

---

Subject: [railML2] Extension proposal: pattern trains, distributions and slots

Posted by [Janne Möller](#) on Tue, 13 Oct 2020 12:32:40 GMT

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

---

Dear railML-community,

For the use case of long-term timetabling we added extensions to railML2.4 in the Norwegian railway sector (railML2.4nor) concerning pattern trains, that we would like to present to you.

A pattern train is a template for other trains. The container element `<nor:patternTrains>` is located directly under `<timetable>`. A `<nor:patternTrain>` is not itself a train that is supposed to run. It uses the same attributes and sub-elements as the element `<train>` and additionally the following newly introduced attributes: `@interval` (intervals in seconds between trains), `@trainsPerHour` (number of trains per normal traffic hour), `@trainsPerDay` (number of trains per normal traffic day), `@trainsPerWeek` and `@distributionRef` (reference to a more detailed distribution, as described below).

When the interval is fixed, it can be given as a single value, e.g. "600" for a pattern that will run every 10 minutes. If the interval varies in cycles, the cycle can be given as space-separated values, e.g. "600 900 300" for a pattern that will run with 10 minutes between the first and second departure, 15 minutes between the second and third and 5 minutes between the third and fourth, before the interval pattern repeats, in this case every 30 minutes. It is not required that the (sum of the) interval(s) evenly divides a whole hour, and a pattern that does not repeat at the whole hour carries over into the next. As an example, an interval of "2400", i.e. 40 minutes, would give a pattern that produces the same minutes of the hour every two hours.

A fixed number of departures per hour, day or week can be given using the respective attributes. As the number of departures per hour in a pattern can vary during a day, and similarly the number of departures per day over a week, we also needed a way to describe a more generalised distribution. The container element `<nor:distributions>` that includes any number of `<nor:distribution>` elements is placed directly under `<timetable>` in the schema. A `<patternTrain>` element can refer to a distribution using the attribute `@distributionRef`. In this way, one distribution can also be used for multiple pattern trains. With `<nor:distribution>`, the distribution of trains over the course of a time period such as one day can be described in a detailed and flexible way. For this, one or more `<nor:slot>` sub-elements are used. Each slot describes the number of trains (`@numberOfTrains`) in a certain time window (`@duration`) from a given starting time (`@startTime`).

Additionally, a `<trainGroup>` can have an attribute `@nor:patternTrainRef`, that refers to the `<nor:patternTrain>` functioning as a template for trains in that `<trainGroup>`.

Any feedback is highly appreciated.

Best regards,  
Janne Möller  
Jernbanedirektoratet

Subject: Re: [railML2] Extension proposal: pattern trains, distributions and slots

Posted by [Milan Wölke](#) on Thu, 15 Oct 2020 09:47:47 GMT

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

---

Hi Janne,

first of all thanks for your proposal.

I understand that you are interested in adding this to the standard in 2.5. We have discussed this in a first round in the timetable developer group and in general there was positive feedback regarding the idea of adding the possibility to model this kind of patterned approach. However there was also concern regarding the complexity of the chosen approach.

For me one major question is this, if the patternTrain has all the attributes and elements of a train (patternTrain is derived from train in your approach AFAIK), is the meaning of all those attributes and elements clear. For example, what is the meaning of a trainNumber that could be specified for a patternTrain. Same goes for the tafTapTsID that can be specified for a train and thus as well for a patternTrain?

I read in another post regarding the nor extensions that you introduced rules regarding the train numbers for trainGroups (trainNumbersFrom, trainNumbersTo - <https://www.railml.org/forum/index.php?t=msg&th=764&start=0&>), which can refer to patternTrains (as they already can refer to trains and patternTrain is intended to be a subclass of train). Wouldnt it make more sense to allow specifying a trainNumber pattern along with the patternTrain rather than providing them only with trainGroup?

Another feedback from the community was, that as the newly introduced complexity of this is rather high, could you provide some examples for this, especially some that refer to trainParts with different operatingPeriods, along with some explanation on how to interpret those?

