
Subject: Attribute processStatus to be declared deprecated for <train>, <trainPart> and <trainGroup>

Posted by [Milan Wölke](#) on Thu, 29 Aug 2019 15:59:00 GMT

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

Hallo,

bei der letzten Timetable Entwickler Telco wurde beschlossen, dass mit der Version railML 2.5 das Attribut @processStatus am <train>, <trainPart> und <trainGroup> für deprecated zu erklären. Hintergrund ist, dass es für dieses Attribut keine standardisierte Semantik gibt und die Teilnehmer, die dieses Attribut verwenden erklärten, dass sie die Verwendung mittelfristig durch andere dem Standard entsprechende Modellierungen ersetzen wollen.

Sollte es in der Community weitere Verwendungen dieses Attributes geben, so bitte ich diese hier zu kommunizieren. In diesem Fall würde die Entscheidung nochmals diskutiert werden müssen. Vielen Dank schon mal im Voraus.

at the last Timetable developer Telco it was decided that with version railML 2.5 the attribute @processStatus am <train>, <trainPart> and <trainGroup> should be declared deprecated. The background is that there is no standardized semantics for this attribute and the participants who use this attribute declared that they want to replace the use with other standard compliant representations in the medium term.

If there are other uses of this attribute in the community, please communicate them here. In this case the decision would have to be discussed again. Many thanks in advance.

Best regards, Milan

Subject: Re: Attribute processStatus to be declared deprecated for <train>, <trainPart> and <trainGroup>

Posted by [Milan Wölke](#) on Tue, 15 Oct 2019 11:43:01 GMT

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

Ticket for this issue was created at <https://trac.railml.org/ticket/364>

Ein Ticket für diesen Vorgang wurde unter <https://trac.railml.org/ticket/364> angelegt.

Subject: Re: Attribute processStatus to be declared deprecated for <train>, <trainPart> and <trainGroup>

Posted by [Torben Brand](#) on Thu, 06 Feb 2020 14:08:30 GMT

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

The Norwegian railML community uses @processStatus in it's data exchange via railML. We even plan to extend the use with other:*. We suggest to adopt these other: values in railML2.5/3

or to adopt them in the mentioned (but unpublished) new solution in the initial forum posting. The background UC for the use of @processStatus is:

- UC:TT:LongTermStrategicTimetabling
- UC:TT:ATimetableForACompetition
- UC:TT:SlotOrdering

We plan to extend to values in @processStatus to form the complete life cycle of a train(part). This from pre designed conceptual slots from a governmental agency that oversees developing the rail network over operators giving bids to the infrastructure manager approving the timetable.

We use the attribute @processStatus and not @pathStatus as it gives us more flexibility as it is placed on <trainPart>, <train> and <trainGroup>. @pathStatus is only under <trainpartSequence>. This "middle position" gives an implicit referring to the parent <train> and the child <trainPart>. We would like to be more flexible. Especially as we are working on extensions for <trainGroup> and using <trainPart> without a <train> to make generic path descriptions (forum post currently in draft).

What happened with the development idea for a general sub element <status> (similar to IS:state) in this forum string?: https://www.railml.org/forum/index.php?t=msg&goto=578&&srch=status#msg_578

We should consolidate the enumeration lists in @processStatus and @pathStatus (and maybe even look to IS:state@status?) and think for a proper placement in the scheme.

In principal we like the enumeration list in @pathStatus extended with a "conceptual" value and the "actual" value from @processStatus.

Both enumeration lists (@processStatus and @pathStatus) lack definition of their values. This should be amended. We have some suggestions.

I would like to ask the TT coordinator or the stated participants who use this attribute and have declared that they "want to replace the use with other standard compliant representations in the medium term", to declare the intended replacement here in the forum.

Subject: Re: Attribute processStatus to be declared deprecated for <train>, <trainPart> and <trainGroup>

Posted by on Mon, 17 Feb 2020 14:53:39 GMT

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

Hello,

we discussed the future of the "processStatus" attribute during the railML telephone conference, as the documentation in the wiki is inadequate and none of the participants could provide a plausible way of using it. All participants of the telephone conference currently using this attribute use the attribute in a different context, so that a

functioning exchange is unlikely. Furthermore, only a few of the possible values of the enumeration are used or the value range is extended by individual "other:xxx" values. For this reason, we have proposed to declare the attribute as "deprecated".

