
Subject: Some small changes, wishes and recommendations for RailML2 timetable schemes

Posted by on Mon, 25 Oct 2010 14:57:53 GMT

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

Dear RailML2 users and all who want it to become,

based on the experience of the usage of RailML2 during the last months there are some changes, wishes and recommendations from iRFP for RailML2 schemes. Some of these changes are mandatory from our side, so we have defined an own derivation of the schemes as an intermediate solution. We hope this can be adopted to the "official" RailML schemes. (Our derivation is available at http://www.irfp.de/download/schemata_fbs_railml.zip.)

-- English -- (German version follows in relative braking distance)

1) in railwayBaseTypes.aPlaces correction:

from `<xs:attribute name="count" type="rail:tCounter" use="required" />`

to `<xs:attribute name="count" type="xs:nonNegativeInteger"`

`use="required"/>`

(it should be possible to set the no. of seats explicitly to zero)

Reason: Although it is normally (at the formation) not necessary to set the no. of seats to zero, it actually is necessary at the train usage of a formation. One can have vehicles with seats (>0) at the formation but with seats set to zero at the usage - this means that the vehicle is closed for passengers. This way it is also possible to define an empty run without explicit knowledge of the train's categories.

2) timetableTypes.aOperatingCode added:

`<xs:attribute name="onRequest" type="xs:boolean" use="optional"`

`default="false"/>`

(to define operation codes as "on request")

Reason: Anywhere it should be possible to define a train as running on request (additionally or only). We need solutions for

"runs Monday to Friday and on request only"

as well as

"runs Monday to Friday, and additionally Saturday and Sunday on request". That's why we should declare it at operatingCode.

3) timetableTypes.FormationTT added:

`<xs:attribute name="load" type="rail:tWeightTons"/>`

(load of a formation in tons)

Reason: So far, there is a "weight" only (including mass of engine). Experts normaly use load (weight without engine mass). This is useful because the mass of an engine may be debatable.

4) timetableTypes.aBreakUsage added:

```
<xs:attribute name="regularBrakePercentage"
type="rail:tBrakePercentage"/>
<xs:attribute name="emergencyBrakePercentage"
type="rail:tBrakePercentage"/>
```

(brake settings of a train)

Reason: A timetable should differ brake usage for emergency cases and for regular cases (because some of the auxiliary brakes are only to be used at emergencies - e. g. Mg - while others are not fail-safe and therefore are not to be counted for emergency brake capacity - e. g. electric or hydraulic brakes).

Our suggestion is to use "regularBrakePercentage" and "emergencyBrakePercentage" when explicitly wanting to distinguish between them but to use (the already defined) "BrakePercentage" (w/o "regular" or "emergency") when leaving uncertain which one is meant.

5) timetableTypes.aBooking added:

```
<xs:attribute name="posInFormation" type="xs:integer"/>
```

(position of a vehicle in a formation)

Reason: A vehicleRef can exist several times in a formation but there has been no chance to define which one has which booking number. So there has to be an index which makes it possible to link this information from a train to a formation.

6) timetableTypes.aCategory added:

```
<xs:attribute name="abbreviation" type="xs:string"/>
```

(abbreviaton of a train category)

Reason: It might have been forgotten to define the abbreviation of a train category. Maybe it was thought to be written in "name" but this might be very capable of being misunderstood. One should think that "InterCityExpress" is a name and "ICE" is the abbreviation. The attribute description is for the typical phrases like "fast running long distance train with a wide range of inside temperature".

7) timetableTypes.aTrainPart added:

```
<xs:attribute name="line" type="xs:string"/>
```

(name of line of a train part)

Reason:

So far there is only a "trainLine" defined. But this is not the same as a line of a train part (e.g. when coupling and sharing of train parts of different lines). So, we should need a "trainPartLine" but to bring this in line with other attributes of a train part the prefix "trainPart" is not used.

Hint: There is also an attribute "trainNumber" defined there which does not include the train part number (or name) in the data coming from FBS. The name of the train part is filled in trainPart.name.