Another question I came across is the meaning of the cancelled flag for a pattern train? Would cancelled=true mean that all trains described thus are cancelled? How would one model a scenario, where only one/some of the described trains need to be cancelled?

Would be great if you could help my clarifying those so we can better discuss the pros and cons of the modelling. Thanks in advance.

Best regards, Milan

---

Subject: Re: [railML2] Extension proposal: pattern trains, distributions and slots

Posted by [Thomas Nygreen JBD](#) on Thu, 26 Nov 2020 06:36:33 GMT

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

---

Dear Milan,

Dear all,

I gave a presentation at the TT telco on 12.11.2020, where I tried to answer some of the questions.

Regarding the elements inherited from <train>, not all are actually desired (such as trainNumber, tafTapTsilID, cancellation). Ideally the <train> and <patternTrain> extend a common parent class that contain the shared elements and attributes. In railML2.4 nor the closest alternative was to extend the existing eTrain class used for <train>. On page 4 of the presentation linked above, a possible redistribution of elements and attributes defined in the tTrain and eTrain classes is shown. The resulting eTrain class is identical to railML 2.4.

The suggested <patternTrain> class uses <trainPartSequence>s to describe the train path in the same manner as for <train>s. Christian Rößiger suggested in the telco to at least define clearly (e.g. in the wiki) how to interpret the properties of a <trainPart> depending on whether it is referenced from a <train> or a <patternTrain>. Possibly, a new type of train parts should be created and used by <patternTrain>s. Even though this increases the number of changes in the schema, it may narrow down the scope and reduce the complexity. One example is that we could skip using full <operatingPeriods> and use only <operatingDay>s instead.

I think the new train number attributes suggested for trainGroups were meant to be used with individual trains and as part of the yearly timetable process. So I'm not sure they fit on <patternTrain>. We do however already have a use case for some kind of train number pattern. In the different planning software I have used, it is normally given as an initial number (e.g. 501) and an increment (e.g. 2, resulting in 501, 503, 505, ...).

It has also been mentioned that the use of xs:list is uncommon in railML. That is true, but it is already used for @coord in <geoCoord>, and was chosen because it is simple, compact and readable. When using a schema-aware parser, the attribute value will be split automatically into a numeric integer array, so no text parsing by the receiving software should be necessary. As element values (e.g. <interval>1200</interval>) are not used in railML, the alternative would be something like

```
<intervals>
  <interval seconds="1200"/>
  <interval seconds="2400"/>
</intervals>
```

to replace interval="1200 2400". And in many (or most?) cases the value will be a single number.

Best regards,  
Thomas

---

Subject: Re: [railML2] Extension proposal: pattern trains, distributions and slots  
Posted by [Milan Wölke](#) on Wed, 09 Dec 2020 12:17:55 GMT  
[View Forum Message](#) <> [Reply to Message](#)

I like the idea of introducing a new base class for train and trainGroup in terms of the internal xsd

modelling. Could you provide a complete list of attributes that the patternTrain would need?

Regarding the new kind of trainPart I'm not yet convinced. I'm afraid this is gonna be far bigger than we anticipate now. I think it's well worth considering because the existing trainPart will most probably also allow specifying information that is not exactly clearly understood in the context of pattern trains. However, I would really appreciate it if you could provide the developer group with some kind of draft for the necessary modelling. We should at least be able to identify which information of a trainPart would be relevant and which wouldn't be (as a complete list). This would be necessary anyway even if we discard modelling a new kind of trainPart for patternTrains in order to properly document this in the wiki.

Best regards, Milan

---

Subject: Re: [railML2] Extension proposal: pattern trains, distributions and slots  
Posted by on Fri, 08 Jan 2021 18:36:58 GMT

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

Hello all,

I agree with Milan. In my opinion, the train parts referenced by <patternTrain>s should be a separate XML class, apart from the existing <trainPart>s.

If the existing <trainPart> were to be reused, we would have to document very extensively which attributes may be used in which use case and what their meaning is in each context.

By creating a separate class, we can provide for the new "<patternTrainPart>" exactly the attributes that are needed.

