Subject: Extensions to railML for passenger information at stations
Posted by on Wed, 24 Apr 2019 22:04:05 GMT
View Forum Message <> Reply to Message

[English text below]

Hallo zusammen,

im Rahmen eines Projektes mit der Rhätischen Bahn, dass sich mit dem Upgrade der IT-Infrastruktur von railML 1.0 auf railML 2.X befasst, sind eine Reihe von Anforderungen aufgekommen, die derzeit noch nicht mit railML erfüllt werden können. Dabei geht es im Wesentlichen darum verschiedene Fahrgastinformationen in railML abzubilden um diese standardisiert zwischen den Systemen austauschen zu können. In den verlinkten PDFs sind die Anforderungen erklärt und mögliche Lösungswege skizziert. Um zu entscheiden, wie eine optimale Abbildung in railML Aussehen sollte, wäre es super, wenn ihr euch das Dokument unter http://forum.railML.org/userfiles/2019-04-24_psi_fahrgastinf ormation-bahnhof-railml25.pdf (Seiten 1-11) mal anschauen könntet und Feedback geben könntet. Haben auch andere Interesse an einer Abbildung dieser Informationen in railML? Welche Aspekte sollten zusätzlich oder anders berücksichtigt werden? Ich freue mich über jedes Feedback.

Hello, everybody,

Within the scope of a project with the Rhaetian Railway dealing with the upgrade of the IT infrastructure from railML 1.0 to railML 2.X, a number of requirements have arisen which cannot yet be met with railML. The main goal is to map different passenger information in railML in order to exchange it between the systems in a standardized way. The linked PDFs explain the requirements and outline possible solutions. In order to decide how an optimal representation in railML should look like, it would be great if you could have a look at the document at http://forum.railML.org/userfiles/2019-04-24_psi_fahrgastinf ormation-bahnhof-railml25.pdf (pages 12-22) and give feedback. Are there others interested in having this information mapped to railML? Which aspects should be considered additionally or differently? I am looking forward to any feedback.

Best regards, Milan
--
Milan Wölke
Dipl.-Inf. (FH)
Technischer Projektleiter

PSI TransCom GmbH
Dircksenstraße 42-44
10178 Berlin
Deutschland

Subject: Re: Extensions to railML for passenger information at stations
Posted by on Fri, 03 May 2019 13:07:10 GMT
View Forum Message <> Reply to Message

[English text below]

Hallo Milan, hallo Forum

ich habe mir Deine Erweiterungs-Vorschlag angesehen und finde ihn grundsätzlich sehr gelungen. Im folgenden einige Anmerkungen:

- Aufzähltypen würde ich generell individuell erweiterbar definieren (mittels "other:anything"), so wie das z.B. bei "processStatus" der Fall ist. Dies betrifft in Deinem Vorschlag z.B. <annotation type="..."> und <trigger event="...">

- Bei <origin> / <destination> würde ich überlegen, auf die ocpRef zu verzichten. Wenn die Ausgangs-/Ziel-Betriebsstelle eines Zuges in den Infrastrukturdaten der railML-Datei enthalten ist, dann sollte es auch möglich sein, den Zuglauf auf diesen Betriebsstellen anzugeben.

- Zur Erweiterung des <blockPart> um einen Aufzähltyp (ShortTurn|LongTurn|ContinueInSameDirection):
Ich denke, diese Informationen sind (in indirekter Form) bereits jetzt abbildbar. In der letzten Telefonkonferenz hatten wir ja bereits darüber gesprochen, dass die Unterscheidung ShortTurn <> LongTurn sich im Umlaufplan auch mit einem eingeschobenen <blockPart> ohne referenzierten Zugteil (mission="shunting" o.ä.) realisieren ließe. Die Information "ContinueInSameDirection" könnte man über einen commercial <train> (Komplett-Fahrzeug-Durchlauf auf anderen betrieblichen Zug) abbilden. Somit könnte man auf die Erweiterung des <blockPart> verzichten. Ich gebe zu, dass diese Informationen bei dieser Lösung im Vergleich zu Deinem Vorschlag sehr verstreut sind. Generell sehe ich den Umlaufplan aber primär als innerbetriebliche Unterlage an, so dass ich aus "struktureller" Sicht dort eher keine Daten der Fahrgastinformation ergänzen würde.

- Ich würde mich Philips Vorschlag anschliessen, die <passengerInfoLanguages> als Unterelement der <annotationRef> abzubilden. Der einzige Einwand dagegen wäre der Fall, dass man die Info-Sprachen eines <trainPart> ohne eine konkrete Annotation angeben möchte.

