
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 StopDescription nicht ganz. Hättest du hier ein Beispiel? Ich sehe aber schon, dass man den gleich Type "tAnnouncementRef" 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 announcementRef: I don't really see the need of the "periodic" announcement on the StopDescription. Do you have any example here? I still do see the advantage of reusing the type tAnnouncementRef.

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 it 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 announcements 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