For mapping different planning states or versions of a train I would suggest to introduce a new attribute or element to avoid confusion. For this purpose we would appreciate a modeling suggestion from you. If you want to keep the existing attribute "processStatus", we would also be interested in information about which values of the enumeration you use and what exactly is encoded with them.

I would not try to merge "pathStatus" and "processStatus". With "pathStatus" different states of a train are described during a train path allocation process. A train can have several different pathStatus if it is ordered at several different infrastructure managers. I understand the "pathStatus" attribute here as additional information of a train. The new attribute for the planning statuses, on the other hand, should help to distinguish different instances of <train> elements, so it becomes part of an external primary key of a train/train part. For these reasons I would also leave "pathStatus" at <trainPartSequence> and the new attribute at <trainPart> / <train> / <trainGroup>.

Best regards
Christian Rößiger

--

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: Attribute processStatus to be declared deprecated for <train>, <trainPart> and <trainGroup>

Posted by [Milan Wölke](#) on Wed, 19 Feb 2020 11:23:43 GMT

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

Hi,

i think this issue can be linked with the topic discussed here. A discussion in the timetable developer group showed that in order to communicate a track change the use of a status like element is to be preferred over some individual modelling for the use case of a track change. This would have the benefit to be able to communicate all kind of changes to trains (track change, formation change, etc).

Looking at the requirements mentioned in the other thread, it should be possible to express the originator of the change (or reason for the new version) as well as the time when this change occurred. Furthermore the timetable developer group discussion considered introducing some kind of version counter which would allow to describe different version within the same planning state. In general I still think that processStatus needs to be replaced with some new well defined

element to express state information, since as Christian pointed out data exchange regarding the state seems highly unlikely given the different semantics currently in use by implementing systems.

Best regards, Milan

Subject: Re: Attribute processStatus to be declared deprecated for <train>, <trainPart> and <trainGroup>

Posted by [Torben Brand](#) on Wed, 19 Feb 2020 12:57:25 GMT

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

Dear TT community,

We value the quick reply and the suggestions by iRfP. However, we suggest not to make a new attribute, but to reuse @processStatus, as a pragmatic approach, as railML2 is in the end of its life cycle (But as they say in German: "the doomed live the longest ;-)"). This will NOT alter the values or the use of @pathStatus. We suggest reconstructing a cleaner model in railML3. Especially one that is aligned with the new TTR process of Rail Network Europe (<http://rne.eu/sales-timetabling/ttr/>).

Jernbanedirektoratet has discussed the TT process stages including the slot allocation process with the Norwegian IM Bane NOR. We have reached a common suggestion for a flexible and pragmatic implementation suggestion in railML2.

The general process phase/stage of the model is global for the railML file/transmission. This stage should be submitted, if desired, in the <metadata> element of railML. As these stages differ per country/IM today and differ in TTR we suggest NOT to standardise this information. An example for the global file @processStatus/@pathStatus metadata information could be:

```
<dc:source>processPhase: tactical (long term timetable negotiations) SubPhase: (X-5) Slot proposal published (offer from IM back to operators after their initial slot order) </dc:source>.
```

We have generalised the phases/stages of a timetable from idea to actual operated train (also see illustration) to the values: strategic, tactical, operational, historic. These could be used in the metadata together with more detailed description (see example above).

The individual elements could use the set value list of @pathStatus. For completeness we suggest extending the value list in either end of the timeline with "conceptual" and "actual".

We suggest the following value definitions:

We refer to "element" as being either trainPart, trainPartSequence, train or trainGroup

"conceptual" for slot construction or commissioning of the element, which is planned for the medium or long term. However, there are still no concrete (planning) activities for the construction of the element beyond the preliminary planning. Used in UC/phase TT:LongTermStrategicTimetabling and as a draft example from the commissioner in UC/phase TT:ATimetableForACompetition.

"planned": The construction or commissioning of the element is planned concretely and at short notice or concrete planning activities for the construction take place, e.g. design, approval or implementation planning, cost calculation, award of contracts (bid). It is not normally possible to use the element (run the train) as it has not been ordered and approved. Used in the proposal in UC/phase TT:ATimetableForACompetition and in UC/phase TT:SlotOrdering.

"ordered" The element is ordered for a slot allocation process from the bid party (railway undertaking/operator) to the capacity allocation authority (infrastructure manager).

"detailsRefused" The element has been reviewed (by the capacity allocation authority) and details have been modified/changed/adjusted pending approval by the initial ordering/bid party.