Soweit von mir. Insgesamt finde ich die Umsetzung dieser Anforderungen sehr gut.

Viele Grüße
Christian Rößiger

Hello Milan, hello Forum

I have looked at your extension proposal and find it basically very useful. In the following some remarks:

- I would generally define enumerated types as individually extensible (using "other:anything"), as is the case with "processStatus", for example. This applies in your suggestion e.g. <annotation type="..."> and <trigger event="...">

- With <origin> / <destination> I would consider to not use the ocpRef. If the origin/destination point of a train is contained in the infrastructure data of the railML file, then it should also be possible to specify the train run on these points.

- Concerning the extension of the <blockPart> by one enumeration type (ShortTurn|LongTurn|ContinueInSameDirection):
I think this information can already be modelled (in indirect form). In the last telephone conference we had already talked about the fact that the distinction ShortTurn <> LongTurn could also be implemented in the rostering with an inserted <blockPart> without a referenced train part (mission="shunting" or similar). The information "ContinueInSameDirection" could be mapped via a commercial <train> (complete vehicle run-through on another operational train). Thus one could do without the extension of the <blockPart>. I admit that this information is very scattered compared to your suggestion. In general, however, I see the circulation plan primarily as an internal document, so that from a "structural" point of view I would rather not supplement any passenger information data there.

- I would agree with Philip's proposal to map the <passengerInfoLanguages> as a sub-element of the <annotationRef>. The only objection to this would be if you wanted to specify the info languages of a <trainPart> without a concrete annotation.

So far from me. Generally I find the implementation of these requirements very good.

Kind regards
Christian Rößiger

Am 25.04.2019 um 00:04 schrieb Milan Wölke:
> [English text below]
>
> Hallo zusammen,
>
> im Rahmen eines Projektes mit der Rhätischen Bahn, dass sich mit dem

> Upgrade der IT-Infrastruktur von railML 1.0 auf railML 2.X befasst, sind
> eine Reihe von Anforderungen aufgekommen, die derzeit noch nicht mit
> railML erfüllt werden können. Dabei geht es im Wesentlichen darum
> verschiedene Fahrgastinformationen in railML abzubilden um diese
> standardisiert zwischen den Systemen austauschen zu können. In den
> verlinkten PDFs sind die Anforderungen erklärt und mögliche Lösungswege
> skizziert. Um zu entscheiden, wie eine optimale Abbildung in railML
> Aussehen sollte, wäre es super, wenn ihr euch das Dokument unter
> http://forum.railML.org/userfiles/2019-04-24_psi_fahrgastinf ormation-bahnhof-railml25.pdf
> (Seiten 1-11) mal anschauen könntet und Feedback geben könntet. Haben
> auch andere Interesse an einer Abbildung dieser Informationen in railML?
> Welche Aspekte sollten zusätzlich oder anders berücksichtigt werden? Ich
> freue mich über jedes Feedback.
>
> Hello, everybody,
>
> Within the scope of a project with the Rhaetian Railway dealing with the
> upgrade of the IT infrastructure from railML 1.0 to railML 2.X, a number
> of requirements have arisen which cannot yet be met with railML. The
> main goal is to map different passenger information in railML in order
> to exchange it between the systems in a standardized way. The linked
> PDFs explain the requirements and outline possible solutions. In order
> to decide how an optimal representation in railML should look like, it
> would be great if you could have a look at the document at
> http://forum.railML.org/userfiles/2019-04-24_psi_fahrgastinf ormation-bahnhof-railml25.pdf
> (pages 12-22) and give feedback. Are there others interested in having
> this information mapped to railML? Which aspects should be considered
> additionally or differently? I am looking forward to any feedback.
>
> Best regards, Milan

--
iRFP e. K. · Institut für Regional- und Fernverkehrsplanung
Hochschulstr. 45, 01069 Dresden
Tel. +49 351 4706819 · Fax. +49 351 4768190 · www.irfp.de
Registergericht: Amtsgericht Dresden, HRA 9347

---

Subject: Re: Extensions to railML for passenger information at stations
Posted by Tobias Bregulla on Wed, 08 May 2019 14:51:54 GMT
View Forum Message <> Reply to Message

[English text below]

Guten Tag,

wir stellen auch eine Vielzahl von Infrastrukturdaten für

Fahrgastinformationssysteme bereit. Daher halten wir diese Erweiterung auch für sinnvoll, würden dazu bereits jetzt folgende (erste) Anmerkungen aus Infrastruktursicht machen:

