Subject: Different representations of a train
Posted by Philip Wobst on Tue, 19 Nov 2013 08:49:34 GMT
View Forum Message <> Reply to Message

Hi all,

I would like to start a discussion on this topic and I have found two topics started by Andreas Tanner, IVU with similar content:

train uniqueness constraint II
http://www.railml.org/forum/ro/index.php?group=2&offset= 0&thread=109&id=391

problems with s: uniqueness constraints, scope
http://www.railml.org/forum/ro/index.php?group=2&offset= 0&thread=101&id=360

However, I would like to have a representation of two identical trains that shows the origin of the train which cannot be represented by the processStatus of a train (or trainPart). One of the use cases is a view of a train in a bid/offer process and this would include two, three or even more 'variants' of the same train.

For this reason we need additional information regarding the source of a train/trainPart and the following approaches were found. These approaches do not solve the problem of modeling a bid/offer process in railML although it would be nice to find a solution that would be suitable to support this in the future:

1. Create one file for each source
2. Create one timetablePeriod element for each source
3. Use a special trainGroup to identify the trains for one source

With 2 & 3 the constraint for a train could be extended to include the referenced element - e.g. trainNumber, scope and additionalTrainNumber have to be unique for each trainGroup.

These solutions are based on existing elements and I would be happy to get some feedback regarding your views and possibly suggestions on a meanigful extension that might make it into railML 3.0 to support this.

Kind regards,

Philip Wobst

--
Consultant
Planning and Dispatching Systems
HaCon Ingenieurgesellschaft mbH

Lister Str. 15
30163 Hannover
Germany/Deutschland

Tel. +49 511 33699-498
Fax. +49 511 33699-99
mailto: philip.wobst@hacon.de
http://www.hacon.de

railML Partnerlink:
http://www.railml.org//index.php/entwickler.html?show=112

Registry Court/Amtsgericht: Hannover HRB 1712
Managing Directors/Geschäftsführer:
Michael Frankenberg, Werner Sommerfeld, Peter Talke

Subject: Different representations of a train - meeting 11.11.
Posted by on Tue, 19 Nov 2013 11:59:38 GMT
View Forum Message <> Reply to Message

- English summery below -

Beim Treffen am 11.11. wurde die Frage diskutiert, inwieweit in Zukunft in RailML die Definition eines Zuges klarer durch Struktur ausgedrückt werden könnte und sollte.

Hintergrund ist, dass in der Praxis geläufige Ansichten, was „ein Zug“ ist, bisher nur durch _mehrere_ Instanzen des Elements <train> ausgedrückt werden können. Gemeint sind

Im Allgemeinen
- kann in Betracht kommen, dass alle <train>-Elemente mit der gleichen Zugnummer (dem gleichen Wert des Attributs >trainNumber<), in Zukunft eine neue übergeordnete Struktur bilden,
- ist die neue Struktur – wie auch immer sie geartet sein würde – zunächst äquivalent zum Zusammenfassen von Elementen gleicher Attribute (wie z. B. /trainNumber/) durch „Suchen“ (Parsen) beim Importieren von RailML-Daten.

Insofern geht es hier primär „nur“ um die Frage, ob gewisse Zusammengehörigkeiten zukünftig durch Struktur auszudrücken sind, die bisher durch „gleiche Attribute“ ausgedrückt werden.. Es geht zunächst (noch) nicht um mehr oder anderen Inhalt.

