

Greetings to RailML community!

This is my first post to the forum so I will start with a introduction. I am a Software Design Engineer with strong Java/C++ programming/design background. I am currently working for Finnish company called Mitron Oy. We have headquarters in Forssa/Finland and other offices in Tampere/Finland, Mittenaar/Germany and Warszawa/Poland. Mitron focuses on passenger information, display, entertainment, announcement and security systems for trains, trams, subways, stops, stations and platforms. More information about company can be found from www.mitron.com and I am happy to answer further queries about me or the company.

Within Mitron we have ongoing discussion about RailML and I have now been studying it from technical perspective. Goal of this study is to make decision about our commitment to RailML and what our role would be.

During this technical investigation I have had some difficulties related to the semantic specification explained (or more accurately not explained) in the RailML wiki pages.

I have so many questions about the semantics, but I have to start from somewhere so here it goes:

We have thing called "Departure" which I think is close to timetable->trains->train in RailML. Our departure knows route and timetable for the route for example. Departure knows also list of possible next Departures that might come next from the terminal station of first departure. What would be the place in RailML to get that information? Is the circulations/rostering/blocks semantically identical to this? Does the block mean part of train or part of track as an example? I have tried to figure out the semantical relations of those mentioned RailML elements, but without documentation in wiki it has proved difficult.

Other thing I don't quite get, even though it is mentioned in wiki is relation between operatingperiod->operatingday->operatingcode->bitmask and operatingperiod->bitmask and operatingdaydeviance and holiday and specialservice. Which one overrides which? Why there are period bitmask separately from week bitmask and then deviances and holidays?

What is ocpTT->sectionTT->distance. Distance from where to where? Is this in relation to infrastructures track->tracktopology->trackBegin/trackEnd->pos -attribute? If I would like to know distance of two stations along the track/line what is the correct place?

With Kindest Regards,
Mr. Tuomas Tiihonen

--

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

Subject: Re: RailML semantics, nextdeparture, recurringschedule
Posted by [Joachim Rubröder railML](#) on Tue, 10 May 2011 08:28:13 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear Tuomas,

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?

Here is some rough explanation about the railML basics you mentioned for clarification:

> timetable->trains->train

If your dealing with passenger information, your "train" will probably be with type="commercial" and correspond to the train, a passenger is used to. A train with type="operational" in comparance is a train considered by a signal box, which is mostly but not necessary the same kind of train.

> circulations/rostering/blocks

The blocks are describing the routine of the day for a specific piece of rollingstock. This could be within different trains and could include empty rides before starting the routine for the following day which is described in the next block.

> operatingperiod->bitmask

This is a bitmask for every day of a timetable period, decribing 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.

> ocpTT->sectionTT->distance

Is the running distance between one ocpTT within the path of a train and the next ocpTT for this train. This information is redundand and could be calculated by

> track->tracktopology->trackBegin/trackEnd->pos

wich is the correct length of a single track between two switches in a microscopic view.

But if you consider track/Begin/trackEnd as macroscopic nodes corresponding to stations, then you will get the same result:

- > ocpTT->sectionTT->distance (running distance for the train between stations)
- > track->tracktopology->trackBegin/trackEnd->pos (macroscopic infrastructure distance between stations)

I hope this will clear up intentions behind the complex structures of railML a little bit.

Kind regards
Joachim

Tuomas Tiihonen wrote:

- >
- > Greetings to RailML community!
- >
- > This is my first post to the forum so I will start with a introduction. I
- > am a Software Design Engineer with strong Java/C++ programming/design
- > background. I am currently working for Finnish company called Mitron Oy.
- > We have headquarters in Forssa/Finland and other offices in
- > Tampere/Finland, Mittenaar/Germany and Warszawa/Poland. Mitron focuses on
- > passenger information, display, entertainment, announcement and security
- > systems for trains, trams, subways, stops, stations and platforms. More
- > information about company can be found from www.mitron.com and I am happy
- > to answer further queries about me or the company.
- >
- > Within Mitron we have ongoing discussion about RailML and I have now been
- > studying it from technical perspective. Goal of this study is to make
- > decision about our commitment to RailML and what our role would be.
- >
- > During this technical investigation I have had some difficulties related
- > to the semantic specification explained (or more accurately not explained)
- > in the RailML wiki pages.
- >
- > I have so many questions about the semantics, but I have to start from
- > somewhere so here it goes:
- >
- > We have thing called "Departure" which I think is close to
- > timetable->trains->train in RailML. Our departure knows route and
- > timetable for the route for example. Departure knows also list of possible
- > next Departures that might come next from the terminal station of first
- > departure. What would be the place in RailML to get that information? Is
- > the circulations/rostering/blocks semantically identical to this? Does the
- > block mean part of train or part of track as an example? I have tried to
- > figure out the semantical relations of those mentioned RailML elements,
- > but without documentation in wiki it has proved difficult.
- >