"confirmed" The element has been reviewed and approved (by the capacity allocation authority) as ordered by the initial ordering/bid party.

"cancelled" The element has been cancelled, but service will be maintained through a train replacement service (usually a buss). The service will be announced. Used in combination with @cancellation="true"

"notAvailable" The element has been retraced and there will be no other (train) service. This possible due to an error in the planning. The service will not be announced. Used in combination with @cancellation="true"

"actual" These are actual driven/operated trains and the time values with time@scope="actual" CAN have been set.

We suggest making the value list more flexible placeable. Comparable to IS:state which is placed on several elements including the <infrastructure> root element. For this we suggest to use, in addition to @pathStatus,@procesStatus as this is already available under <train>, <trainPart> and <trainGroup>. In railML2.5 it should be considered placing the attribute also on <timetable> for a set global value that inherits/apply to all elements unless defined further down the hierarchy and thus breaking/overwriting the inherited value.

This makes it also possible to mix different process state values for different trains in the same file. For instance, you can indicate "conceptual" trains to indicate a desired template/pattern (but which shall never run as actual trains) on a <trainGroup> together with the "confirmed" trains (that run later in the process after the path allocation has been closed unless operative cancelled) on <train>.

With the correct use of other:* and extended to the Norwegian use case we will use the following values:

@pathStatus="nor:conceptual" and "nor:actual" in addition to the existing values.

@processStatus="nor:conceptual","planned","nor:ordered",

"nor:confirmed","nor:detailsRefused","nor:cancelled","nor:notAvailable " and "actual".

We suggest NOT to use in @processStatus the values: "calculated", "toBeChecked", "changed" and "imported". This as the general data state can be set in <metadata> (calculated/imported) and annotations can be set in combination with @processStatus/pathStatus (toBeChecked/changed).

Example:

An operator orders 6 train slots at two IM's. All train@processStatus="ordered". Additionally, the operator delivers a pattern train with trainGroup@proccesStatus="conceptual" to indicate the desired pattern for a service.

The review of one of the IM's result in: two of the trains are partially outside its jurisdiction. The but the part inside its area is ok for one train (trainPartSequence@pathStatus="confirmed"), but can only be accommodated for one of two trainParts for the other train

(trainPart@proccesStatus="notAvailable" and "confirmed" for the other) For the remaining 4 trains completely inside it's area 2 trains are ok as ordered (train@proccesStatus="confirmed" or trainPartSequence@pathStatus="confirmed" for all trainPartSequence of the train). For one train some of the departure times of a trainPart must be adjusted

(trainPart@proccesStatus="detailsRefused") and the last train cannot be accommodated at all (train@proccesStatus="notAvailable").

Please note that this is just a suggestion and should not interfere with existing use of railML.

PS. I was not able to upload a picture or an PDF attachment to this reply. And formating is a cumbersome affair in the forum. So I have placed a pdf with formating and illustrations here: <http://cloud.railml.org/index.php/s/JeZ5qNffx8Aetqz> (public link that will expire 4.3.2020 as set by the cloud system)

Subject: Documentation of attribute "pathStatus" and proposal for new value "offered"

Posted by on Mon, 24 Feb 2020 11:08:33 GMT

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

Hello,

[German Text below]

for the currently missing description of the allowed values of the attribute "pathStatus" in the railML wiki I would offer the following interpretation:

planned: The train is planned for ordering at the infrastructure manager (IM), but no order has yet been transmitted (initial state).

ordered: The train has been ordered at the IM.

confirmed: The IM has submitted an offer for a train path to the railway undertaking (RU) and the RU has accepted this offer.

detailsRefused: I have no idea what this value represents.

cancelled: The RU has cancelled the train at the IM.

notAvailable: The IM has rejected the order of then train path.

From my point of view, the value "offered" (between "ordered" and

"confirmed") is missing in this enumeration, which describes the status that the train path was offered by the IM to the RU.

Best Regards,
Christian Rößiger

[German]

für die derzeit fehlende Beschreibung der möglichen Werte des Attributs "pathStatus" im railML-Wiki würde ich folgende Interpretation anbieten:

planned: Der Zug ist für die Bestellung beim Infrastruktur-Manager (EIU) vorgesehen, es wurde aber noch keine Bestellung übermittelt (initialer Zustand).

ordered: Der Zug wurde beim EIU bestellt.

confirmed: Das EIU hat ein Trassen-Angebot an das Verkehrsunternehmen (EVU) übermittelt und das EVU hat dieses Angebot angenommen.

detailsRefused: Ich habe keine Ahnung, was mit diesem Wert ausgedrückt werden soll.

cancelled: Das EVU hat den Zug beim EIU storniert.

notAvailable: Das EIU hat die Bestellung des Zuges abgewiesen.