- Der Ansage-Zeitpunkt (Trigger) wird bei vielen Systemen nicht nur alleine über die Zeit vor einem Ereignis bestimmt, sondern auch über die Entfernung (Weg); weitere Trigger (wie Uhrzeit) sind denkbar.
Von daher sollte bei der Modellierung die Aufzählung des Attributes @event um folgende Trigger erweitert werden:
  - Strecke & Kilometer (z.B. beim Erreichen von km 71,880 ansagen "Wir erreichen Disentis")
  - Betriebsstelle (z.B. beim Erreichen von von Disentis ansagen "10 Minuten Aufenthalt zum Lokwechsel")
  - Geokoordinate und Fangradius
  - ggf. auch reine Uhrzeiten (z.B. Ansage "Ab jetzt Happy Hour im Speisewagen; 3 Bier zum Preis von 2!")

- Die Modellierung der Triggerpunkte sollte vielleicht etwas strukturiert und klarer beschrieben werden (z.B. Unterschied "Arrival" und "ScheduledArrival"? Ist das in der Schweiz nicht immer identisch? ;-)). Da kommen bestimmt noch weitere Trigger dazu im Laufe der Zeit.

- Fehlt dem @periodic nicht noch ein Intervall und eine Beschreibung, was mit dem Rest nach dem letzten Intervall (mathematischer Rest des Bruchs) passieren soll?

- Für die Ansagen sollten Prioritäten vergeben werden können, wenn mehrere Ereignisse gleichzeitig stattfinden. Irgendwie muss das Systeme ja entscheiden können, welcher Ansage dann der Vorrang gegeben wird.

Beste Grüße,
--
Tobias Bregulla
Bahnkonzept Dresden/Germany

Dear all,

we also provide a variety of infrastructure data for passenger information systems. Therefore, we consider this extension to be useful and would like to make the following (first) comments from an infrastructure point of view:

- In many systems, the announcement time (trigger) is not only determined by the time before an event alone, but also by the distance (distance); further triggers (such as time) are conceivable.
For this reason, the enumeration of the @event attribute should be extended by the following triggers during modelling:

- Distance & Mileage (e.g. when reaching km 71,880 announce "We reach Disentis")
- Service station (e.g. when reaching Disentis announce "10 minutes stop at this station to change locomotive")
- Geocoordinate and snap radius
- if necessary also pure times (e.g. announcement "From now on Happy Hour in the dining car; 3 beers for the price of 2!")

- The modelling of the trigger points should perhaps be more structured and clearly described (e.g. difference between "Arrival" and "ScheduledArrival"? Isn't this always the same in Switzerland? ;-)). There will certainly be more triggers in the course of time.

- Doesn't @periodic still lack an interval and a description of what should happen to the rest after the last interval (mathematical remainder of the fraction)?

- It should be possible to assign priorities to the announcements if several events take place simultaneously. Somehow the system must be able to decide which announcement is given priority.

Best regards,
--
Tobias Bregulla
Bahnkonzept Dresden/Germany

---

Subject: Re: Extensions to railML for passenger information at stations
Posted by on Mon, 20 May 2019 23:20:25 GMT
View Forum Message <> Reply to Message

[English text below]

Hallo zusammen,

erstmal vielen Dank für die bereits eingegangenen Hinweise. Ich habe versucht, diese in das Dokument einzuarbeiten:
https://forum.railml.org/userfiles/2019-05-20_psi_fahrgastin formation-bahnhof-railml25.pdf
(Seiten 1-12)

Zu den einzelnen Anregungen:
- Aufzählungstypen sollten erweiterbar sein
Sehe ich auch so. Habe ich im Dokument vervollständigt.
- Verzicht auf ocpRef bei <origin>/<destination>
Tatsächlich habe ich das auf Anregung von HaCon aufgenommen. Ich kann deine Sicht nachvollziehen, habe aber auch kein Problem, wenn die Möglichkeit erhalten bleibt auf einen Ort als Ursprung oder Ziel zu verweisen, der nicht Teil des Laufwegs ist. Gerade wenn man daran denkt,

dass OCPs ganz prinzipiell hierarchisch aufgebaut sein können, kann das Sinn machen.
- Erweiterung von <blockPart>
 Sehe ich mittlerweile auch so. Habe ich im Dokument gestrichen.
- Weitere Ereignisse, die Ansagen auslösen können
 Ich habe die Anregung aufgenommen und zusätzlich zum Fahrtevent (Ankunft, Abfahrt, etc.) auch die logische Position, die physikalische Position und Zeitpunkte aufgenommen.
- Fehlendes Intervall bei periodischen Ansagen
 Das fehlende Intervall wurde ergänzt. Danke für den Hinweis.
- Priorität von Ansagen
 Habe ich auch übernommen.