> Other thing I don't quite get, even though it is mentioned in wiki is
> relation between operatingperiod->operatingday->operatingcode->bitmask and
> operatingperiod->bitmask and operatingdaydeviance and holiday and
> specialservice. Which one overrides which? Why there are period bitmask
> separately from week bitmask and then deviances and holidays?
>
> What is ocpTT->sectionTT->distance. Distance from where to where? Is this
> in relation to infrastructures
> track->tracktopology->trackBegin/trackEnd->pos -attribute? If I would like
> to know distance of two stations along the track/line what is the correct
> place?
>
> With Kindest Regards,
> Mr. Tuomas Tiihonen
>

--

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

Subject: Re: RailML semantics, nextdeparture, recurringschedule
Posted by [tuomas.tiihonen](#) on Tue, 10 May 2011 11:52:48 GMT
[View Forum Message](#) <> [Reply to Message](#)

> 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 -----

Subject: Re: RailML semantics, nextdeparture, recurringschedule

Posted by [Joachim Ruböder railML](#) on Tue, 10 May 2011 12:51:11 GMT

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

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 -----

Hi,

- > 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.

Our Next departure is used by the driver of the train. So after driver finishes one departure the driver can choose next departure for the train he/she is currently driving. If I understand correctly it is perhaps the vehicle->block closer to this.

- > 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.

Oh, ok. That clarifies it a bit. We have to choose which way we have to define the days in which context.

Thank you again for the answers!

Br,
Tuomas

- > 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 -----

Subject: Re: RailML semantics, nextdeparture, recurringschedule
Posted by [Joachim Rubröder railML](#) on Wed, 11 May 2011 08:09:34 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Tuomas,

> Our Next departure is used by the driver of the train. So after driver
> finishes one departure the driver can choose next departure for the train
> he/she is currently driving. If I understand correctly it is perhaps the
> vehicle->block closer to this.

now we're getting closer. If the train driver chooses the next departure
its probably kind of a service plan for train drivers. This is not yet
implemented in railML but very similar to the rostering structure, wich is
a service plan for vehicles. The problem with the train drivers is that
they are much more complex (different skills, payment, pause times, ...) to
deal with.

Sincerely,
Joachim

T wrote:

>
> Hi,
>
>> 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 -> connection
>> or vehicles -> block) between trainParts is the result of a planning
>> process.
>
> Our Next departure is used by the driver of the train. So after driver
> finishes one departure the driver can choose next departure for the train
> he/she is currently driving. If I understand correctly it is perhaps the
> vehicle->block closer to this.
>
>> 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 conceptual
>> planning purposes deal with a standard week and the
>> operatingperiod->operatingday->operatingcode->bitmask.
>
> Oh, ok. That clarifies it a bit. We have to choose which way we have to
> define the days in which context.
>
>
> Thank you again for the answers!
>
> Br,
> Tuomas
>
>
>
>> 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 -----

Subject: Re: RailML semantics, nextdeparture, recurringschedule
Posted by [tuomas.tiihonen](#) on Wed, 11 May 2011 12:01:19 GMT
[View Forum Message](#) <> [Reply to Message](#)

Joachim RubrÄ¶der wrote:

>
> Hi Tuomas,
>
>> Our Next departure is used by the driver of the train. So after driver
>> finishes one departure the driver can choose next departure for the train
>> he/she is currently driving. If I understand correctly it is perhaps the
>> vehicle->block closer to this.
>
> now we're getting closer. If the train driver chooses the next departure
> its probably kind of a service plan for train drivers. This is not yet
> implemented in railML but very similar to the rostering structure, wich is
> a service plan for vehicles. The problem with the train drivers is that
> they are much more complex (different skills, payment, pause times, ...)
> to deal with.
>

Hi Joachim,

actually the next departure is not related to the driver in our system.
Driver is just the user who selects the next departure from the system. So
in this context there is no need to know any information about the driver,
or even who is the driver or even if there is the driver :)

As driver is the user of the information he/she needs to know the next
departures for the train he/she is sitting in not for driver itself.

