

Anlass und Zielstellung

Für Planungsprozesse der Eisenbahnen existieren mittlerweile vielfältige und weitreichende Computerprogramme. Diese kommen nur in den seltensten Fällen ohne Daten aus anderen Anwendungen aus. Für diesen Austausch werden Schnittstellen benötigt, die die Aufgabe der Datenformatkonvertierung übernehmen. Historisch bedingt sind heute meist eigene Schnittstellen für den Datenaustausch zwischen zwei festgelegten Programmen im Einsatz. Diese können aufgrund ihrer inneren Struktur nicht für dritte Programme verwendet werden. Wird evtl. darüber hinaus noch ein viertes oder fünftes Programm hinzugezogen, ergibt sich für den Datenaustausch schnell ein unüberschaubarer Aufwand. Dieser wird zusätzlich noch dadurch verstärkt, dass sich mit jeder Programmversion die Datenstruktur ändern kann. Folglich wäre die jeweilige Schnittstelle funktionsunfähig. Um dies zu vermeiden sind aufwändige Wartungs- und Pflegemaßnahmen aller Seiten notwendig, die mit einem beträchtlichen Kosten- und Zeitbedarf einhergehen.

An diesen Standard knüpft die railML®-Schnittstelle mit ihrer Datenstruktur an. Das datenerzeugende Programm erstellt zunächst eine validierte XML-Datei nach dem betreffenden railML®-Schema. Anschließend wird diese Datei veröffentlicht oder direkt an das jeweilige Empfängerprogramm übergeben, das daraufhin mit einem Parser die empfangene Datei einliest und die benötigten Daten übernimmt. Auf diese Art können auch beliebig viele Programme von den zur Verfügung gestellten Daten Gebrauch machen; sogar dann, wenn deren interne Datenstrukturen nicht völlig identisch sind.

Struktur

Der XML-Syntax gemäß wird jedes Dokument mit einem Schlüsselbegriff (*tag*) eingeleitet, der in diesem Fall „railml“ heißt. Daraus leiten sich die einzelnen Modelle ab, die die Daten thematisch strukturieren. Bis heute stehen den Anwendern drei unterschiedliche Schemen zur Verfügung, die objektbezogen voneinander abgegrenzt sind und bereits einen Großteil der Eisenbahnwelt abdecken. Mit Hilfe von Referenzierungsschlüsseln kann von einem Schema auf ein anderes verwiesen werden, so dass redundante Angaben in verschiedenen Strukturen, die zu Mehraufwand oder Fehlinformationen führen, vermieden werden.

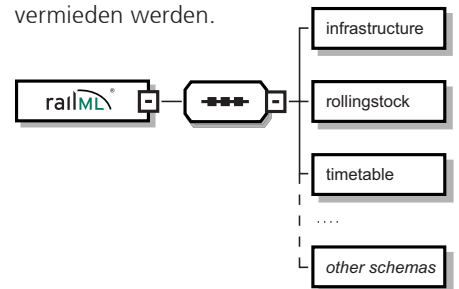


Bild: railML.org-Schemen

Lösungsansatz

Das Fraunhofer-IVI Dresden und die ETH Zürich legten den Grundstein für die Entwicklung der railML®-Schnittstelle, die dieser Forderung nach Vereinheitlichung auf der Basis von XML (eXtensible Markup Language) nachkommt. Die XML-Sprache wurde durch das World Wide Web-Konsortium konzipiert. Sie hat sich als Industriestandard etabliert und wird bereits von einer großen Anzahl diverser Anwendungen in anderen Industriebereichen unterstützt.