Hello, everybody,

first of all many thanks for the already received hints. I have tried to include them in the document:
 https://forum.railml.org/userfiles/2019-05-20_psi_fahrgastin formation-bahnhof-railml25.pdf (pages 13-23)

To the individual suggestions:
- Enumeration types should be extendable
 I agree. I have corrected it in the document.
- No ocpRef for <origin>/<destination>
 In fact, I included this at the suggestion of HaCon. I can understand your point of view, but I also have no problem, if the possibility remains to refer to a place as origin or destination, which is not part of the route. Especially if you think of the fact that OCPs can be structured hierarchically in principle, this can make sense.
- Extension of <blockPart>
 I feel the same way now. I deleted it from the document.
- More events that can trigger announcements
 I took up the suggestion and in addition to the trip event (arrival, departure, etc.) also added the logical position, the physical position and time points.
- Missing interval for periodic announcements
 The missing interval was added. Thanks for the hint.
- Priority of announcements
 I adopted that too.

Best regards, Milan
--
Milan Wölke
Dipl.-Inf. (FH)
Technischer Projektleiter

PSI TransCom GmbH

Dircksenstraße 42-44
10178 Berlin
Deutschland
Am 03.05.2019 um 15:07 schrieb Christian Rößiger:
> [English text below]
>
> Hallo Milan, hallo Forum
>
> ich habe mir Deine Erweiterungs-Vorschlag angesehen und finde ihn
> grundsätzlich sehr gelungen. Im folgenden einige Anmerkungen:
>
> - Aufzähltypen würde ich generell individuell erweiterbar definieren
> (mittels "other:anything"), so wie das z.B. bei "processStatus" der Fall
> ist. Dies betrifft in Deinem Vorschlag z.B. <annotation type="..."> und
> <trigger event="...">
>
> - Bei <origin> / <destination> würde ich überlegen, auf die ocpRef zu
> verzichten. Wenn die Ausgangs-/Ziel-Betriebsstelle eines Zuges in den
> Infrastrukturdaten der railML-Datei enthalten ist, dann sollte es auch
> möglich sein, den Zuglauf auf diesen Betriebsstellen anzugeben.
>
> - Zur Erweiterung des <blockPart> um einen Aufzähltyp
> (ShortTurn|LongTurn|ContinueInSameDirection):
> Ich denke, diese Informationen sind (in indirekter Form) bereits jetzt
> abbildbar. In der letzten Telefonkonferenz hatten wir ja bereits darüber
> gesprochen, dass die Unterscheidung ShortTurn <> LongTurn sich im
> Umlaufplan auch mit einem eingeschobenen <blockPart> ohne referenzierten
> Zugteil (mission="shunting" o.ä.) realisieren ließe. Die Information
> "ContinueInSameDirection" könnte man über einen commercial <train>
> (Komplett-Fahrzeug-Durchlauf auf anderen betrieblichen Zug) abbilden.
> Somit könnte man auf die Erweiterung des <blockPart> verzichten. Ich
> gebe zu, dass diese Informationen bei dieser Lösung im Vergleich zu
> Deinem Vorschlag sehr verstreut sind. Generell sehe ich den Umlaufplan
> aber primär als innerbetriebliche Unterlage an, so dass ich aus
> "struktureller" Sicht dort eher keine Daten der Fahrgastinformation
> ergänzen würde.
>
> - Ich würde mich Philips Vorschlag anschliessen, die
> <passengerInfoLanguages> als Unterelement der <annotationRef>
> abzubilden. Der einzige Einwand dagegen wäre der Fall, dass man die
> Info-Sprachen eines <trainPart> ohne eine konkrete Annotation angeben
> möchte.
>
> Soweit von mir. Insgesamt finde ich die Umsetzung dieser Anforderungen
> sehr gut.
>
> Viele Grüße
> Christian Rößiger

