

RailML – Einheitliche Datenschnittstellen für Eisenbahnen

D. Hürlimann^{a,1} und V. P. Krauß^{b,2}

^a ETH Zürich, Institut für Verkehrsplanung und Transportsysteme
ETH Hönggerberg, CH-8093 Zürich, Schweiz

^b Fraunhofer-Institut für Verkehrs- und Infrastruktursysteme
Zeunerstr. 38, D-01069 Dresden, Deutschland

Abstract / Kurzfassung: Durch den fortschreitenden Einsatz von Rechnertechnik in den letzten Jahren existiert mittlerweile eine unüberschaubare Anzahl von Applikationen im Bahnbereich. Diese kommen nur in den seltensten Fällen ohne einen Datentransfer mit anderen Anwendungen aus. Ein Konsortium, anfangs bestehend aus dem Fraunhofer-Institut für Verkehrs- und Infrastruktursysteme (IVI) Dresden und dem Institut für Verkehrsplanung und Transportsysteme (IVT) an der Eidgenössischen Technischen Hochschule (ETH) Zürich, legte den Grundstein für die Entwicklung der RailML-Beschreibungssprache, die dieser Forderung nach Vereinheitlichung auf der Basis von XML (*eXtensible Markup Language*) nachkommt. Auf diese Art können beliebig viele Programme von den zur Verfügung gestellten Daten Gebrauch machen; sogar dann, wenn deren interne Daten-Strukturen nicht völlig identisch sind. Der Vortrag beschreibt das Vorgehen zu europaweiten Koordinierung dieser Schnittstellen, das Datenmodell sowie den praktischen Nutzen und die dazu bestehenden Anwendungen im heutigen Bahnumfeld.

Key Words / Schlagworte: Datenaustausch; Eisenbahn; RailML; Schnittstelle; Softwareentwicklung; XML

¹E-mail: huerlimann@ivt.baug.ethz.ch, URL: <http://www.ivt.baug.ethz.ch/oev.html>

²E-mail: krauss@ivi.fhg.de, URL: <http://sifa.ivi.fhg.de>

1 Einleitung

Aus verschiedenen Forschungsarbeiten im Bereich der Bahninformatik sind in den letzten Jahren Werkzeuge für die Planung und den Betrieb von Eisenbahnsystemen entstanden. Jedes dieser Verfahren basiert auf mehr oder weniger ähnlichen Datenbeständen, in welchen Angaben über Fahrpläne, eingesetztes Rollmaterial und über zur Verfügung stehende Bahninfrastruktur abgelegt sind. Durch den Einsatz proprietärer Dateiformate sind diese Daten typischerweise zwischen den verschiedenen Anwendungen nicht austauschbar. Dieser Mangel wurde von einigen obengenannten Arbeiten identifiziert (z.B. [3,4]), wobei vorgeschlagen wurde, diesen Missstand durch die Entwicklung offener Dateiformate zu beheben.

Ein Konsortium, anfangs bestehend aus dem Fraunhofer-Institut für Verkehrs- und Infrastruktursysteme (IVI) Dresden und dem Institut für Verkehrsplanung und Transportsysteme (IVT) der Eidgenössischen Technischen Hochschule (ETH) Zürich nahm sich diesem Thema an und initiierte ein gemeinsames Forschungsprojekt zum Thema *Interoperabilität von Bahndaten*.

Dieses Projekt wird anhand dieses Aufsatzes vorgestellt, wobei zuerst die Lösungsidee bzw. der Lösungsansatz präsentiert wird, bevor auf die eigentliche Struktur des entwickelten Datenformates eingegangen wird. Weiter wird die Art und Weise der Entwicklung vorgestellt und aufgezeigt, in welcher Form zukünftige Anwendungen aus dem Eisenbahnwesen das neue Format einsetzen können.

2 Lösungsidee und Lösungsansatz

Beim Blick über das Gebiet des Eisenbahnwesens hinaus wurde schnell klar, dass das Thema "offene und universell verwendbare Datenformate" in verschiedensten Anwendungsbereichen aktuell ist. Die Analyse zeigte, dass für verschiedene Anwendungsgebiete solche Formate erfolgreich auf der Basis der Extensible Markup Language (XML) entwickelt wurden (z.B. MathML für mathematische Ausdrücke oder GML für geografische Daten).