Aus meiner Sicht fehlt in dieser Enumeration der Wert "offered" (zwischen "ordered" und "confirmed"), mit dem der Status beschrieben wird, dass die Trasse vom EIU an das EVU angeboten wurde.

Viele Grüße
Christian Rößiger

--

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: Documentation of attribute "pathStatus" and proposal for new value "offered"

Posted by [Milan Wölke](#) on Wed, 26 Feb 2020 14:21:51 GMT

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

Hi all,

I think detailsRefused refers to the reply when an order of a train is refused by the IM. If I understand correctly the IM may also include a counteroffer annotated with that @processState.

What I dont really understand is how to work with the proposed @processStatus "cancelled". I get the feeling that this state isnt the same kind of @processStatus as the rest of the suggested values. I also get the feeling that we break the semantics of the @cancellation attribute of train

and trainPart by introducing this, because the behavior of the importing system would need to change when evaluating the attribute @cancellation depending on what the @processStatus is saying.

The same applies to the @processStatus in general. I think, although railML 2.x is at the end of its lifecycle, we shouldn't alter the behavior of existing attributes. We can add new ones, alright, but changing existing ones is something we always tried to avoid in the past for the sake of compatibility. Accordingly I would rather introduce a new element <status> for train, trainGroup and trainPart. I also think that an element would be a better choice here in order to store additional metadata regarding the state(change), like when did the state change to the current value.

Best regards, Milan

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

Posted by [Milan Wölke](#) on Tue, 07 Apr 2020 13:36:35 GMT

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

Hi all,

in the last timetable developer phone conferences we discussed about the different enum values of @pathStatus. We came to the conclusion that the current states of @pathStatus are not enough for modelling of a basic slot ordering process. Therefore we decided to extend the enum with two new values: offered and conceptual. The general idea of the statemachine behind that is the following:

A train starts out as "conceptual", which means that a slot for that train is not to be ordered (yet). If a slot for a train is to be ordered but no order has been issued yet, the train is annotated with the state "planned".

Once a slot for a train has been ordered the state "ordered" is used. Now the IM will have to respond to the order. The possible results of that could be: "notAvailable", if the IM cannot fulfill the request; "offered", if the IM can provide the requested slot or "detailedRefused" if the IM cannot fulfill the request but offers a slot that is similar to the requested one.

If the IM chooses "detailedRefused" or "offered" in the response to the request is up to the IM. Very often an offered slot of the IM will not be exactly fitting the requested one. By choosing "offered" or "detailedRefused" the IM can decide if from his point of view the request is fully complied with or not and inform the RU accordingly.

If the slot offered by the IM suits the RU the state "confirmed" is used. If it is not, the state "cancelled" is used to cancel the ordering process.

As I said initially this means introducing two new states: conceptual as a new starting point and allowing to include trains in railML that are not (yet) ready for the ordering process; and offered for the IM to indicate that a request has been fully complied with.

If it is necessary to encode further states for modelling slot ordering those states should be defined using other:anything.

@Torben: Would you agree with the meaning and usage of the existing and new enum values of @pathStatus.

Best regards, Milan

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 dont 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. Thats 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.

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

Posted by on Tue, 16 Jun 2020 08:31:37 GMT

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

Dear Milan and all,

first, Milan, let me thank you for your work and moderation from my side.

> Was ist eure Meinung dazu?

So far, from my understanding, a railML file was only able to encode a "current" planning state, no planning history. Any "next" railML file overwrites the "previous" railML file. What is next and what is previous could be encoded by meta-data information.

Since we can assume that a software which receives railML files can somehow store the context of a railML file (or the railML file itself), we can assume that a receiving software can also restore the "planning history" of each element of a railML file if necessary.

I understand that you want to introduce the possibility to encode a planning history of trains in one railML file. This opens a huge field and therefore leads to possible problems as such:

- It would be a redundancy to the already existing "planning history" which comes automatically from an "previous" and a "next" railML file containing the same train. How to solve any conflicts coming out of the two ways (e. g. a train is encoded in the latest file with 3 different planning states but I actually have 5 different railML files of that train) can only be solved by the use case or not at all.

