
Subject: Re: timetable-Schema 0.93

Posted by [Raik Hoffmann](#) on Fri, 21 Jun 2002 13:23:37 GMT

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

Hallo,

ich habe fuer die Unterscheidung der train's die Attribute type, source und date ergaenzt.

type ist ein string-typ und soll fuer die Unterscheidung nach Art von Eintraegen stehen.

source ist ebenfalls string-typ und steht fuer die Datenherkunft.

date ist vom date-Datentyp und steht fuer das Datum des Zuglaufes.

Was fuer moegliche Auspraegungen waeren bei type und source denkbar? Eventuell ist es besser, die nachgelagerte IST-Auswertung eines Zuglaufes komplett in das STATISTICS-Schema zu uebernehmen?

Zu der Benennung der ID bei entry.

Diese ID kann nicht nur auf Bahnhofe verweisen, sondern referenziert - abstrakt gesprochen - einen "Fahrzeitmesspunkt". An diesem Fahrzeitmesspunkt kann sich (quasi zufaellig) auch ein Bahnhof, ein Signal, Weiche, ... befinden. Im einfachsten Fall ist dies aber einfach nur ein Punkt auf der freien Strecke.

> Was ich auch noch gut faende, ist wenn man pro entry mehrere trackIDs
> angeben koennte (z.B. nach Prioritaeten alle in der Station benutzbaren).

Gehoert das nicht eher in ein Bahnhofsfahrordnungs-Schema oder Betriebsmanagement-Schema?

Der Zug sucht sich doch nicht (aktiv) einen Weg anhand seiner mitgegebenen Informationen durch den Bahnhof, sondern wird (passiv) auf einem vorgegebenen Weg durch den Bahnhof gefuehrt.

Allerdings kann man im header eines trains eine "stationTrack"-list referenzieren. Diese Liste beinhaltet ja die zu befahrenden Gleise in einem Bahnhof. In dieser koennte man auch Prioritaeten oder Regeln fuer die zu befahrenden Gleise angeben...

> Das im Beispielfile timetable.xml verwendete Vermischen von Zeitwerten
> und Typ des Fahrplaneinrages (z.B. arrival = "pass") wuerde ich
> vermeiden und wuerde dies in einen speziellen Type des
> Fahrplaneinrages reinnehmen (z.B. begin, end, stop, pass) .

Ja, ich bin mit der existierenden Loesung auch nicht zufrieden.
Ich hatte das auf dem Auftakttreffen schon versucht anzusprechen.

Das bisherige System im Schema funktioniert wie in der Tabelle unter

http://www.railml.org/table_1.gif dargestellt.

Die Attribute arrival und departure sind dabei required.

Es gibt natuerlich die Moeglichkeit, fuer jedes Szenario spezielle Attribute einzufuehren, oder nur Zeitwerte in den Attributen zuzulassen. Ich habe aber festgestellt, dass sich dies nur mit erhoehetem Aufwand implementieren laesst. Man muss also jedesmal auf eine z.B. Durchfahrt testen, indem man die Werte in arrival und departure auf Gleichheit prueft.

Die Attribute optional zu machen und damit weitere optionale Attribute begin, end und pass einzufuehren blaecht die verarbeitende Routine auf. Da diese aber *sehr* oft durchlaufen werden muss, wuerde ich dabei auf kurze Rechenzeit optimieren, also so wie in der Tabelle...

Bin mit der Loesung aber auch noch nicht ganz zufrieden und freue mich auf weitere Vorschlaege!

Viele Gruesse,
Raik Hoffmann
