

Hello Tuomas,

- > Departure is concept in our system that knows following things:
- > trainnumber, vehicle type, route (route is ordered list of stations ~
- > OCPsTT), departureTime, other driving times (times when it arrives to
- > other stations)

This is like what we call in railML a trainPart.

A train (either "operational" or "commercial") is a combination of such trainParts with a certain trainnumber. Such a train could change part of its vehicles on an intermediate station A "commercial" train with its trainNumber could be shown by a passenger information or in a public timetable.

- > Departure 1 goes route: city1-city2-city3
- > Departure 2 goes route: city3-city4-city5
- > Departure 3 goes route: city3-city4-city1
- > Departure 4 goes route: city5-city1-city2

The possible "next departures" in city3 in your sense could be either a "connection" in railML. Then its meant like a passenger information in city3 : "Passengers for city5 should change to the train Departure 5".

Or maybe its like in the planning process for a rostering. Then the vehicles of Departure1 could be further used in Departure2 or Departure3. Then your departures are corresponding to blockParts which are referencing trainParts. The result of the decision process in your programm (take Departure2 after Departure1) will lead to a "block" in railML that is used as part of a vehicle circulation.

Anyway, all possible trainParts could be listed in railML and they don't know about each other. The chosen connection (for passengers -> conection or vehicles -> block) between trainParts is the result of a planning process.

The two bitmasks are not in competition but two different models. If you have an operational system, you are familiar with the operatingperiod->bitmask for every day. Other systems for conceptional planning purposes deal with a standard week and the operatingperiod->operatingday->operatingcode->bitmask.

Kind regards,  
Joachim

Tuomas Tiihonen wrote:

>  
>  
>> I'm not sure if I understand right, what you mean by your "Departure".  
>> Could you please describe this a little bit more in detail?  
>  
> Departure is concept in our system that knows following things:  
> trainnumber, vehicle type, route (route is ordered list of stations ~  
> OCPsTT), departureTime, other driving times (times when it arrives to  
> other stations) AND next possible departures. So it is one train that goes  
> around some route with specified times and with specified vehicle with  
> unique train number. It sounds something like commercial train in RailML?  
>  
> And the question was that, when one of such departures has ran from the  
> beginning of the route to the end of the route it is time to make decision  
> about the next departure.  
> Example:  
> Departure 1 goes route: city1-city2-city3  
> Departure 2 goes route: city3-city4-city5  
> Departure 3 goes route: city3-city4-city1  
> Departure 4 goes route: city5-city1-city2  
> Train 1 has ran departure 1 and are now in city 3. Now choice has to be  
> made if next departure is departure 2 or 3 (both starts from city 3 and  
> departure time is near the current time). Departure 4 is not one of the  
> choices as it is not starting from city 3. The departure 1 knows the list  
> of possible next departures (departure 2 and 3 in the example).  
>  
> If all this is applied to RailML can you consider this:  
> Is the commercial train equivalent to Departure in RailML?  
> If the train is equivalent how can one train get list of next trains  
> (=next departures)? in RailML?  
>  
>  
>>> operatingperiod->bitmask  
>> This is a bitmask for every day of a timetable period, describing if the  
>> train is running on this specific day.  
>>  
>>> operatingperiod->operatingday->operatingcode->bitmask  
>> This is a different more generic way of describing, like "running Mondays  
>> to Fridays only" with a week based bitmask. This is valid for any week  
>> with some further described deviances.  
>  
> Can you please clarify the relations of the bitmasks. Which one overrides  
> which?  
>  
>  
>> I hope this will clear up intentions behind the complex structures of  
>> railML a little bit.

>  
> Thank you for the clarifications so far, great help!  
>  
> Sincerely,  
> Tuomas Tiihonen  
>  
>

--  
----- posted via PHP Headliner -----

---