- You want to introduce versioning of trains only, not of all railML elements. But as long as trains change, also other elements can change which are connected with trains. Operating periods,

categories etc... not to speak about infrastructure. At the end, it would be necessary to introduce a compatible versioning of the same kind for all elements.

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

- Please be aware that this touches the definition of primary keys within a railML file. A train is identified by its trainNumber, additionalTrainNumber and scope, so far. After your introduction, it would also be identified by status and/or version number.

The existing attribute processStatus was never to be part of a primary key, so processStatus was only to encode the latest status of a train, not a history. As you wrote, it was also probably never used in a common way. But that does not mean that there must be a versioning of trains only to improve processStatus.

-END OF CONCERNS-

```
> What would you prefer, an integer based
> rank or an enum based status or both?
```

However, if such a versioning should be introduced, I would prefer something like you suggested: Easy and flexible. May be a bit generic. Not too many semantic constraints:

- an integer version number or something similar,
- possibility to version not only <train> elements but also <trainPart> elements (which is very essential I think because I can imagine many cases where only <trainPart>s change but not the data of the <train> element change), <category> elements, possibly also <operatingPeriod> elements. I would also welcome, for the sake of consequence, to be able to do the same with infrastructure, for instance <ocp>s.

SUMMARY:

- I have many concerns on opening such a Pandora's box in railML. I would not do it at the moment because we (iRFP) do not have or see use cases where it is actually really necessary.
- If a versioning should be introduced, I would prefer something in the direction you suggested. It should be easy and flexible, possibly a bit generic, and include other elements as well.

Best regards,
Dirk.

