

---

Subject: Re: timetable-Schema 0.93

Posted by [huerlimann](#) on Tue, 02 Jul 2002 09:12:50 GMT

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

---

Hallo zusammen

Besten Dank an Raik für die Ergänzung des Schemas mit den bei den Trains angesiedelten Attributen type, source und date.

Was mir aufgefallen ist, ist das beim aktuellen Format zwei nicht gleichbedeutende Elemente (Tags) timetable gibt, einmal unter railml und einmal unter train. Dies sollte natürlich verhindert werden. Ich schlage vor, den unter train angesiedelte timetable traintimetable zu nennen und die obengenannte Attribute type, source und date unter traintimetable ansiedelt. D.h. dass es pro Zugnummer einen Train geben wuerde mit einem bis mehreren traintimatables.

Fuer die Verwendung der neuen Attribute mache ich folgenden Vorschlag:

type: planned, actual oder calculated (je nachdem ob die vorliegenden Daten Sollvorgaben, Istdaten oder Resultate von Berechnungen sind)

source: Herkunft der Daten. Dies soll ein freier String sein welcher das System der Datenherkunft beschreibt (z.B. FBS, Viriato, OpenTrack, SBB-KVZ, SBB-SURF, DB-RUT, ...). Beim Import mehrerer Fahrplänen pro Zug (z.B. Solldaten aus FBS und berechnete Ist-Daten aus OpenTrack) könnte der User dann auswählen, welche Daten er laden will, wobei der source-String zusammen mit dem type und dem date diese Datenbestände unterscheiden würde.

Nochmals zur ID bei den entry-Elementen. Die heute verwendete ID bezeichnet einen Ort der Infrastruktur, für welchen der oder die angegebenen Zeitwerte gelten. Das kann ein Signal, eine Betriebsstelle etc. sein. Aber es handelt sich immer um eine identifizierbare Position (Location) irgendwo im Gleisnetz. Daher schlage ich nochmals

vor, diesen Sachverhalt im Attributnamen zu verankern. locID oder locationID heisst, dass es in der Infrastrukturbeschreibung ein Element geben muss (z.B. Station, Signal), welches diese locationID ebenfalls als Attribut trägt, d.h. die Zeit kann dann einem realen Objekt der Anlage zugeordnet werden.

Zu begin, end, pass, stop, arrival und departure. Ich bin immer noch fuer die Teilung der beiden Informationen Zeit und Typ des Eintrages, wobei auch die Zeiten optional sein muessen, d.h. wir muessen keine Ankunftszeit definieren, wenn dies nicht nötig ist. Es muss z.B. möglich sein für einen Stationshalt nur die minimale Haltezeit anzugeben ohne einen effektiven Zeitwert. Ich denke nicht, dass die Importfunktion damit komplexer wird. Man hat dann zwei verschiedene Datentypen zu handeln, einen Typ und eine Zeit. Fehlt eine Angabe, so muss man mit geschickten Defaultwerten arbeiten. Typischerweise muss man eben dann für das Feststellen einer Durchfahrt nicht arrival und departure auf Gleichheit prüfen sondern das Attribut type='pass' sagt dies explizit. Fehlt dann z.B. sowohl eine stop wie eine pass Information kann immer noch geschaut werden, ob sowohl arrival wie departure definiert sind, um diese dann zu vergleichen.

Zu den verwendbaren Stationsgleisen (trackID): Es wäre halt schön, wenn man in den Planungsdaten angeben könnte, welche Gleise ein Zug befahren darf (1. Priorität, 2. Priorität, ...) und dann in den Istdaten sehen könnte, welches Gleis der Zug effektiv befahren hat. Würde man einen RailML-Timetable z.B. innerhalb eines Fahrplanentwurfstools verwenden könnte das Tool selbständig prüfen, ob für die gewünschte Solllage des Zuges noch eines der angegebenen Stationsgleise verfügbar ist.

Ich hoffe, dass Ihr diese Vorschläge analysiert und wir für RailML eine gute Lösung bzw. Integration der beschriebenen Sachverhalte finden.

Grüsse aus Zürich

Dani Hürlimann

---