Als Sammlung von Regeln zur Schaffung von sog. Auszeichnungssprachen (engl. Markup Language) funktioniert XML als Metasprache zur Definition neuer Sprachen. Der grosse Vorteil von XML-basierten Dokumenten ist, dass sowohl der Inhalt wie auch die Struktur darin beschrieben wird. Für den Transport und die Datenhaltung eisenbahnrelevanter Daten bot sich deshalb XML in idealer Weise an.

Jede dieser Sprachen benützt für deren Dokumente ein sehr einfaches und flexibles ASCII-Textformat. Die Dokumente sind jeweils hierarchisch, in Form eines Baumes aufgebaut, d.h. jedes Dokument hat ein eindeutiges Wurzelement, von welchem aus die Navigation durch die Dokumentstruktur begonnen werden kann.

Der effektive Datenaustausch zwischen zwei Applikationen benötigt in jedem Programm je eine Funktion für das Generieren von XML-Daten und eine Funktion für das Einlesen und die Interpretation von XML-Dateien. Durch die grosse Verbreitung von XML und die Standardisierung durch das World Wide Web-Konsortium existieren für die meisten Entwicklungsplattformen vordefinierte Bibliotheken und Funktionen für die Verarbeitung von XML-basierten Datenbeständen, was die Entwicklungszeit für die jeweilige Import- und Exportfunktionalität massiv reduziert.

3 Struktur

Mit der Wahl von XML als Grundlage des zu entwickelnden Formates RailML (Railway Markup Language) sind gewisse Vorgaben bezüglich der Dokumentstruktur gegeben. Das RailML-Dokument hat ein eindeutiges Wurzelement, welches im vorliegenden Fall `<railml>` heisst. Davon abgeleitet werden dann die eigentlichen Teilschemen, welche die eisenbahnrelevanten Daten enthalten.

Die Abbildung 1 zeigt die Grundstruktur der RailML bzw. die Modularisierung auf oberster Ebene.

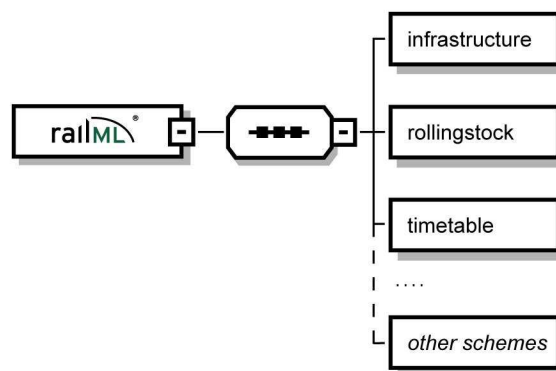


Abbildung 1: Struktur von RailML

In einem RailML-Dokument können in beliebiger Kombination Daten der verfügbaren Eisenbahninfrastruktur (Teilschema *infrastructure*), von eingesetztem Rollmaterial (Teilschema *rollingstock*) sowie Informationen zum Fahrplan (Teilschema *timetable*) verwaltet werden. Durch diesen flexiblen Ansatz obliegt es jeder einzelnen RailML-kompatiblen Applikation, welche Teile des Gesamtschemas verwendet werden sollen.

Die Daten in XML-Dokumenten werden mittels Elementen und Attributen organisiert und verwaltet. Der Anfang eines Elements wird mit einem sog. Starttag (z.B. `<railml>`) gekennzeichnet, das Ende wird mit dem entsprechenden Endtag (z.B. `</railml>`) versehen. Ein Ele-

ment selbst kann Attribute zur näheren Beschreibung tragen und wiederum weitere Elemente enthalten.

Die Abbildung 2 zeigt ein Beispiel eines RailML-Dokuments, welches Fahrplandaten enthält. Unter dem Element <timetable> sind die einzelnen Züge im Element <train> aufgelistet. Als wichtiges Attribut eines Zuges ist die Zugnummer (trainID) zwingend nötig, andere Attribute sind optional.