8) timetableTypes.aSectionTT added:

```
<xs:attribute name="trackRef" type="rail:tGenericRef"/>
```

(link from a train to the used track in a line section)

Reason: There is a "lineRef" defined to get the information of the used line only but not the information of the used track. The attribute "trackInfo" is not a defined link. But the link is required to make sure which one of the tracks is use.

9a) timetableTypes.aCirculation added:

```
<xs:attribute name="nextOperatingPeriodRef" type="rail:tGenericRef"/>
```

(link to the operation day of the next train run inside of the circulation)

Reason: The "name" of the next block is not enough to describe and rebuild the correct circulation plan. There has to be an information to which operation day of the next block the current block should be linked to. A vehicle can make a standstill for serveral days but the defined next block inside of the cirulation may run every day.

9b) timetableTypes.aCirculation deleted:

```
use="required" at <xs:attribute name="nextBlockRef" type="rail:tGenericRef"/>
```

(nextBlockRef is optional)

Reason: Open circulation plans can not fill this attribute at the last block (of each vehicle) otherwise they would not be _open_ circulations.

9c) timetableTypes.aCirculation added:

```
<xs:attribute name="vehicleIdx" type="xs:nonNegativeInteger"/>
```

```
<xs:attribute name="groupIdx" type="xs:nonNegativeInteger"/>
```

(group and vehicle number of a run or job inside of the circulation)

Reason: This attribute is completely redundant because the information can be recreated by counting links (nextBlock...) which link to earlier days (vehicleIdx) or to lower vehicles number (groupIdx). Nevertheless, we have been asked by other users to provide these values to get the data easier and to avoid misunderstandings (skipping of vehicles during standstill).

Infrastructure:

The element "mileageChange" includes the attributes "absPosIn" and "type".

Both of them are now defined as "use=required". This should be changed to "use=optional".

It should be possible to define a mileageChange at the very beginning of a line as an initial mileage change or descriptor. Such an initial mileage change has no "absPosIn" and needs no "type" (or the type is something like "initial").

In opposition to this, the attribute "dir" has to be given at every line. There is no situation known which makes it impossible to fill this attribute. So the use should be changed from "optional" to "required".

The element OperationControlPoint includes the attribute "abbreviation" which misses an "i".

An additional element is defined as "Line" which includes the following attributes:

```
<xs:attribute name="uicNumber" type="xs:positiveInteger"
use="optional"/>
<xs:attribute name="lineNumber" type="xs:string" use="optional"/>
```

It should be possible to define these data independently of the attributes "name" or "description".

Some other hints:

The zip-code of an ocp should be defined as a string. For example in Great Britain it is not possible to use a numeric-only data type. Also, the use should be changed to optional.

The propService may also include the information of an airport connection (consequently, since ship and bus are provided).

-- German version --

Timetable:

1) in railwayBaseTypes.aPlaces korrigiert:

```
von    <xs:attribute name="count" type="rail:tCounter" use="required" />
auf    <xs:attribute name="count" type="xs:nonNegativeInteger"
use="required"/>
```

(Sitzplatzanzahl darf auch 0 sein)

Begründung: Es muss möglich sein, eine Sitzplatzanzahl beim Zug (usage) explizit auf 0 zu setzen, obwohl in der Formation eine Anzahl >0 angegeben wurde. Dies entspricht dem expliziten Absperren / nicht freigeben von Plätzen im Zug, ggf. dem Absperren ganzer Wagen. So kann man auch deutlich sagen, welche Züge Leerzüge sind, ohne dass das lesende Programm

sich der Bedeutung der Zuggattungen bewusst sein muss.

2) in `timetableTypes.aOperatingCode` hinzugefügt:

```
<xs:attribute name="onRequest" type="xs:boolean" use="optional" default="false"/>
```

(Verkehrstageregelung als "bei Bedarf" kennzeichnen)

Begründung: Irgendwo müssen wir angeben, dass ein Zug (nur/auch) bei Bedarf verkehrt. Dabei werden Lösungen gebraucht für "verkehrt Mo-Fr und nur bei Bedarf" sowie für "verkehrt Mo-Fr immer sowie Sa+So nur bei Bedarf" - daher bei `OperatingCode`.

