Subject: New working draft of the infrastructure schema available
Posted by Volker Knollmann on Thu, 29 Mar 2007 09:34:35 GMT
View Forum Message <> Reply to Message

(English summary follows after the German text)

Hallo zusammen,

wie auf dem letzten Treffen in Zürich versprochen habe ich die Schemadatei umstrukturiert und so eine Möglichkeit für die einfache Wiederverwendung und Erweiterung von Schemateilen vorbereitet.

Hier erstmal der Link, für die ganz Ungeduldigen:

http://www.railml.org/genesis/infrastructure/drafts/r16/infr astructure_r16.xsd

Die wichtigsten Änderungen:

* Vorbereitung der Aufsplittung in mehrere Dateien:

Die grundsätzliche Auftrennung in

- Typen für physikalische Grundeinheiten (generisch)
- Typen für eisenbahnspezifische Grundeinheiten (generisch)
- Typen für railmlspezifische Daten (generisch)
- Typen für die Infrastruktur (infrastruktur-spezifisch)
- Elemente für die Baumstruktur eines Infrastruktur-XML-Files (infrastruktur-spezifisch)

ist vorbereitet. Die entsprechenden Bereich sind in der o. g. Datei deutlich durch Kommentare abgegrenzt. Die tatsächliche Aufteilung in einzelne Dateien möchte ich möglichst weit an das Ende der Entwicklung der neuen Version schieben, um eine leichtere Editierbarkeit (vor allem mit XML-Spy Home Edition) zu erreichen.

* Trennung von Datentypen und Baumstruktur

Früher waren Datentypen (also im wesentlichen die Attributkombinationen) und die Dokumentenstruktur (also die Definition der Elemente und Kindelemente) miteinander vermischt, was die Schemadatei sehr unübersichtlich machte.

Ich habe nun strikt zwischen Typ- und Element-Definitionen getrennt und die Gliederungstiefe der Schemadatei radikal reduziert. Ein Blick auf die XSD-Datei verdeutlicht sicher sofort, was ich meine.

Auf diese Weise können Datentypen leichter wiederverwendet werden und die Struktur kann unabhängig von den Datentypen verändert werden.

* Aufräumen von Bezeichnern, Datentypen, etc.

Für eine bessere Übersicht, vor allem für die Unterscheidung zwischen Elementnamen und Typennamen, habe ich eine einfache Namenskonvention eingeführt:

- Datentypen beginnen mit "t" (z. B. "tSpeed", "tLength")
- Attributgruppen mit "a" (z. B. "aSignal")
- Elementdefinitionen mit "e" (z. B. "eTracks")

Desweiteren habe ich unbenutzte Typen und Attributgruppen entfernt und die an vielen Stellen sehr freie Typisierung mit "xs:string" oder "xs:integer" strenger gestaltet.

* Einführung einer Typenhierachie

Es gibt nun einen Satz von Typen, von denen viele andere Typen abgeleitet wurden. Beispiele dafür sind "tPlacedElement" oder "tElementWithIdAndName", die die Grundlage für Streckenelemente mit Namen, Identifier, absoluter und relativer Kilometrierung bilden.

Durch Ableitung aller weiteren Streckenelementstypen von diesen Grundtypen erreicht man einheitliche Attributnamen (z. B. einheitlich "id" anstelle von "elemID", "elemId", "id", "trackId", ...) und man schafft einen zentralen Ansatzpunkt für Erweiterungen.

* Einführung von xs:any und xs:anyAttribute

Für die Erweiterbarkeit wurden die Grundtypen mit xs:anyAttribute ausgestattet, so dass sie projekt- oder unternehmensspezifisch mit eigenen Attributen erweitert werden können. Dies gilt dann natürlich auch für alle abgeleiteten Datentypen.