Jeder Zug enthält beliebig viele Fahrplaneinträge <entry>, welche unter dem Element <timetableentries> geordnet aufgelistet sind. Die einzelnen Fahrplaneinträge enthalten Attribute wie Stationsabkürzung (posID), Ankunfts- und Abfahrtszeit (arrival, departure) oder die minimale Haltezeit (minStopTime).

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with OpenTrack (http://www.opentrack.ch) -->
<railml xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="timetable.xsd">
  <timetable version="0.95" scheduleformat="hh:mm:ss" periodformat="s">
    <train trainID="RX 100.2" type="planned" source="opentrack">
      <timetableentries>
        <entry posID="ZUE" departure="06:08:00" type="begin">
        </entry>
        <entry posID="ZWIP" departure="06:10:30" type="pass">
        </entry>
        <entry posID="ZOER" arrival="06:16:00" departure="06:17:00"
          minStopTime="60" type="stop">
        </entry>
        <entry posID="WS" departure="06:21:00" type="pass">
        </entry>
        <entry posID="DUE" departure="06:23:00" type="pass">
        </entry>
        <entry posID="SCWE" departure="06:27:00" type="pass">
        </entry>
        <entry posID="NAEN" departure="06:29:00" type="pass">
        </entry>
        <entry posID="UST" arrival="06:34:30" type="stop">
        </entry>
      </timetableentries>
    </train>
  </timetable>
</railml>
```

Abbildung 2: Beispiel eines RailML-Dokuments

4 Entwicklung

Die Entwicklung des RailML-Formats wird vor einer Gruppe von Softwareentwicklern und interessierten Benutzern von Bahninformatikwerkzeugen durchgeführt, die sich unter dem Namen *RailML.org* zusammengefunden haben. Eine Koordination in technischer Hinsicht und das Bereitstellen der heute üblichen Medien (Webseite, Diskussionsplattform) wird seit Beginn durch das Fraunhofer-Institut für Verkehrs- und Infrastruktursysteme in Dresden übernommen. Für die einzelnen Schemen (Infrastruktur, Fahrplan und Rollmaterial) gibt es fachliche Teilkoordinatoren, die die Diskussion moderieren und die neuen Versionen der Schemen zur Veröffentlichung vorbereiten (analog den RFC des Internets).

RailML ist bewußt als lockere und offene Plattform für Softwarehersteller, Bahnen, Verbände und Behörden sowie Universitäten und Softwarehersteller ausgelegt. Standards entstehen im Rahmen von fachlichen Diskussionen und vor allem durch die Anwendung, die jedem Beteiligten offensteht.

Bereits heute haben verschiedene Bahninformatikwerkzeuge anwendungsreife Schnittstellen zum RailML-Format entwickelt. So können z.B. Fahrplandaten zwischen den Anwendungen *FBS*, *Viriato* und *OpenTrack* ausgetauscht werden. Wie die Abbildung 3 zeigt, lesen und schreiben diese Anwendungen XML-basierte RailML-Dateien.

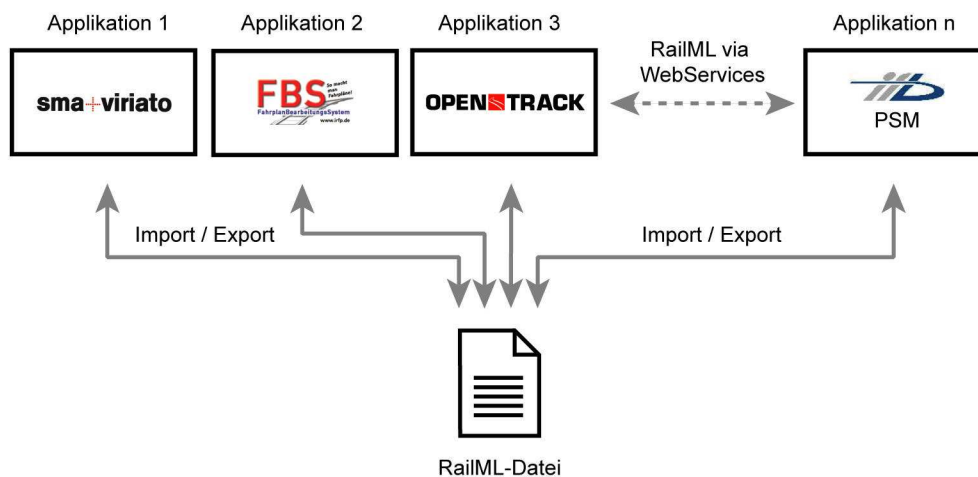


Abbildung 3: Datenaustausch mit RailML

Weitere Applikationsschnittstellen werden in naher Zukunft folgen und den offenen Standard RailML nutzen.