>
>
> Hello Milan, hello Forum
>
> I have looked at your extension proposal and find it basically very
> useful. In the following some remarks:
>
> - I would generally define enumerated types as individually extensible
> (using "other:anything"), as is the case with "processStatus", for
> example. This applies in your suggestion e.g. <annotation type="...">
> and <trigger event="...">
>
> - With <origin> / <destination> I would consider to not use the ocpRef.
> If the origin/destination point of a train is contained in the
> infrastructure data of the railML file, then it should also be possible
> to specify the train run on these points.
>
> - Concerning the extension of the <blockPart> by one enumeration type
> (ShortTurn|LongTurn|ContinueInSameDirection):
> I think this information can already be modelled (in indirect form). In
> the last telephone conference we had already talked about the fact that
> the distinction ShortTurn <> LongTurn could also be implemented in the
> rostering with an inserted <blockPart> without a referenced train part
> (mission="shunting" or similar). The information
> "ContinueInSameDirection" could be mapped via a commercial <train>
> (complete vehicle run-through on another operational train). Thus one
> could do without the extension of the <blockPart>. I admit that this
> information is very scattered compared to your suggestion. In general,
> however, I see the circulation plan primarily as an internal document,
> so that from a "structural" point of view I would rather not supplement
> any passenger information data there.
>
> - I would agree with Philip's proposal to map the
> <passengerInfoLanguages> as a sub-element of the <annotationRef>. The
> only objection to this would be if you wanted to specify the info
> languages of a <trainPart> without a concrete annotation.
>
> So far from me. Generally I find the implementation of these
> requirements very good.
>
> Kind regards
> Christian Rößiger

---

Subject: Re: Extensions to railML for passenger information at stations
Posted by Milan Wölke on Mon, 24 Jun 2019 09:46:49 GMT
View Forum Message <> Reply to Message

Ticket for this issue was created at https://trac.railml.org/ticket/358.
Ein Ticket für diesen Vorgang wurde unter https://trac.railml.org/ticket/358 angelegt.

---

Subject: Re: Extensions to railML for passenger information at stations
Posted by Cédric Lavanchy on Wed, 07 Aug 2019 07:04:57 GMT
View Forum Message <> Reply to Message

[English text below]

Hallo Milan,

Allgemein bin ich mit den Ideen einverstanden. Habe gleich meine Inputs zu den Elementen deines Dokuments hier erfasst:

* annotation vs announcement: ich verstehe den Bedarf an 2 Elementen. Ich wurde aber das sehr gut zu dokumentieren, da die Ähnlichkeit der Elemente zur Verwirrung führen kann.

* Für die Erweiterung des trainParts/stopDescriptions mit announcementRef sehe ich den Bedarf an "periodic" announcement auf dem StopDescrption nicht ganz. Hättest du hier ein Beispiel? Ich sehe aber schon, dass man den gleich Type "tAnnoucementRef" verwendet.

* origin/destination: ok für mich.

* passengerInfoLanguages: was gilt wenn keine passengerInfoLanguages auf einem StopDescription definiert ist? Ist es den aus dem TrainPart? Ich würde hier gut dokumentieren, wie es zu interpretieren/Vortrittsregeln ist.

* trainPart -> announcementRefs: wie ist es zu interpretieren wenn zwei announcements mit den gleichen Regeln (periodic/trigger) definiert sind? Müssen sie in Reihenfolge gespielt werden oder eine überschreibt den anderen?

Hello Milan

I generally agree with your ideas. I still do give here some inputs about the elements of your document:

* annotation vs announcement: I understand the need of two elements. I would here write a really good documentation because the similarity of the two elements can lead to confusion.

* For the extension of the trainParts/stopDescription with the annoucementRef: I don't really see the need of the "periodic" announcement on the StopDescrption. Do you have any example here? I still do see the advantage of reusing the type tAnnoucementRef.

* origin/destination: ok for me.

* passengerInfoLanguages: what applies when no passengerInfoLanguages is defined on a StopDescription? Is i the one from the TrainPart? I would write a good documentation about the

way to interpret this and about the priorities.

* trainPart -> announcementRefs: how should it be interpreted when two annoucements have the same "rules" (periodic/trigger)? Are they played one after the other or do one overwrite the second?

Best Regards
Cédric Lavanchy

---

Subject: Re: Extensions to railML for passenger information at stations
Posted by Stefan Hubrig on Mon, 12 Aug 2019 14:18:12 GMT
View Forum Message <> Reply to Message

Hello Milan,

I think these additions are good and useful. No further comments.

Best regard,
Stefan Hubrig

---

Subject: Re: Extensions to railML for passenger information at stations
Posted by Milan Wölke on Wed, 21 Aug 2019 16:57:49 GMT
View Forum Message <> Reply to Message

Hi Cedric, hi Stefan,

zuerst mal vielen Dank, dass ihr euch die Zeit genommen habt, euch den Vorschlag anzuschauen.