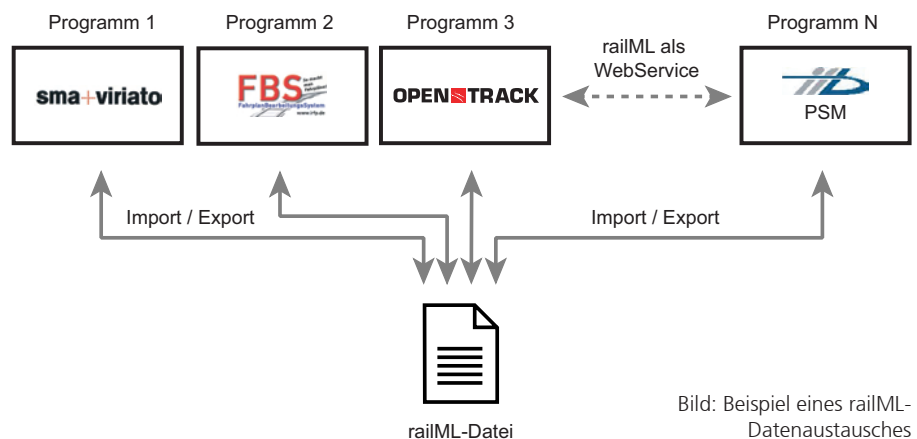


Bild: Beispiel eines railML-Datenaustausches



Kontakt:

DI Vasco Paul Kolmorgen
Dr. Daniel Hürlimann

Zeunerstraße 1
01069 Dresden
Deutschland

Tel.: +49 351 466 76 939
eMail: coordination@railml.org
Internet: www.railml.org

Stand: Juni 2014

Entwicklung

Das Projekt railML® befindet sich in einem ständigen Entwicklungsprozess, um den unterschiedlichen Anwendungen gerecht zu werden. Als OpenSource-Projekt wird es von den beteiligten Projektpartnern mitgetragen und weiterentwickelt, zudem fortdauernd erweitert. Eine Beteiligung am Entwicklungsprozess aus Wissenschaft und Praxis über die Projektseite www.railml.org sowie eine Implemen-

tierung von Schnittstellen nach railML®-Schemen sind jederzeit möglich und erwünscht. Programme, die Schnittstellen nach einem aktuellen railML®-Standard bereitstellen, können diese durch das railML-Konsortium zertifizieren lassen und dürfen dann das railML®-Logo tragen und damit werben sowie die railML®-Funktionalität in der Dokumentation zum Programm aufführen.

Das **INFRASTRUCTURE**-Schema enthält Angaben zur physischen Infrastruktur einer Strecke. Von einem mikroskopischen Modell ausgehend, besitzt eine Strecke beliebig viele Detailelemente, wie z.B. Lage, Gradiente, Schienenform, Weichen oder Signale.

Das **TIMETABLE**-Schema dient zur Übergabe von Fahrplandaten eines Bahnnetzes. Für beliebig viele Züge können wichtige Eigenschaften wie Abfahrts- und Ankunftszeiten, Bespannung, Last oder Verkehrstage übergeben werden.

Das **ROLLING STOCK**-Schema umfasst alle Angaben zu Triebfahrzeugen und Wagenmaterial. Hier können fahrdynamische Spezifikationen, Eigenschaften der Zug-sicherung oder Stromversorgung gespeichert werden. Für den Bereich der Fahrzeugeinsatz- und Umlaufplanung werden fahrzeuggenaue Daten einer Baureihe zur Verfügung gestellt.

Die Versionen 2.2 der entwickelten Schemen *Infrastructure*, *Timetable* und *Rolling stock* sind zum produktiven Einsatz freigegeben. Ein neues Schema für den Bereich *Interlocking* ist in Entwicklung.

Jedes der hier aufgeführten Schemen kann beliebig innerhalb einer Datei mit einem anderen dieser Art kombiniert werden, so dass also nicht für jedes Modell eine gesonderte Datei erstellt werden muss. Die innere Struktur der railML®-Schemen ist von der Bedienung her identisch. Gemäss railML®-Philosophie werden möglichst wenige Angaben als notwendig definiert, um dem Anwender freizustellen, wie viele der zur Verfügung stehenden Eigenschaften er mit seinen Daten belegen will. Damit entstehen sehr flexible Datenstrukturen, die einerseits auf die Bedürfnisse des einzelnen Nutzers zugeschnitten, andererseits aber weitestgehend universell einsetzbar sind, ohne unübersichtlich zu werden.

railML-Entwickler und Anwender (Auswahl)



Bild: Ausgewählte railML.org-Partner

Programme mit railML - Schnittstelle (Auswahl)

Die hier aufgeführten Anwendungen verfügen derzeit über eine railML®-Schnittstelle. Weitere Anwendungen sind bereits im Entwicklungs- & Zertifizierungsprozeß.



Bild: Programme mit railML®-Schnittstelle

railML® Zertifizierung

Programme, welche den railML®-Zertifizierungsprozeß bestanden haben, dürfen das railML-Logo verwenden:

