja wohl auch so angesprochen worden) zwischen Soll-Vorgaben und dem oder den konkreten Umlaufplan/-plänen.

Bei vollständiger Angabe von Wendevorgaben an allen Zügen ließe sich der Umlaufplan eindeutig rekonstruieren. Es sollte hier aber nicht darum gehen, eine zweite Möglichkeit zur Umlaufbeschreibung in RailML einzubringen - Redundanzen sind immer schlecht vor allem für ein lesendes Programm. Vielmehr sollten es wirklich "Wendevorgaben" im Sinne von Zwangsübergängen (-> Teilmenge eines Umlaufs) oder vor allem für den Fall, dass noch gar kein Umlauf besteht, sein.

Ein etwaiger Widerspruch zwischen den Umlauf und Wendevorgabe sollte mit klarer Priorität versehen sein, vmtl. auf dem Umlauf. Die Wendevorgaben haben also den Charakter "nice to have".

Solche Wendevorgaben waren unter dem Oberbegriff "fahrplantechnische Bindungen" bereits vorgesehen. Leider ist dies (wie so vieles) in den Wirren von RailML 0.9x und der "Verenglischung" untergegangen. Wir finden die unsterblichen Reste in der fahrplantechnischen Bindungen in trainPart -> ocpTT -> connections. Dass es hier wirklich um alle fahrplantechnischen

Bindungen und nicht nur um Anschlüsse (wie die Übersetzung suggeriert) geht, erkennt man am Element "connOperation": Hier gibt es auch "turnaround", was auf die Wendevorgaben abzielt.

Natürlich sind die alten fahrplantechnischen Bindungen - egal, ob nun Anschlüsse oder Wendevorgaben - nicht hinreichend (sind sie in RailML nie gewesen, da vmtl. etwas schwieger-, äh stiefmütterlich behandelt). Vor allem fehlt eine Möglichkeit, sie verkehrstage- und zugteilabhängig zu machen. Für Anschlüsse mag es egal (oder selbsterklärend) sein, auf welchen Zugteil der Anschluss stattfindet; spätestens für Wendevorgaben ist es nicht mehr egal.

Alternativ würde ich vorschlagen, im Umlauf selbst bei den Übergängen (Abfolgen) eine Möglichkeit zu schaffen, dort anzugeben, ob der Übergang aus einer Zwangsbindung heraus festgelegt wurde oder ob er "frei" (z. B. aus einer Optimierung heraus) entstanden ist. Damit ließen sich die Wendevorgaben ermitteln durch "scannen" des Umlaufs auf Zwangsbindungen hin. Als solche Zwangsbindungs-Kennzeichnung würde theoretisch zunächst ein boolescher Wert reichen, und wir kämen damit zunächst viel einfacher zum Ziel. (Im Prinzip geht es schon indirekt durch die Zusammenfassung von blockParts zu blocks, allerdings ist dies nicht zwangsläufig auf Zwangsübergänge beschränkt.)

Fazit (mein Vorschlag):

- Ausbau des connection-Elements auf die Möglichkeit, die fahrplantechnische Bindung etwas exakter zu formulieren (z. B. explizit Zwangsübergang) und sie verkehrstage- und zugteilabhängig machen (Aufnahme von operatingPeriodRef und trainPartRef)

- Aufnahme eines booleschen Werts bei den connections zur Kennzeichnung von Übergängen im Umlauf, die auf Grund von Zwangsbindungen entstanden sind.

- Da wir eine (im Prinzip derzeit schon bestehende) Redundanz haben: Beschreibung der Priorität, dass im Zweifelsfall der konkrete Umlauf (circulation) Vorrang vor der Wendevorgabe (connection) hat und die Wendevorgabe mehr so ein Sollwert sein soll...

Viele Grüße,
Dirk Bräuer.

Am 16.03.2011, 08:37 Uhr, schrieb Christoph Jobmann
<christoph.jobmann@deutschebahn.com>:

> Hallo miteinander,
>

- > ich hatte im Rahmen des gestrigen Treffens in Innsbruck angemerkt, dass
- > Interesse besteht, Wendevorgaben in RailML abbilden zu können. Zunächst
- > möchte ich kurz ausführen, was ich mir darunter vorstelle:
- >
- > In Einzelfällen kann es angebracht sein, für bestimmte Fahrten die
- > jeweils nachfolgende Fahrt bereits als Bestandteil des Fahrplans (nicht
- > des Umlaufplans) zu übergeben. Ein auf Basis solch eines Fahrplans
- > erstellter Umlaufplan sollte solch eine Vorgabe dann natürlich auch
- > berücksichtigen. In diesem Zusammenhang ist - je nach Situation - mal
- > von
- > einer Durchbindung, mal von einer Zwangsbindung oder auch einer
- > Wendevorgabe die Rede.
- >
- > Nach einer kurzen Diskussion kamen wir zu dem Schluss, dass dies am
- > ehesten im Einklang mit dem timetable-Teilschema zu lösen ist, indem
- > jene
- > Elemente verwendet werden, die für die Abbildung von Umläufen
- > vorgesehen
- > sind - auch wenn de facto noch kein (vollständiger) Umlauf vorliegt.
- >
- > Ein Kritikpunkt an dieser Vorgehensweise war zunächst noch, dass diese
- > Vorgaben nicht mehr erkennbar wären, wenn für diesen Fahrplan ein
- > Umlaufplan erstellt und wieder exportiert würde. Die "fixed"-Attribute
- > der Elemente "blockPart" und "block" sehen zwar vielversprechend aus,
- > können jedoch nicht sinnvoll genutzt werden, da die zu fixierenden
- > Entitäten die Überbergänge sind - für welche aber keine konkreten
- > Objekte
- > vorliegen.
- >
- > Für diese Problemstellung habe ich schließlich eine Lösung gefunden -
- > und bitte hiermit um Rückmeldung, ob sie für brauchbar gehalten wird.
- >
- > Diese Lösung stelle ich mir folgendermaßen vor:
- > Es wird ein neuer Typ "tRosteringScope" eingeführt, welcher als Vorgabe
- > die Werte "timetable" oder "schedule" annehmen kann.
- > Weiter erhält das Element "rostering" ein zusätzliches Attribut namens
- > "scope", welches den Typ "tRosteringScope" hat.
- > Ein "rostering"-Element mit "scope=timetable" würde dann Wendevorgaben /
- > Zwangs- / Durchbindungen beschreiben. Dabei würden vermutlich nicht alle
- > Elemente zum Einsatz kommen: "circulations" und "circulation" würde
- > beispielsweise nicht benötigt.
- > Ein "rostering"-Element mit "scope=schedule" würde - wie bisher - einen
- > Umlaufplan abbilden.
- >
- > Für Rückmeldungen jeglicher Art bin ich dankbar.
- >
- > Folgende Punkte habe ich dabei noch bewusst offen gelassen:
- > - Wie sind Zwangsbindungen anzulegen, die über die Grenze eines

> "block"-Elements hinausgehen? [Vorschlag: für "scope=timetable" kann
> die
> Länge eines "block" von der Länge für "scope=schedule" abweichen.]
> - Ist für jede Zwangsbindung ein eigenes "rostering"-Element anzulegen,
> oder können/sollten alle in einem untergebracht werden? [Vorschlag: Ein
> "rostering"-Element pro Zwangsbindung]
>
> Viele Grüße
> Christoph Jobmann
>
> DB Mobility Logistics AG
>

--

Erstellt mit Operas revolutionärem E-Mail-Modul: <http://www.opera.com/mail/>

Subject: Re: Wendevorgaben

Posted by [Christoph Jobmann](#) on Wed, 06 Apr 2011 03:01:41 GMT

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

Hallo Herr Bräuer und weitere Mitlesende,

ich hielt es zunächst auch für die naheliegendste Lösung, das connection-Element zu nutzen, was aber beim Treffen in Innsbruck weitgehend abgelehnt wurde. Die Variante der rosterings mit scope timetable wurde als sauberer im Sinne von railML gehalten und daher auch durch Herrn Rubner umgesetzt.

Den vorgeschlagenen Ausbau des connection-Elements halte ich damit nicht für zielführend. Hinzu kommt, dass ich den Nutzen des vorgeschlagenen boolschen Attributs zur Identifikation von durch Zwangsbindung vorgegebene

Interessant finde ich jedoch den Punkt der Priorität für Wendevorgaben: Tatsächlich ist es Bestandteil unserer Optimierung, dass grundsätzlich eine Wende vorgegeben werden kann, die jedoch nicht zwangsläufig eingehalten werden muss. Ob bzw. wie eine entsprechende Information jedoch übergeben werden sollte ist mir zunächst nicht klar, da wir beispielsweise mit "Modellkosten pro nicht umgesetzter Zwangsbindung" arbeiten und nicht mit Prioritäten, weshalb ich eine entsprechende Erweiterung vermutlich nicht für 2.1 für sinnvoll halte.