Für die Einführung beliebiger Kindelemente habe ich an folgenden Stellen xs:any neu eingeführt:

- als Kind von <track>
- als Kind von <trackElements>
- als Kind von <ocsElements>

Damit können an zentralen, sinnvollen Stellen eigene Unterbäume

eingehängt werden. Durch das Attribut processContents="strict" wird allerdings verlangt, dass diese Unterbäume durch ein entsprechendes eigenes Schema validierbar sind, sonst schlägt die Gesamtvalidierung fehl.

Durch die Einführung von xs:any wird das Datenmodell, das durch die XSD-Datei beschrieben wird, allerdings nicht deterministisch. XML-Spy merkt dies bei der Überprüfung als Hinweis an. Dabei handelt es sich jedoch nicht um einen Fehler; die XSD-Datei ist valide!

Im Zuge der Einführung von xs:any und xs:anyAttribute habe ich die <generalElements> im <track>-Unterbaum entfernt.

Es gibt jetzt noch "others" in den Infrastructureattributen. Wir sollten klären, ob diese Struktur (die noch dazu sehr umständlich ist), jetzt noch beibehalten werden soll. Hier bitte ich also um Rückmeldung aus der Community. Zur Kompatibilität habe ich den Zweig zunächst beibehalten.

Ich habe versucht, die alte Dokumentstruktur einer Infrastruktur-XML-Datei auch im neu organisierten Schema exakt abzubilden (Ausnahme: Löschung von <generalElements>). Aber trotz aller Sorgfalt können mir natürlich Fehler unterlaufen sein. Daher bitte ich alle railML-Nutzer, den Schemaentwurf einer kritischen Prüfung zu unterziehen und Fehler bzw. Verbesserungsvorschläge zu posten.

Die Koordinatoren von TT und RS bitte ich, sich vor allem die "generischen" Teile der Schemadatei anzuschauen und die Wiederverwendung von Typen in ihren eigenen Schemen zu prüfen. Oder, noch besser, weitere generische Typen zum vorliegenden Entwurf beizutragen, damit diese später ausgelagert und aus RS, TT und IS referenziert werden können.

Alle bisher vorgenommenen Änderungen am IS-Schema sind auch in einer Logdatei dokumentiert:

http://www.railml.org/genesis/infrastructure/drafts/r16/chan gelog.txt

Die Daten unterliegen hier einem Versionierungssystem (Subversion), durch das sich die Änderungen komfortabel nachverfolgen lassen. Bis zum "richtigen" Release V1.2 werde ich daher der Eindeutigkeit halber immer unsere interne Revisionsnummer (aktuel: r16) an den Dateinamen anhängen.

Neben der Arbeit an der XSD-Datei selber habe ich auch noch ein kleines Quick-And-Dirty-Skript geschrieben, dass als Grundlage für eine spätere Dokumentation (z. B. im Wiki) alle Attribute samt Datentyp und Definitionsort aus der Schemadatei extrahiert. Die Ausgabe des Skriptes sieht so aus:

http://www.railml.org/genesis/infrastructure/drafts/r16/allA ttributes.txt

Wer sich für das Skript an sich interessiert, kann es sich hier herunterladen (der direkte Aufruf des Links für zu einer Fehlermeldung des Servers, daher bitte die Datei direkt speichern):

http://www.railml.org/genesis/infrastructure/drafts/r16/extr actAttributes.py

So, ich gratuliere allen, die diesen Roman bis hierher gelesen haben und hoffe auf zahlreiche Rückmeldungen, die dann in den nächsten Vorentwurf des 1.2er-Schemas einfließen können!

