
Subject: Re: [railML2] Re: Documentation of attribute "pathStatus" and proposal for new value "offered"

Posted by [Milan Wölke](#) on Mon, 25 May 2020 13:36:24 GMT

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

Hi all,

just to keep you in the loop. The discussion on the possible values of pathStatus is more or less finished. The new statemachine seems acceptable to JBD and noone else has added further requirements.

However the original topic of this thread somehow got lost. Since the discussion on pathStatus was fruitful I don't think that is a problem, the original issue should still be discussed, though.

Originally we noticed that the attribute processStatus was not being used in a common way, not even by the members of the timetable developer group. That's why we wanted to make this attribute obsolete to indicate that there is no common understanding of its purpose. We still stand by this opinion. However we also have a requirement for 2.5 to be able to communicate scheduled information vs. actual information. E.g. the requirement is to be able to communicate that the track a train is supposed to arrive at has changed.

In the course of the discussion it turned out, that modelling this individually is not in the interest of the developer group as it may lead to an attribute by attribute approach to allow for description of changing information. The developer group decided that it may be an option to introduce a well defined element in place of processStatus that would allow us to transfer different versions of the same train thus allowing for detection of changed attributes. This new element would have an attribute to define the status as well as a timestamp to indicate when that status was reached (also a requirement of a railML partner).

One could imagine a structure like this:

```
<train>
  <version status="yearly_schedule" changeTimestamp="..." />
  <trainPartSequence>...</trainPartSequence>
</train>
```

This would allow to provide a train with the new information while still being able to transfer the original data. Maybe it would also be a good idea to provide some kind of rank based on an integer rather than a status that is based on an enumeration, as it would allow to define which version would overwrite which version without the need of specifying a statemachine, which would most probably be use case dependent.

What is your opinion on this. Should railML support this kind of versioning? What would you prefer, an integer based rank or an enum based status or both? Do you have other ideas on how to model this?

Looking forward for your views on this.

Best regards, Milan

Hallo zusammen,

nur um euch auf dem Laufenden zu halten. Die Diskussion über die möglichen Werte von pathStatus ist mehr oder weniger abgeschlossen. Die neue Statemachine scheint für JBD akzeptabel zu sein, und niemand sonst hat weitere Anforderungen hinzugefügt.

Allerdings ist das ursprüngliche Thema dieses Threads irgendwie verloren gegangen. Da die Diskussion über pathStatus fruchtbar war, denke ich nicht, dass das ein Problem ist, aber das ursprüngliche Thema sollte trotzdem noch diskutiert werden.

Ursprünglich stellten wir fest, dass das Attribut processStatus nicht einheitlich verwendet wurde, nicht einmal von den Mitgliedern der Fahrplanentwicklergruppe. Deshalb wollten wir dieses Attribut obsolet machen, um anzuzeigen, dass es kein gemeinsames Verständnis über seinen Zweck gibt. Wir stehen nach wie vor zu dieser Meinung. Wir haben jedoch auch die Anforderung, dass 2.5 in der Lage sein sollte, geplante Informationen gegenüber tatsächlichen Informationen zu kommunizieren. Die Anforderung besteht z.B. darin, mitteilen zu können, dass sich das Gleis, auf dem ein Zug ankommen soll, geändert hat.

Im Laufe der Diskussion stellte sich heraus, dass eine individuelle Modellierung nicht im Interesse der Entwicklergruppe ist, da sie zu einem attributweisen Ansatz führen kann, um die Beschreibung der sich ändernden Informationen zu ermöglichen. Die Entwicklergruppe entschied, dass es eine Option sein könnte, anstelle von processStatus ein klar definiertes Element einzuführen, das es uns erlauben würde, verschiedene Versionen desselben Zuges zu übertragen und so die Erkennung von geänderten Attributen zu ermöglichen. Dieses neue Element hätte ein Attribut zur Definition des Status sowie einen Zeitstempel, der anzeigt, wann dieser Status erreicht wurde (ebenfalls eine Anforderung eines railML-Partners).

Man könnte sich eine Struktur wie diese vorstellen:

```
<train>
  <version status="yearly_schedule" changeTimestamp="..." />
  <trainPartSequence>...</trainPartSequence>
</train>
```

Dies würde es ermöglichen, einen Zug mit den neuen Informationen zu versorgen, während die ursprünglichen Daten weiterhin übertragen werden können. Vielleicht wäre es auch eine gute Idee, eine Art Rang auf der Basis eines Integerwertes statt eines Status, der auf einer Aufzählung basiert, bereitzustellen, da so definiert werden könnte, welche Version welche Version überschreiben würde, ohne dass eine Statemachine festgelegt werden müsste, die höchstwahrscheinlich vom Anwendungsfall abhängig wäre.

Was ist eure Meinung dazu? Sollte railML diese Art der Versionierung unterstützen? Was würdet ihr bevorzugen, einen Integer-basierten Rang oder einen Enum-basierten Status oder beides? Habt Ihr andere Ideen, wie man das modellieren könnte?

Ich freue mich auf eure Meinung dazu.

Mit freundlichen Grüßen, Milan.