

---

Tickets #188 und #247

Posted by on Fri, 18 Sep 2015 08:14:03 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Hallo allerseits,

Ticket #247 wurde als für railML 2.3 geschlossen und das damit zusammenhängende Problem damit als gelöst deklariert.

Wir haben in den letzten Jahren schon oft über die Problematik "Änderungsübertragung vs. Gesamtübertragung" und damit im Zusammenhang stehende Aspekte gesprochen. Bisher war die dabei unter'm Strich herauskommende Lesart immer "erfordert weitgehendere Lösungen, die erst mit 3.x und Refactoring möglich werden", u. a. nachzulesen beim Diskussions-Thread 26.02.2013..24.09.2013 [1].

Insofern sollte Einigkeit darüber bestehen, dass die Schließung von Ticket #247 bei Weitem keine Lösung ist, sondern eher ein Provisorium, über das man geteilter Meinung sein kann. Möglicherweise wäre hier eine Verwendung von any-Attributen die bessere Wahl gewesen, um auszudrücken, dass ein allgemeingültiger railML-Konsens noch nicht erreicht ist.

Ich möchte anregen, beim Ticket #247 einen Querverweis auf Ticket #188 (gleiches Thema bei railML 3.0) anzubringen und zu vermerken, dass es sich um eine provisorische, nicht allgemeingültige Lösung handelt.

Viele Grüße,  
Dirk Bräuer.

[1] [news://news.railml.org:119/l176vg\\$tbv\\$1@sifa.ivi.fhg.de](news://news.railml.org:119/l176vg$tbv$1@sifa.ivi.fhg.de)

P.S.:

Gleichzeitig sollte für das provisorische "Löschkennzeichen" eines <trainPart>s von railML 2.3 im Wiki durch den Einführer oder Koordinator festgehalten werden,

a) ob ein <trainPart> mit "Löschkennzeichen" immer an allen seinen Verkehrstagen auf seinem gesamten Laufweg ausfällt und er demzufolge keine /operatingPerodRef/ und keine <ocpTT>s haben darf bzw. dass diese zu ignorieren sind oder

b) ob ein <trainPart> mit "Löschkennzeichen" nur an dem konkret genannten Laufweg und Verkehrstagen ausfällt, d. h. eventuell im Zielsystem existierende weitere Verkehrstage / Laufwegteile unverändert erhalten bleiben.

Da im Ticket #247 schon erwähnt, dass die Bezugsbasis in railML derzeit nicht definiert ist, sollte es auf (a) hinauslaufen (was hiermit als mein konkreter Vorschlag gelten kann).

Wenn wir das undefiniert belassen, werden wir ein großes Problem auf die Praxis abwälzen mit der Folge, dass die meisten Anwendungen das "Löschkennzeichen" ablehnen oder ignorieren. Dieses Dilemma verdeutlicht schon etwas, wo das Unbefriedigende an der 2.3er Lösung liegt.

---