Viele Grüße aus Braunschweig,
Volker Knollmann

--
Dipl.-Ing. Volker Knollmann
Coordinator RailML-Infrastructure
Phone: +49 (0) 531 295-3461, Fax: -3402

German Aerospace Center (DLR)
Institute of Transportation Systems
Lilienthalplatz 7, D-38108 Braunschweig

********** English Summary **********

As promised in the last meeting in Zurich, I prepared a new draft for a future V1.2-release of the infrastructure-schema. Here is the link to the XSD-file:

http://www.railml.org/genesis/infrastructure/drafts/r16/infr astructure_r16.xsd

The most important changes:

* Prepared a splitting into multiple files:

The single file may later be splitted easily into

- Types for basic physical units (generic)
- Types for basic railways units (generic)
- Types for railml-specific data (generic)
- Types for infrastructure data (IS-specific)
- Elements for the document-tree of a IS-XML-file (IS-specific)

The final splitting should be postponed as much as possible to the end of the development process, because the single file allows easier editing with XML-Spy Home Edition.

* Separation between types and document structure

The definitions for the data types (attributes) and the document structure (elements) were mixed in former versions. This has been separated in order to increase the readibility of the XSD-file and to ease the re-use of types.

* General clean up of names and types

The names of types and the lose typing of many elements has been cleaned up and improved. Unused types and attribute groups have been deleted.

* Introduction of a hierachy of types

Many types are now derived from a set of few basic types. Thus, common attribute names are harmonized the extendibility is improved.

* Introduction of xs:any and xs:anyAttribute

Many elements inherit xs:anyAttribute from a more generic type and are therefore easily extendable with new attributes.

Furthermore, thanks for xs:any arbitrary elements or subtrees can now be inserted at the following positions:

- as child of <track>
- as child of <trackElements>
- as child of <ocsElements>

At the same time, the former extension point <generalElements> has

been removed from the schema.

I asked everyone to check the new schema and to report any errors, incompatibilities or suggestions!

All changes made so far to the IS-Schema are listed in the following changelog:

http://www.railml.org/genesis/infrastructure/drafts/r16/chan gelog.txt

Besides modifying the XSD-File, I created a small script to extraxt all attributes, their types and their place of definition from the schema. The resulting list can be the basis for a detailed documentation, e. g. in the wiki.

The list itself can be found here:

http://www.railml.org/genesis/infrastructure/drafts/r16/allA ttributes.txt

The script can be downloaded from this position (pls. use "Save Link as..."):

http://www.railml.org/genesis/infrastructure/drafts/r16/extr actAttributes.py

I hope for many comments and suggestions regarding the draft version of the schema!

Best regards from Braunschweig,
Volker Knollmann

--
Dipl.-Ing. Volker Knollmann
Coordinator RailML-Infrastructure
Phone: +49 (0) 531 295-3461, Fax: -3402

German Aerospace Center (DLR)
Institute of Transportation Systems
Lilienthalplatz 7, D-38108 Braunschweig

Subject: Re: New working draft of the infrastructure schema available
Posted by Joachim Rubröder railML on Wed, 18 Apr 2007 19:50:04 GMT
View Forum Message <> Reply to Message

Hallo Volker,

ich habe deine heftigen Umstrukturierungen auch mal für das
timetable-Schema durchgeführt (mit multiplen XSD) und für durchaus
praktikabel befunden. Siehe timetable V1.05 auf der Entwicklerseite!

Einen neuen generischen Typ könnte ich beisteuern - "tElementDescription"
ist ein bei mir häufig auftauchendes Freitextfeld zur Beschreibung.

An simpleTypes nutze ich fleissig Distanzen (tLength) und
Geschwindigkeiten (tSpeed) . Wobei ich gerne auch Geschwindigkeit = 0
zulassen möchte. Zusätzlich hätte ich noch einen alignmentType
(Ausrichtung des Zuges) als enumeration (head/center/rear) anzubieten,
weiss aber nicht ob der zu den railwayUnits passt und von allgemeinem
Interesse ist.

Deine "xs:any" haben keinen namespace, ich habe bei meinen den namespace
#other verwendet. Vielleicht weiss einer wie es am empfehlenswertesten
ist, dann sollten wir es einheitlich machen.