Konkret wurden folgende Ansätze diskutiert:

1)

Verwenden eines Attributs wie /code/, um die Zusammengehörigkeit mehrerer <train>-Elemente auszudrücken. Allerdings war noch nicht unstrittig,
a) ob /code/ den (externen) Primärschlüssel über alle <train>-Elemente enthalten soll (d. h. alle Vorkommen von /code/ müssen eindeutig sein)
b) oder ob /code/ wiederholbar ist, um gruppierend zu wirken.
Im Falle a hätte /code/ in etwa die gleiche Funktion wie /id/ mit dem Unterschied, dass /code/ u. U. außerhalb der RailML-Datei existieren (z. B. angezeigt werden) könnte.
Im Falle b hätte /code/ in etwa die gleiche Funktion wie /trainNumber/ mit dem Unterschied, dass es nicht „inhaltlich vorbelastet“ ist, d. h. beliebige nichtnumerische Zeichengruppen usw. enthalten könnte, ohne inkompatibel zu werden.

Nachteilig in Ansatz 1 erscheint, dass wenig bis keine Struktur in der Bildung von /code/ stecken würde, jedoch offensichtliche Redundanz zu /trainNumber/. Der praktische Nutzen von /code/ zur Schaffung von Klarheit und Eindeutigkeit steht damit in Frage.

2)
Einführung einer übergeordneten Hierarchie-Ebene, d. h. eines übergeordneten gruppierenden XML-Elements über <train>:
a) mit neuem Titel, so dass sich in etwa die Hierarchie (Arbeitstitel)
<trainGroup> <train> <trainPartSequence> <trainPart> ergeben würde,
b) mit Umbenennung des bisherigen <train>-Elements, so dass sich in etwa die Hierarchie (Arbeitstitel)
<train> <trainElement> <trainPartSequence> <trainPart> ergeben würde.

Die Varianten 2a und 2b unterscheiden sich nur begrifflich, nicht inhaltlich.

Die Diskussion ließ m. E. eine Tendenz zu 2a erkennen, da scheinbar wenig Bedenken gegen eine zusätzliche gruppierende Hierarchie-Ebene existieren.

- - -
Dieses Thema wird berührt durch die Frage, wie sich gegenseitig überschreibende Planungszustände abzubilden sind: Beispielsweise ein Zug, der ursprünglich an einem bestimmten Verkehrstag verkehren sollte und zu einem späteren (möglicherweise dem letzten = aktuellen) Planungszustand an diesem Verkehrstag nicht mehr verkehren sollte. Für diesen Fall gibt es bisher im RailML (m. E.) keine vorgesehene Abbildungsweise – bisher ist RailML (<timetable>) nur so konzipiert, dass nur der jeweils ein integerer Planungszustand abgebildet werden kann, typischer Weise der letzte = aktuelle. Dass ein Zug in diesem nicht mehr verkehren soll, der zu einem früheren Planungszustand einmal verkehren sollte, kann man bestenfalls durch den Vergleich zweier RailML-Dateien erkennen.

Man könnte diese Frage der Abbildung verschiedener Planungszustände nun als Teilkomplex der Frage verschiedener Zug-Instanzen ansehen:

Vielleicht typisch wäre
- Planungszustand 1: Rahmenvertrag,
- Planungszustand 2: Jahresfahrplan (überschreibt 1),
- Planungszustand 3: Kurzfristplanung (überschreibt 2),
- Planungszustand 4: Disposition (überschreibt 3).

Was allerdings möglicherweise gegen diese Ansicht spricht, ist der Umstand, dass zeitlich verschiedene Zustände nicht nur Züge betreffen, sondern im Prinzip jedes RailML-Element – auch Infrastruktur, Fahrzeuge usw. Insofern könnte hierfür eine allgemeingültigere Lösung in Frage kommen als für die oben angesprochenen, ausschließlich zug-bezogenen Aspekte wie Alternativfahrpläne.

- - -

English summery

There is a discussion on the question whether to express the definition of a “train” by more structure. In the current RailML, some variations of a train can only be expressed by grouping of attributes (“select all <train> instances with the same attribute…”).

There was a suggestion to introduce one more level of hierarchy above the current <train> such as a <trainGroup>. However, this is “only” a question on “more structure vs. grouping of attributes”, so far not a question on more or other contents.

If you prefer a full English conversation on this topic, please do not hesitate to tell us.

Dirk.