3) in `timetableTypes.FormationTT` eingefügt:

```
<xs:attribute name="load" type="rail:tWeightTons"/>
```

(Last einer Formation in t)

Begründung: Bisher gibt es dort nur "weight" (also Masse inkl. Tfz.); in der Fachwelt ist jedoch der Wert "Last" (also Masse exkl. Tfz.) weiter verbreitet und ungeachtet der Gesamtmasse wichtig (da die Masse des Tfz. nicht immer unstrittig ist).

4) in `timetableTypes.aBreakUsage` eingefügt:

```
<xs:attribute name="regularBrakePercentage" type="rail:tBrakePercentage"/>
```

```
<xs:attribute name="emergencyBrakePercentage" type="rail:tBrakePercentage"/>
```

(Bremseinstellungen eines Zuges)

Begründung: Bisher gibt es dort nur "brakePercentage". Aus bekannten Gründen brauchen wir beim Fahrplan die Unterscheidung von Betriebs- und Zwangsbremshundertstel. Um hier Klarheit zu schaffen, sollte es entweder die expliziten "regular..." und "emergency..." geben oder die (bereits bestehenden) "brakePercentage" ohne weitere Spezifikation, wobei bei Verwendung der letzteren offen bleibt, ob sie die Betriebs- oder Zwangsbremshundertstel angeben.

5) in `timetableTypes.aBooking` eingefügt:

```
<xs:attribute name="posInFormation" type="xs:integer"/>
```

(Stellung eines Fahrzeugs -vehicleRef- innerhalb der Formation)

Begründung: Eine `vehicleRef` kann innerhalb einer Formation mehrfach vorkommen, d. h. es kann mehrere Wagen(gruppen) der gleichen Gattung geben (z. B. Bom am Zuanfang und -ende, dazwischen Aom, WRm, Bdomsb usw.). Um hier klar anzugeben, auf welchen der Wagen vom Typ `vehicleRef` sich die Reservierungsnummer bezieht, muss es einen Index geben.

6) in `timetableTypes.aCategory` eingefügt:

```
<xs:attribute name="abbreviation" type="xs:string"/>
```

(Abkürzung einer Zuggattung)

Begründung: sollte eigentlich selbstverständlich sein... So komisch es kling, aber die Abkürzung der Zuggattung scheint man übersehen zu haben. Möglicherweise war "Name" dafür vorgesehen; ich halte das aber nicht für selbstverständlich. Man sollte meinen, dass "ICE" die Abkürzung und "InterCityExpress" der Name ist. Description wäre dann sowas wie

"schnellfahrender Fernreisezug mit undefiniert, bisweilen extremen Innentemperaturen".

7) in timetableTypes.aTrainPart eingefügt:

```
<xs:attribute name="line" type="xs:string"/>
```

(Linienbezeichnung des Zugteils)

Begründung: Bisher gibt es dort nur "trainLine", das ist aber nicht dasselbe wie die Linie des Zugteils und dürfte auch nicht als solche verstanden werden. Richtig wäre "trainPartLine", wobei das übergeordnete Element (trainPart) offensichtlich grundsätzlich nicht wiederholt wird.

8) in timetableTypes.aSectionTT eingefügt:

```
<xs:attribute name="trackRef" type="rail:tGenericRef"/>
```

(Querverweis auf ein vom Zug in diesem Abschnitt benutztes Streckengleis)

Begründung: Es gibt dort bisher nur lineRef, was als Querverweis auf die Strecke verwendet wird. Es gibt außerdem trackInfo, was kein definierter Querverweis ist. Es ist jedoch für die Regelgleis-/Gegengleis-Unterscheidung wichtig, auch eine klare Referenz auf das Streckengleis zu haben.

9a) in timetableTypes.aCirculation eingefügt:

```
<xs:attribute name="nextOperatingPeriodRef" type="rail:tGenericRef"/>
```

(Querverweis auf Verkehrstag des Nachfolgers im Umlauf)

Begründung: Die alleinige Angabe der Nachfolgefahrt / des Nachfolgeeinsatzes genügt nicht, da der nachfolgende Einsatz an mehreren Verkehrstagen vorkommen kann und nicht zwangsläufig das nächste Vorkommen gemeint ist (Stillstand über mehrere Tage, z. B. Wochenende).