Another argument is that there are implementations of railML readers that are initially only interested in the <trainPart>s (not trains) of a railML file. If a railML file contained both "classic" <trainPart>s and new <patternTrainPart>s, it would be difficult to decide which type of <trainPart> it was.

As with Thomas' suggestion for modelling the <patternTrain>, I would prefer to create a common (internal) base class for the existing <trainPart> and the new <patternTrainPart> to reuse as much of the modelling as possible.

Best regards  
Christian

--

iRFP e. K. · Institut für Regional- und Fernverkehrsplanung  
Hochschulstr. 45, 01069 Dresden  
Tel. +49 351 4706819 · Fax. +49 351 4768190 · [www.irfp.de](http://www.irfp.de)  
Registergericht: Amtsgericht Dresden, HRA 9347

---

---

Subject: Re: [railML2] Extension proposal: pattern trains, distributions and slots

Posted by [Torben Brand](#) on Tue, 19 Jan 2021 08:58:43 GMT

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

---

Dear TT community,

Thank you for valuable input for extension of the pattern train. As suggested we at Jernbanedirektoratet are implementing a separate element for both <patternTrain> and <patternTrainPart> with it's own sub elements and attributes. We have selected attributes and sub-elements of <train> and <trainPart> that also fit the use case for pattern trains. The selected elements and attributes are mirrored 1:1 in the implementation of <patternTrain> and <patternTrainPart>, with the same descriptions and types. You find the description of the suggested model in the Word file (chapter 4.4 and 4.5), Illustration Png file (I'll also try to add it in the forum bellow as an image) and Excel file and in the railML cloud for the TT working group (folder /railML2.5/patternTrain). Unfortunately I'm not able to attach the files here in the Forum post reply, due to file attachment seems to be dissabled for me.

How to interpret the times (@arrival and @departure <times>) set in a pattern train relative to the rest of the trains in the group must be defined. We have 4 different alternatives. See also illustration in Word document.

Alternative 1: use <times> representing a possible instance of the pattern train.

In this alternative, <times> is used in the same manner as with normal <trainPart>s. The times used represent a train that can be formed from the pattern train, e.g. the first train of the resulting train group. The drawback with this alternative is that the times can be interpreted as an actually desired path, not just a template.

Alternative 2: use <times> with departure at midnight from first OCP

As alternative 1, but fix the departure from the first OCP to 00:00. The following arrival and departure times will then be interpreted as durations from the origin. The drawback with this alternative is that durations should use type xs:duration, not xs:time.

Alternative 3: do not use <times> and depend on <runTimes> and <stopTimes> for time differences per ocp

In this alternative, <runTimes> and <stopTimes> are used in the same manner as with normal <trainPart>s. This is also allowed in the other alternatives, but in this alternative it is the only information about arrival and departure times. The drawbacks with this approach are that it requires adding together all the time differences and that the separation between @minimalTime, @operationalReserve and @additionalReserve is not necessarily known.

Alternative 4: implement something new

This can for instance be an implementation similar to alternative 2, but with time shifts from the origin using xs:duration instead of timestamps using xs:time.

Subject: Re: [railML2] Extension proposal: pattern trains, distributions and slots  
Posted by [Burkhard Franke](#) on Mon, 25 Jan 2021 21:04:11 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Dear all

Sorry for asking (another) stupid question: Why do you need to include the pattern information in the export?

I guess you have a long-term planning tool and want to export the long-term timetable to other tools (simulation, short-term planning tool, ...). In such a usecase (and other usecases as well) it would be much easier if the exporting tool generates a series of "normal" trains based on the pattern trains. By doing so, the railML-export would be usable by all railML-capable tools.

In contrast, exporting "pattern" trains has two consequences/requirements:

- importing tools/interfaces have to be extended
- to make use of the imported "pattern" trains the importing tool must use a similar modelling. I can only think of long-term planning tools doing this.

So do you want to send a railML-file from one LTP tool to another LTP tool and vice versa? This is probably the only sensible usecase which needs "pattern" trains. Any other usecase would be better off exporting normal trains.

According to my taste this proposal is very much driven by "my tool uses a certain modelling approach and I want to include this approach in railML".

