
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
