

offiziellen Status erlangte? Und das nach Verkehrstagen unterschieden: Zug ist im Juli bis September bestellt worden, im Juli soll er immer noch fahren, im August ist er bereits abbestellt, im September ist er noch abzubestellen...

- - -

Letztendlich kann man es auch anders aufrollen: Bei Datenübertragung in einem komplexen Prozess muss üblicher Weise irgendwo ein Änderungsabgleich stattfinden. Dies kann beim Export `_vor_ RailML` stattfinden - RailML überträgt dann Änderungsinformationen und bezieht sich auf einen früheren Export - oder es kann beim Import `_nach_ RailML` stattfinden, d. h. beim „mergen“ (=soll ein Anglizismus sein - bitte englisch „auslesen“) der RailML-Daten mit den aktuell im System vorhandenen Daten.

Es scheint, dass beide Wege gleichwertig sind. Die bisherige RailML-Philosophie war „keine Änderungsinformationen“, d. h. „mergen“ beim Einlesen. Das Einlesen ist ohnehin ein komplexerer Prozess als das Rausschreiben, wegen der zusätzlich notwendigen Datenintegritätsprüfungen.

Zu bedenken ist immer auch, dass u. U. verschiedene Datenquellen in Betracht kommen: Ein „mergen“ beim Einlesen kann auch Daten zusammenführen, die aus verschiedenen Quellen kommen. Ein Änderungsexport hingegen würde sich immer auf die zuvor aus dem eigenen System exportierten Daten beziehen, d. h. hier ist im Gesamtprozess kein Informationsgewinn möglich.

Wir (iRFP) haben uns dieser Lesart angepasst, d. h. wir können beim Einlesen „mergen“, können aber derzeit keinen Änderungsexport anbieten. Wir würden aus Aufwandsgründen in absehbarer Zeit nicht beides anbieten, zumal wie gesagt derzeit kein Mehrwert erkennbar ist. Insofern würde ich das von unserer Seite erst einmal zurückhaltend betrachten wollen.

Ich verstehe, dass die Situation vielleicht anders aussieht, wenn zwei nicht gleichwertige Systeme Daten austauschen - d. h. wenn beim Einlesen aus irgendwelchen Gründen weniger Prozesskapazität vorliegt als beim Rausschreiben. Das wäre dann aber vielleicht ein Fall, der nicht mehr im allgemeingültigen RailML auftauchen muss - eher ein individueller Aufsatz oder eine bilaterale Lösung.

Viele Grüße,
Dirk.