I think one should pay more attention to the point of view of the importing tools. Any sophisticated changes to the railML-standard is a wasted effort if the same data could be written in the conventional format. The more sophisticated the format, the closer this is to a 1:1 interface between two dedicated tools.

Or am I wrong with my assumptions?

Best regards

Burkhard

---

---

Subject: Re: [railML2] Extension proposal: pattern trains, distributions and slots  
Posted by [Thomas Nygreen JBD](#) on Tue, 26 Jan 2021 23:40:43 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Dear Burkhard,

The use case is to exchange the service definitions that a timetable can be or has been built from, either in combination with that timetable or by itself. It is part of the use case that the timetable may not (yet) exist, and in those cases, it would be difficult to exchange the non-existing trains. After the timetable has been created, the service definitions are still useful, and can be linked to the resulting "normal" trains through train groups with the patternTrainRef attribute. Any reading

software that only concerns itself with "normal" trains can ignore the pattern train definitions.

Best regards,  
Thomas

---

---

Subject: Re: [railML2] Extension proposal: pattern trains, distributions and slots  
Posted by [Thomas Nygreen JBD](#) on Tue, 26 Jan 2021 23:42:52 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Dear all,

I have committed a draft implementation as revision 1083 in the SVN repository. The matching TT documentation with the new elements can also be browsed in the repository.

Best regards,  
Thomas

---

---

Subject: Re: [railML2] Extension proposal: pattern trains, distributions and slots  
Posted by [Milan Wölke](#) on Mon, 08 Feb 2021 15:56:09 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hi all,

in response to the question Torben posted regarding how to model times for a pattern train the timetable developer group decided at their last meeting to follow up with alternative 1. So times would be specified the same way they are for a normal trainPart.

Best regards, Milan

---

---

Subject: Re: [railML2] Extension proposal: pattern trains, distributions and slots  
Posted by [Thomas Nygreen](#) on Tue, 23 Feb 2021 05:39:12 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Dear all,

I have updated the patternTrain implementation, including changes after feedback from the community:

same names for descendants of patternTrainPart as trainPart

same <times> for patternTrainPart as trainPart (but not allowing scope="actual")

removed @numberOfTrainsPer\* from patternTrain (always use distributions)

moved @interval from patternTrain to distribution/slot

removed default for slot/@startTime and made slot/@duration and slot/@numberOfTrains

required

added reference OCP for distribution



As before, the schema files are available in the patternTrain branch in the SVN repo, along with HTML documentation. The change log can be browsed in Trac if logged in, or using your favourite SVN client.

I have now also added two very simple examples. The first one shows a few options for how to use <distribution> and how to link a patternTrain with the individual resulting trains. The second example shows two patternTrains operating on the same line, one as a regular service and one shorter express service only during rush hours. This second file has no individual trains.

I had thought the easiest way to link the two patternTrains together, to form one service with an increased frequency during rush hours, would be to use trainGroups with the proposed @parentRef attribute. However, a trainGroup must have at least one (individual) train. One option is to also have a @parentRef on patternTrain, to allow a hierarchy of pattern trains. I see another option, which is to move @distributionRef from patternTrain to patternTrainPart, to work in a similar manner as @operatingPeriodRef for normal trainParts. Then both patternTrainParts can be referenced from the same patternTrain.

I welcome any and all input!

Best regards,  
Thomas

---

---

Subject: Re: [railML2] Extension proposal: pattern trains, distributions and slots  
Posted by [Milan Wölke](#) on Wed, 24 Feb 2021 10:54:08 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hi Thomas,

Thanks for the update. From my POV the changes to patternTrain and distribution really improve the modelling here. I also think the examples are a good idea to give an idea on how this could be used. I think we should include those in the wiki documentation as well.

Regarding the linking of patternTrains, I dont quite get the point. Wouldnt it be possible to specify a train service with increased frequency during rushhour using different slots of the distribution as well? Or is it that for ur use case the train path would need to be different in rush hours? If we simply move distributionRef down to patternTrainPart then I guess we would also need to come up with a new documentation of (pattern)TrainPartSequence, dont we?

Best regards, Milan

---