Christoph Jobmann

Dirk Bräuer wrote:

>

> Hallo Herr Jobmann und hallo miteinander,

>

> das Interesse an den Wendevorgaben ist für mich durchaus nachvollziehbar.

> Wir sollten nur klar unterscheiden (und das ist ja wohl auch so

> angesprochen worden) zwischen Soll-Vorgaben und dem oder den konkreten

> Umlaufplan/-plänen.

>

> Umlaufplan eindeutig rekonstruieren. Es sollte hier aber nicht darum

> gehen, eine zweite Möglichkeit zur Umlaufbeschreibung in RailML

> einzubringen - Redundanzen sind immer schlecht vor allem für ein lesendes

> Programm. Vielmehr sollten es wirklich "Wendevorgaben" im Sinne von

> Zwangsübergängen (-> Teilmenge eines Umlaufs) oder vor allem für den
> Fall,

> dass noch gar kein Umlauf besteht, sein.

>

> Ein etwaiger Widerspruch zwischen den Umlauf und Wendevorgabe sollte mit

> klarer Priorität versehen sein, vmtl. auf dem Umlauf. Die Wendevorgaben

> haben also den Charakter "nice to have".

>

> Solche Wendevorgaben waren unter dem Oberbegriff "fahrplantechnische

> Bindungen" bereits vorgesehen. Leider ist dies (wie so vieles) in den

> Wirren von RailML 0.9x und der "Verenglichung" untergegangen. Wir finden

> die unsterblichen Reste in der fahrplantechnischen Bindungen in trainPart

> -> ocpTT -> connections. Dass es hier wirklich um alle fahrplantechnischen

> geht, erkennt man am Element "connOperation": Hier gibt es auch

> "turnaround", was auf die Wendevorgaben abzielte.

>

> Natürlich sind die alten fahrplantechnischen Bindungen - egal, ob nun

> Anschlüsse oder Wendevorgaben - nicht hinreichend (sind sie in RailML nie

> gewesen, da vmtl. etwas schwieriger-, eher stiefmütterlich behandelt). Vor

> allem fehlt eine Möglichkeit, sie verkehrstage- und zugteilabhängig zu

> machen. Für Anschlüsse mag es egal (oder selbsterklärend) sein, auf

> welchen Zugteil der Anschluss stattfindet; spätestens für Wendevorgaben

> ist es nicht mehr egal.

>

> aus einer Zwangsbindung heraus festgelegt wurde oder ob er "frei" (z. B.

> Wendevorgaben ermitteln durch "scannen" des Umlaufs auf Zwangsbindungen

> hin. Als solche Zwangsbindungs-Kennzeichnung würde theoretisch zunächst

> ein boolescher Wert reichen, und wir kämen damit zunächst viel einfacher

> zum Ziel. (Im Prinzip geht es schon indirekt durch die Zusammenfassung von

> blockParts zu blocks, allerdings ist dies nicht zwangsläufig auf

- > Zwangsübergang beschränkt.)
- >
- > Fazit (mein Vorschlag):
- > - Ausbau des connection-Elements auf die Möglichkeit, die
- > fahrplantechnische Bindung etwas exakter zu formulieren (z. B. explizit
- > Zwangsübergang) und sie verkehrstage- und zugteilabhängig machen (Aufnahme

- > von operatingPeriodRef und trainPartRef)
- >
- > - Aufnahme eines booleschen Werts bei den connections zur Kennzeichnung

- > sind.
- >
- > - Da wir eine (im Prinzip derzeit schon bestehende) Redundanz haben:
- > Beschreibung der Priorität, dass im Zweifelsfall der konkrete Umlauf
- > (circulation) Vorrang vor der Wendevorgabe (connection) hat und die
- > Wendevorgabe mehr so ein Sollwert sein soll...
- >

- > Dirk Bräuer.

--

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

Subject: Re: Wendevorgaben
Posted by [Christoph Jobmann](#) on Wed, 06 Apr 2011 03:03:52 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hallo Joachim,

vielen Dank für die Erweiterung und die Klarstellung bezüglich des
connOperation-Wertebereichs.
Darüber hinaus benötige ich dringend für 2.1 keine weiteren Felder.

Christoph Jobmann

Joachim Rubner wrote:

- >
- > Hallo Christoph,
- > das entsprechende scope-Feld habe ich aufgenommen.
- >
- > Sind damit aus Deiner Sicht die dringenden Anforderungen für eine
- > Version 2.1 erfüllt, oder benötigst Du noch weitere Felder?

>
> Viele GrÃ¼sse,
> Joachim RubrÃ¼der
>
>

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