Zu den Anmerkungen:

Quote:* annotation vs announcement: ich verstehe den Bedarf an 2 Elementen. Ich wurde aber das sehr gut zu dokumentieren, da die Ähnlichkeit der Elemente zur Verwirrung führen kann.

Ja, die Dokumentation wird sicher nicht einfach, sehe ich auch so. Ich hoffe ich kann auf eure Unterstützung beim Review zählen.

Quote:* Für die Erweiterung des trainParts/stopDescriptions mit announcementRef sehe ich den Bedarf an "periodic" announcement auf dem StopDescrption nicht ganz. Hättest du hier ein Beispiel? Ich sehe aber schon, dass man den gleich Type "tAnnoucementRef" verwendet. Tatsächlich ist das in den Entwurf gekommen, weil ich gerne eine vergleichbare Abbildung für annotations und announcements haben wollte. Da annotationRef bereits an der stopDescription existiert, sollte auch announcementRef dort existieren. Da schon ein entsprechender Type definiert war, hab ich den wiederverwendet. Um aber trotzdem ein Beispiel zu liefern, könnte ich mir vorstellen, dass man periodisch darauf hinweist, in welchen Sektoren die Auslastung des

Zuges geringer ist, um damit die Fahrgastströme zu steuern. Das macht natürlich nur Sinn an einer Online Schnittstelle und ist zugegebenerweise eher weit hergeholt.

Quote:* passengerInfoLanguages: was gilt wenn keine passengerInfoLanguages auf einem StopDescription definiert ist? Ist es den aus dem TrainPart? Ich würde hier gut dokumentieren, wie es zu interpretieren/Vortrittsregeln ist.
Prinzipiell würde ich erwarten, dass wenn keine expliziten Sprachen beim Datenaustausch angegeben wurden, die Systemdefaults des empfangenden Systems gelten. Man könnte aber auch darüber nachdenken, ob das nicht eher für die zweite Art der Abbildung, die ich im Entwurf beschreibe (outputLanguages an annotationRef und announcementRef) spricht. Sollten wir in der Entwicklergruppe noch diskutieren.

Quote:* trainPart -> announcementRefs: wie ist es zu interpretieren wenn zwei announcements mit den gleichen Regeln (periodic/trigger) definiert sind? Müssen sie in Reihenfolge gespielt werden oder eine überschreibt den anderen?
Wenn mehrere announcementRefs existieren mit den gleichen Regeln existieren, müssen diese nacheinander ausgegeben werden. Die Reihenfolge kann dabei über das Attribut @priority gesteuert werden.

---------- ----------

Hi Cedric, hi Stefan,

first of all, thank you so much for taking the time to look at the proposal.

To the comments:

Quote:* annotation vs announcement: I understand the need of two elements. I would here write a really good documentation because the similarity of the two elements can lead to confusion.

Yes, the documentation won't be easy, I see it that way too. I hope I can count on your support during the review.

Quote:* For the extension of the trainParts/stopDescription with the annoucementRef: I don't really see the need of the "periodic" announcement on the StopDescrption. Do you have any example here? I still do see the advantage of reusing the type tAnnoucementRef.
In fact, this came into the draft because I wanted to have a similar representation for annotations and announcements. Since annotationRef already exists at the stopDescription, announcementRef should also exist there. Since a corresponding type was already defined, I reused it. But to give you an example, I could imagine that you would periodically point out in which sectors the load of the train is lower to control the passenger flows. Of course, this only makes sense at an online interface and admittedly is rather far-fetched.

Quote:* passengerInfoLanguages: what applies when no passengerInfoLanguages is defined on a StopDescription? Is i the one from the TrainPart? I would write a good documentation about the way to interpret this and about the priorities.
In principle, I would expect that if no explicit languages were specified during data exchange, the system defaults of the receiving system would apply. One could also consider whether this is

more in favor of the second approach (outputLanguages an annotationRef and announcementRef) I describe in the draft. We should discuss this in the developer group.

Quote:* trainPart -> announcementRefs: how should it be interpreted when two annoucements have the same "rules" (periodic/trigger)? Are they played one after the other or do one overwrite the second?
If several announcementRefs exist with the same rules, they must be output one after the other. The order may be controlled by the @priority attribute.