Gruss,
Joachim

Subject: Re: New working draft of the infrastructure schema available
Posted by Volker Knollmann on Wed, 13 Jun 2007 11:55:51 GMT
View Forum Message <> Reply to Message

On 18.04.2007 21:50, coord@timetable.railml.org wrote:
> Einen neuen generischen Typ könnte ich beisteuern - "tElementDescription"
> ist ein bei mir häufig auftauchendes Freitextfeld zur Beschreibung.

Habe ich in einen neuen Schema-Zwischenstand übernommen.

> Wobei ich gerne auch Geschwindigkeit = 0
> zulassen möchte.

Fixed. v=0 ist nun ein erlaubter Wert.

> Zusätzlich hätte ich noch einen alignmentType
> (Ausrichtung des Zuges) als enumeration (head/center/rear) anzubieten,
> weiss aber nicht ob der zu den railwayUnits passt und von allgemeinem
> Interesse ist.

Habe ich den railwayUnits hinzugefügt (Name: tTrainAlignment)

> Deine "xs:any" haben keinen namespace, ich habe bei meinen den namespace
> #other verwendet. Vielleicht weiss einer wie es am empfehlenswertesten
> ist, dann sollten wir es einheitlich machen.

Zum xs:any siehe nächste Nachricht, die auch die URL zum neuen Schemazwischenstand enthält.

Leider konnte ich Deine XSD-Version nicht herunterladen (Datei nicht gefunden), so dass die neuen Typen einfach anhand Deiner Beschreibungen selber "konstruiert" habe.

Viele Grüße,
Volker

--
Dipl.-Ing. Volker Knollmann
Coordinator RailML-Infrastructure
Phone: +49 (0) 531 295-3461, Fax: -3402

German Aerospace Center (DLR)
Institute of Transportation Systems
Lilienthalplatz 7, D-38108 Braunschweig

Subject: Re: New working draft of the infrastructure schema available
Posted by Volker Knollmann on Wed, 13 Jun 2007 12:09:23 GMT
View Forum Message <> Reply to Message

For those of you in a hurry:

http://www.railml.org/genesis/infrastructure/drafts/r19/infr astructure_r19.xsd
http://www.railml.org/genesis/infrastructure/drafts/r19/chan gelog.txt

Dear railML-Community,

I just prepared a new issue of the IS-Schema based on the feedback given by the other coordinators and by Heidrun Jost, who reported an error.

The error was related to the new xs:any-element, which has been introduced to allow easy schema extension.

Unfortunately, I used xs:any as follows:

<xs:element name="e1"/>
<xs:element name="e2"/>
<xs:any/>

This kind of usage is wrong, because the content model becomes ambiguous, because a file like

<e1/> <e2/> <e2/>

can be valid (the second e2 is part of xs:any) or invalid (the second e2 is an unallowed repetition of the first e2).

To avoid this ambiguity, I moved xs:any to a separate subtree like this:

```
<xs:element name="e1">
<xs:element name="e2">
<xs:element name="otherElements">
  <xs:complexType>
    <xs:sequence minOccurs="0" maxOccurs="unbounded">
      <xs:any/>
    </xs:sequence>
  </xs:complexType>
```

So a file with customer specific extensions would look like this now:

```
<e1/>
<e2/>
<otherElements>
  <thisIsMyOwnElementAndICanDoWhateverIWant/>
</otherElements>
```

Due to this change, the customer elements are not at the same level as the "genuine" elements of railML, but I think this is more of a aesthetic issue than a real problem.

For all other changes, please take a look at the changelog file (see link above).

I'm looking forward to your comments!

Best regards,
Volker Knollmann

--
Dipl.-Ing. Volker Knollmann
Coordinator RailML-Infrastructure
Phone: +49 (0) 531 295-3461, Fax: -3402

German Aerospace Center (DLR)
Institute of Transportation Systems
Lilienthalplatz 7, D-38108 Braunschweig