9b) in timetableTypes.aCirculation entfernt: use="required" bei

```
<xs:attribute name="nextBlockRef" type="rail:tGenericRef"/>
```

(nextBlockRef ist nicht Pflicht, sondern optional)

Begründung: Bei offenen Umläufen hat die letzte Fahrt im Umlauf per Definition keinen Nachfolger (sonst wäre es kein offener Umlauf). Offene Umläufe gibt es für Einzeltage / Feiertage / Überleitungen zwischen anderen Umläufen / Fahrplanwechsel.

9c) in timetableTypes.aCirculation eingefügt:

```
<xs:attribute name="vehicleIdx" type="xs:nonNegativeInteger"/>
```

```
<xs:attribute name="groupIdx" type="xs:nonNegativeInteger"/>
```

(Gruppen- und Fahrzeugnummer einer Fahrt bzw. eines Einsatzes im Umlauf)

Begründung: Diese Attribute sind völlig redundant, da jederzeit durch Zählen der Rückverweise auf zurückliegende Tage (vehicleIdx) bzw. auf zurückliegende Fahrzeuge (groupIdx) erzeugbar. Dennoch wurde der Wunsch an uns herangetragen, diese Elemente zur leichteren Auswertung und Vermeidung von Missverständnissen (überspringen von Fahrzeugnummern bei Leerstand) vorzusehen.

Infrastruktur:

Im Element `MileageChange` sollten die Eigenschaften der attribute `"absPosIn"` und `"type"` von `"use=required"` in `"use=optional"` geändert werden. Für das Attribut `"dir"` sollte die Eigenschaften `"use=optional"` in `"use=required"` geändert werden.

Begründung: Der Konsequenz und einfacheren Auswertbarkeit sollten wir einen `"MileageChange"` auch am Anfang der Strecke angeben, der den ursächlichen Zusammenhang zwischen relativer und absoluter Kilometrierung angibt. Bei diesem initialen `"MileageChange"` ist `"absPosIn"` nicht zutreffend und `"type"` ist trivial. Hingegen gibt es keinen Grund, die Kilometrierungsrichtung (`"dir"`, aufsteigend/fallend) irgendwo nicht anzugeben. Die Liste der `MileageChanges` gibt dann einen kompakten Formelsatz an, mit dem man zu jeder Position der Strecke die abs. km ausrechnen kann.

Das Attribut `"abbreviation"` im Element `OperationControlPoint` muss um ein fehlendes `i` erweitert werden.

Zusätzlich wurden folgende Informationen im Element `"line"` ergänzt:

```
<xs:attribute name="uicNumber" type="xs:positiveInteger"
use="optional"/>
<xs:attribute name="lineNumber" type="xs:string" use="optional"/>
```

(UIC- und Streckennummer der Strecke)

Begründung: Irgendwo müssen sowas angegeben werden können. Es ist günstiger, wenn dies in diesen Attributen erfolgt, anstatt es in `"name"` oder `"description"` zu pressen.

Weiterhin sind folgende Dinge aufgefallen:

Der `operationalType` eines `ocp` ist in den möglichen Werten (zumindest nach deutscher Lesart) unvollständig. Die andere Frage ist allerdings, warum das überhaupt noch in dieser Art angegeben werden muss, obwohl es implizit in anderen Eigenschaften steckt. Eventuell wäre hier ein Für-obsolet-Erklären das Beste.

Die Postleitzahl (`zip`) eines `ocp` ist nicht zwingend alphanumerisch (siehe z.B. GB). Hier sollte ein `string` wohl die bessere Lösung sein. Weiterhin erscheint es fragwürdig, ob man nicht einen Gemeindennamen auch ohne die Kenntnis einer Postleitzahl angeben kann? Hier sollte wohl die Nutzung auf `"optional"` gesetzt werden.

Das Element `"propService"` sollte auch die Angabe eines Flughafenbahnhofes ermöglichen (wenn es nun schon Schiff und Bus gibt...).

-- FINÉ --