Best regards, Milan

---

Subject: Re: Extensions to railML for passenger information at stations
Posted by Milan Wölke on Tue, 27 Aug 2019 15:01:19 GMT
View Forum Message <> Reply to Message

Hi,

Während unserer letzten timetable Entwickler Telco haben wir folgendes Thema diskutiert:
Quote:
Quote:
* passengerInfoLanguages: was gilt wenn keine passengerInfoLanguages auf einem StopDescription definiert ist? Ist es den aus dem TrainPart? Ich würde hier gut dokumentieren, wie es zu interpretieren/Vortrittsregeln ist.

Prinzipiell würde ich erwarten, dass wenn keine expliziten Sprachen beim Datenaustausch angegeben wurden, die Systemdefaults des empfangenden Systems gelten. Man könnte aber auch darüber nachdenken, ob das nicht eher für die zweite Art der Abbildung, die ich im Entwurf beschreibe (outputLanguages an annotationRef und announcementRef) spricht. Sollten wir in der Entwicklergruppe noch diskutieren.

Wir haben uns unter Berücksichtigung der oben stehenden Anmerkungen für den zweiten Implementierungsansatz entschieden. D. h. daß für jede annotationRef und jedes announcementRef separate Ausgabesprachen angegeben werden können. Das führt dazu, dass die Beschreibung direkter ist und der Bezug leichter herzustellen ist. Außerdem erlaubt es eine größere Flexibilität, weil so wichtige Information mehrsprachig kommuniziert werden können, wogegen weniger wichtige Infos mit einer reduzierten Mehrsprachigkeit ausgegeben werden können.

---

During our last timetableable developers phone conference we discussed the following topic:

Quote:
Quote:

* passengerInfoLanguages: what applies when no passengerInfoLanguages is defined on a StopDescription? Is i the one from the TrainPart? I would write a good documentation about the way to interpret this and about the priorities.

In principle, I would expect that if no explicit languages were specified during data exchange, the system defaults of the receiving system would apply. One could also consider whether this is more in favor of the second approach (outputLanguages an annotationRef and announcementRef) I describe in the draft. We should discuss this in the developer group.

We have chosen the second implementation approach, taking into account the above comments. This means that separate output languages can be specified for each annotationRef and each announcementRef. This makes the description more direct and the reference easier to make. It also allows greater flexibility because this allows important information to be communicated multilingually, while less important information can be output with reduced multilingualism.

Best regards, Milan

---

Subject: Re: Extensions to railML for passenger information at stations
Posted by Milan Wölke on Mon, 11 Nov 2019 14:48:20 GMT
View Forum Message <> Reply to Message

Hi,
für Interessierte gibt es nun im railML 2 SVN eine der Spezifikation entsprechende Implementierung ( https://svn.railml.org/railML2/branches/passengerinfo-at-sta tions/:wink:.

Bei der Implementierung ist mir noch der Gedanke gekommen, dass die Abbildung noch etwas vielseitiger eingesetzt werden könnte, wenn man den Scope der Änderung etwas erweitert. Mit der diskutierten Lösung könnte man nämlich eigentlich auch Ansagen und Fahrgastinfo im Fahrzeug beschreiben, wenn man das entsprechende Ausgabemedium bei <annotationRef> und <announcementRef> angibt. Ich würde mir vorstellen, dafür einfach ein erweiterbares Enum anzulegen, dass über die Werte "Station", "Train" und Other (weitere Werte) verfügt.

Was haltet ihr davon? Welche weiteren Werte sollte ein solches Enum haben?

---

for those interested there is now an implementation in railML 2 SVN according to the specification ( https://svn.railml.org/railML2/branches/passengerinfo-at-sta tions/:wink:.

During the implementation it occurred to me that the modelling could be used more widely if the scope of the change was extended. With the discussed solution one could actually also describe announcements and passenger information in the train, if one specifies the appropriate output medium with <annotationRef> and <announcementRef>. I would imagine to simply create an extensible enum with the values "Station", "Train" and Other (more values).

What do you think? What other values should such an enum have?

Best regards, Milan

---

Subject: Re: Extensions to railML for passenger information at stations
Posted by Martin Griesemer on Thu, 09 Jan 2020 07:20:10 GMT
View Forum Message <> Reply to Message

Hallo Milan

Ich halte es für sinnvoll vor allem in Kombination mit dem Trigger: 1000m vor Ankunft könnten wir die Voransage machen (Audio und Text) sowie 200m nach Abfahrt die Willkommen Ansage. Also für mich sieht es so aus das die kleine Anpassung (scope = Train oder Station) alles beinhaltet was benötig ist für Fahrgastinfo im Zug.

Mit freundlichen Grüßen,

---

Hello Milan

I think it makes sense especially in combination with the trigger: 1000m before arrival we could make the arrival announcment(audio and text) and 200m after departure the welcome announcement. So for me it looks like this small adjustment (scope = train or station) includes everything that is needed for passenger information in the train.