Am 25.05.2020 um 15:36 schrieb Milan Wölke:

```
> 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

> state machine, 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.

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

Posted by [Stefan Jugelt](#) on Thu, 16 Jul 2020 14:20:37 GMT

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

Dear all,

the TAF TSI (see https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv%3AOJ.L_.2014.356.01.0438.01.ENG) contains a legal binding process description of the short term path request.

For the TAF TSI the following messages for a path request exists:

1. PathRequestMessage (RU --> IM)
2. ReceiptConfirmationMessage (IM --> RU)
3. PathDetailsMessage (IM --> RU)

4. PathConfirmedMessage (RU --> IM)
5. PathDetailsRefusedMessage (RU --> IM)
6. PathNotAvailableMessage (IM --> RU)
7. PathCanceledMessage

The messages are exchanged between the railway undertaking and the infrastructure manager in the order above. Some of the messages are optional, such as PathDetailsRefusedMessage.

The messages above may contain an additional element "TypeOfInformation", carrying supplementary information about the status of the path. The following values are allowed:

- 01 harmonisation - in process
- 02 harmonisation - accepted
- 03 harmonisation - rejected
- 04 harmonisation - completed
- 05 path study request
- 06 pre-arranged path/reserve capacity
- 07 create offer
- 08 coordination update
- 09 draft offer
- 10 draft alternative offer
- 11 observation - in process
- 12 observation - complete
- 13 preparation of final offer - in process
- 14 preparation of final offer - accepted
- 15 preparation of final offer - rejected
- 16 final offer
- 17 final offer - accepted
- 18 alternative offer accepted
- 19 pre-accepted offer
- 20 Final Offer rejected
- 21 no alternative available
- 22 booked
- 23 - preparation of draft alternative offer is in progress
- 24 - Preparation of draft offer - accepted
- 25 offer/final offer rejected (without revision)
- 26 alternative offer rejected (without revision)
- 27 offer/final offer rejected (revision required)
- 28 alternative offer rejected (revision required)
- 31 Close Dossier
- 30 Create Dossier
- 40 Fully Assembled Path (FAP, constructed path)
- 41 Final Offer rejected
- 42 Preparation of draft offer accepted
- 43 Preparation of draft offer rejected
- 44 Draft offer rejected
- 45 - Draft no alternative available
- 50 activate path (utilisation notification)

51 deactivate path (utilisation notification)
52 confirmation of utilisation notification
53 Path and train cancelled

More information about the is available in the sector handbook: http://taf-jsg.info/?page_id=280

Kind regards,

Stefan Jugelt

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

Posted by [Milan Wölke](#) on Mon, 14 Sep 2020 10:29:02 GMT

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

Hi Stefan,

looking at the description you provided, I'd say that the new pathStatus interpretation and the added new enum values nicely align with the TAF TSI process. Thanks for the input.

Best regards, Milan

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

Posted by [Milan Wölke](#) on Mon, 14 Sep 2020 10:52:48 GMT

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

Hi there,

in order to keep you informed, I would like to give a short summary of the current state of affairs on this topic. After Dirk documented his concerns here about the introduction of a new general status for <train>s etc., the developer group changed their mind. It is considered a risk to introduce such a general attribute, because the semantic effects on a railML document with this extension are hardly assessable.

As far as the mapping of a scheduled track compared to the actual track, which can currently be specified, a rather pragmatic solution was found that focuses solely on this issue. We have also discussed Dirk's concerns about this, as well as the suggestion that the scheduled track could also be determined from a comparison of previous railML documents, however, this does not apply to the usecase which is the basis for the requirement to include a scheduled track model. Specifically, this concerns passenger information systems that only import the current day and cannot perform such a comparison with a previously delivered document, since the corresponding data may not (yet) have been imported at all.

Therefore an approach was chosen which allows to specify an <originalTrackInfo> at the ocpTT. This offers the same modeling possibilities as for the modeling of the actual track so far.

A corresponding implementation is isolated in the branch
<https://svn.railml.org/railML2/branches/trackchange/> for review.

It remains open, how @processStatus should be handled. @Torben: after we have adapted the @pathStatus in such a way that the different states of a train can be considered in the context of the slot ordering, is there still the objection of JBD regarding the deprecation of the @processStatus attribute? Personally, I would still consider this to be useful, since it has been shown that there is no standardizable use of this attribute.

Hallo zusammen,

um euch zu informieren, möchte ich hier wieder eine kurze Zusammenfassung des aktuellen Standes zu diesem Thema geben. Nachdem Dirk seine Bedenken hier dokumentiert hat, was die Einführung eines neuen allgemeinen Status für <train>s etc. betrifft, hat die Entwicklergruppe ihre Meinung geändert. Es wird als Risiko angesehen ein solches allgemeines Attribut einzuführen, weil die semantischen Auswirkungen auf ein railML Dokument mit dieser Erweiterung kaum abschätzbar sind.

Was die Abbildung eines Sollgleises im Vergleich zum aktuell erfassbaren Ist-Gleis betrifft, wurde eine eher pragmatische Lösung gefunden, die sich allein auf diesen Sachverhalt konzentriert. Wir haben Dirks Bedenken dazu ebenfalls diskutiert, wie auch den Hinweis, dass das Sollgleis auch aus einer Betrachtung vorangegangener railML Dokumente ermittelt werden könnte, allerdings gilt das nicht für den Usecase, der der Anforderung der Aufnahme einer Sollgleismodellierung zugrunde liegt. Konkret geht es dabei um Fahrgastinformationssysteme, die jeweils nur den aktuellen Tag importieren und einen solchen Vergleich mit einem zuvor gelieferten Dokument nicht leisten können, da die entsprechenden Daten gegebenenfalls (noch) gar nicht importiert wurden.

Daher wurde ein Ansatz gewählt, der es erlaubt eine <originalTrackInfo> am ocpTT anzugeben. Dabei bieten sich die selben Modellierungsmöglichkeiten wie auch bei der Modellierung des Ist-Gleises bisher.

Eine entsprechende Implementierung ist isoliert im branch
<https://svn.railml.org/railML2/branches/trackchange/> zu reviewen.

Offen bleibt, wie mit @processStatus umgegangen werden soll. @Torben: gibt es nachdem wir den @pathStatus so angepasst haben, dass die verschiedenen Zustände eines Zuges im Kontext der Trassenbestellung berücksichtigt werden können, noch immer den Einspruch von JBD was das deprecated setzen des Attributs @processStatus betrifft? Ich persönlich würde dies nach wie vor als sinnvoll betrachten, da sich gezeigt hat, dass es keine standardisierbare Verwendung dieses Attributes gibt.

Best regards, Milan

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

Posted by [Milan Wölke](#) on Mon, 28 Sep 2020 16:38:59 GMT

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

Hi all,

in the last timetable developer group meeting we discussed how to go on with the remaining issue of this thread, which is, what to do with `@processStatus`. The developer group, including our colleagues from Norway agreed that given the other changes discussed here, there are no arguments anylonger preventing us from deprecating `@processStatus` as it is not used in a standardized fashion anyway. This change will be made available with version 2.5 of railML.

Best regards, Milan