Es existieren jedoch auch bereits Lösungsansätze, welche einen Schritt weiter gehen und die RailML-Daten zwischen zwei Anwendungen direkt mittels Interprozesskommunikation (z.B. WebServices via TCP/IP) ausgetauscht, d.h ohne den Umweg einer Datei. Damit ist eine Tür zu einem noch grösseren Potential von zukünftigen RailML-Anwendungen aufgestossen.

Die im RailML-Konsortium vertretenen Hochschulen sind bestrebt, die RailML-Entwicklung auch in aktuelle Forschungsprojekte und Arbeiten von Studierenden einfließen zu lassen. In diesem Rahmen wurden so in letzter Zeit an der TU Dresden [1] und an der ETH Zürich [2] entsprechende Arbeiten durchgeführt.

5 Zusammenfassung und Ausblick

Anlässlich der im Anschluss zu den Verkehrswissenschaftlichen Tagen in Dresden stattfindenden RailML-Entwicklerkonferenz wird eine Verabschiedung der Version 1.0 des Fahrplan- und des Infrastruktur-Schemas erwartet. Damit wird ein stabiler und gut dokumentierter Stand des RailML-Formates erreicht sein. Somit steht der Implementierung von RailML-Schnittstellen von weiteren Informatikanwendungen aus dem Bereich des Eisenbahnwesens nichts mehr im Wege.

RailML wird demnächst auch neue, direkte Schnittstellen zwischen Fahrplanung und Sicherungstechnik vereinfachen. So werden z.B. in einem beginnenden Projekt zwischen iRFP Dresden (Deutschland) und der FH Wels (Österreich) Daten für ein neuentwickeltes Streckensicherungssystem der Stern&Hafferl Verkehrsgesellschaft mbH (Gmunden, Österreich) ausgetauscht.

Ebenso hat sich ein Entwicklerteam unter der Leitung von Siemens Niederlande entschlossen, für das zu implementierende *Performance Simulation Model* (Bonus/Malus-System) der neuen Hochgeschwindigkeitsstrecke *HSL Zuid* die darin benötigten Fahrplandaten im RailML-Format auszutauschen.

Applikationen, die dem RailML-Standard entsprechen, sollen zukünftig das untenstehende RailML-Logo tragen. Damit werden Kunden und zukünftige Nutzer über die einfache und offene Möglichkeit, ihre Daten auszutauschen, informiert.



Abbildung 4: RailML Logo

Literatur

- [1] Fries, N., 2003, Modellierung und Abbildung einer Eisenbahn-Infrastruktur auf ein Schema der XML-basierten Schnittstelle RailML (Diplomarbeit, TU Dresden, Professur für Verkehrssicherungstechnik, Dresden).
- [2] Hengartner, M., 2003, Grafikeditor für RailML-basierte Infrastrukturdaten (Diplomarbeit, ETH Zürich, Institut für Verkehrsplanung und Transportsysteme, Zürich).
- [3] Hürlimann, D., 2002, Objektorientierte Modellierung von Infrastrukturelementen und Betriebsvorgängen im Eisenbahnwesen, 2002, Institut für Verkehrsplanung und Transportsysteme, ETH Zürich, Schriftenreihe des IVT, Nr. 125.
- [4] Montigel, M., 2002, Time-triggered exchange of train route data between train control systems, in *Computers in Railways*, edited by J. Allan, R. J. Hill, C. A. Brebbia, G. Sciutto & S. Sone (WIT Press, Southampton), pp. 33–41.