Regards,

---

Subject: Re: Extensions to railML for passenger information at stations
Posted by Milan Wölke on Sun, 12 Jan 2020 15:11:58 GMT
View Forum Message <> Reply to Message

Hi,

gut, dann werde ich das Ticket und die Modellierung im Branch entsprechend anpassen.

---

alright, then I will update the implementation in the svn branch as well as the corresponding ticket accordingly.

Best regards, Milan

---

Subject: [railML2] Re: Extensions to railML for passenger information at stations
Posted by Milan Wölke on Tue, 07 Apr 2020 17:58:50 GMT
View Forum Message <> Reply to Message

Hi guys,

after reviewing the changes regarding this issue, I want to suggest yet one more minor change. The current approach models announcementRefs to always require either a trigger or a periodic element. However I was asked by a railML partner if it was possible to specify announcements without as well. The use case for this is, that a planning system that feeds data via railML into a passenger information system allows specifying announcements for a train that are only potentially to be played. Basically a preselection of announcements is made based on the properties of the train. In order to support this in railML I see two general options:
1) change announcementRef so that it is possible to neither specify a trigger nor periodic playback
2) add a new trigger "manual" which specifies that the refered to announcement is not to be played unless a user decides otherwise

I personally would prefer the first solution. What is your opinion?

---

Nach einem Review zu den Änderungen aus diesem Thread, möchte ich noch eine weitere kleine Änderung vorschlagen.
Der derzeitige Ansatz sieht vor, dass die announcementRefs immer entweder ein trigger oder ein periodic Element benötigen. Ich wurde aber von einem railML-Partner gefragt, ob es nicht möglich sei, Ansagen auch ohne zu spezifizieren. Der Anwendungsfall hierfür ist, dass ein Planungssystem, das Daten über railML in ein Fahrgastinformationssystem einspeist, die Festlegung von Ansagen für einen Zug erlaubt, die nur potentiell abgespielt werden sollen. Im Prinzip wird eine Vorauswahl der Ansagen anhand der Eigenschaften des Zuges getroffen. Um dies in railML zu unterstützen, sehe ich zwei generelle Varianten:
1) Änderung von announcementRef, so dass es möglich ist, weder einen Trigger noch eine periodische Wiedergabe zu spezifizieren
2) einen neuen Trigger "manual" hinzufügen, der angibt, dass die betreffende Ansage nicht ohne Eingreifen des Benutzers abgespielt werden soll.

Ich persönlich würde die erste Lösung vorziehen. Was ist eure Meinung?

Best regards, Milan

Subject: Re: [railML2] Re: Extensions to railML for passenger information at stations
Posted by on Tue, 14 Apr 2020 18:10:49 GMT
View Forum Message <> Reply to Message

Hi Milan,

I am not sure what you mean with "only potentially". In a deterministic system, there should always be a cause for each effect. It seems to me that the cause should only not be exchanged in

your use-case, or the cause should not be standardised by railML. A kind of "trigger=other:...", but surely not "trigger=none".

However, it is not important whether I understand it or not. I only want to ask you, as you yourself already know, to reduce wild-grow in railML. So if possible, please try to describe the actual trigger/cause for the announcement as tellingly as possible.

> 1) change announcementRef so that it is possible to neither
> specify a trigger nor periodic playback

In general, an attribute which end with "...Ref" should always be restricted to values of id's defined in the same railML file. I my opinion, this is a basic "common" rule at least in railML 2.

> 2) add a new trigger "manual" which specifies that the
> refered to announcement is not to be played unless a user
> decides otherwise

Sounds better than option 1 for me. It would be the right one in case the driver/conductor/guard presses a kind of button to trigger the announcement. In other cases - not linked to staff/humans - please specify the trigger in a more technical resolvable way - I think.

Best regards,
Dirk.

---

Subject: Re: [railML2] Re: Extensions to railML for passenger information at stations
Posted by Milan Wölke on Wed, 13 May 2020 15:47:33 GMT
View Forum Message <> Reply to Message

Hi all,

I implemented the change following the requirement to support manual triggers according to Dirks suggestion. The current version can be reviewed under https://svn.railml.org/railML2/branches/passengerinfo-at-sta tions/. Or if you prefer the integrted version of all 2.5 timetable changes so far under https://svn.railml.org/railML2/branches/railML2.5-dev/. Please feel free to comment.

Best regards, Milan

---