You can imagine that the train intelligently answers driver's question:
"Dear train, please tell me all possible next departures where you, train,
can depart from this exact station we are currently at"

After this question train would generate this information, hopefully from
railml by searching next departures from last driven departure. Or in
railml next block from last driven block?

Br,
Tuomas

--

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

Subject: Re: RailML semantics, nextdeparture, recurringschedule
Posted by [tuomas.tiihonen](#) on Wed, 11 May 2011 12:01:24 GMT

Joachim Rubröder wrote:

>
> Hi Tuomas,
>
>> Our Next departure is used by the driver of the train. So after driver
>> finishes one departure the driver can choose next departure for the train
>> he/she is currently driving. If I understand correctly it is perhaps the
>> vehicle->block closer to this.
>
> now we're getting closer. If the train driver chooses the next departure
> its probably kind of a service plan for train drivers. This is not yet
> implemented in railML but very similar to the rostering structure, wich is
> a service plan for vehicles. The problem with the train drivers is that
> they are much more complex (different skills, payment, pause times, ...)
> to deal with.
>

Hi Joachim,

actually the next departure is not related to the driver in our system.
Driver is just the user who selects the next departure from the system. So
in this context there is no need to know any information about the driver,
or even who is the driver or even if there is the driver :)

As driver is the user of the information he/she needs to know the next
departures for the train he/she is sitting in not for driver itself.

You can imagine that the train intelligently answers driver's question:
"Dear train, please tell me all possible next departures where you, train,
can depart from this exact station we are currently at"

After this question train would generate this information, hopefully from
railml by searching next departures from last driven departure. Or in
railml next block from last driven block?

Br,
Tuomas

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

Subject: Re: RailML semantics, nextdeparture, recurringschedule
Posted by [Joachim Rubröder railML](#) on Wed, 11 May 2011 14:58:09 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear Tuomas,
this is a very strange and interesting concept.

A passenger would ask a timetable: "When leaves the next train from here to xy?"

(-> connection)

A train driver could ask the vehicle: "Where are you supposed to go next?"

(-> blocks)

Anyone could ask a railway map: "Where could I go from this station?"

(-> infrastructure)

A commercial train could be asked: "What is published as your next station?"

(-> train)

A train driver could ask the train: "What is our next station in the timetable?"

(-> train)

What is the use case for the choice?

Why is the train driver asking such a question?

Train drivers and vehicles usually have their fixed service plans and the physical trains have their fixed timetable to fulfill - there is no choice.

Your train drivers question sounds like a taxi driver asking about possible further rides. Maybe it is the train asking "where shall we go on?" by providing a list of possibilities (-> operational trains) with a filter function. Then the train driver would choose the correct one, because he knows it. But I would assume that a train driver would type in the train number and the train searches in a given in a timetable where to go next.

Kind regards,
Joachim

Tuomas Tiihonen wrote:

>

> Joachim RubrÄ¶der wrote:

>>

>> Hi Tuomas,

>>

>>> Our Next departure is used by the driver of the train. So after driver
>>> finishes one departure the driver can choose next departure for the train
>>> he/she is currently driving. If I understand correctly it is perhaps the
>>> vehicle->block closer to this.

>>

>> now we're getting closer. If the train driver chooses the next departure
>> its probably kind of a service plan for train drivers. This is not yet
>> implemented in railML but very similar to the rostering structure, wich is

>> a service plan for vehicles. The problem with the train drivers is that
>> they are much more complex (different skills, payment, pause times, ...)
>> to deal with.
>>
>
> Hi Joachim,
>
> actually the next departure is not related to the driver in our system.
> Driver is just the user who selects the next departure from the system. So
> in this context there is no need to know any information about the driver,
> or even who is the driver or even if there is the driver :)
>
> As driver is the user of the information he/she needs to know the next
> departures for the train he/she is sitting in not for driver itself.
>
> You can imagine that the train intelligently answers driver's question:
> "Dear train, please tell me all possible next departures where you, train,
> can depart from this exact station we are currently at"
>
> After this question train would generate this information, hopefully from
> railml by searching next departures from last driven departure. Or in
> railml next block from last driven block?
>
> Br,
> Tuomas
>

--

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

Subject: Re: RailML semantics, nextdeparture, recurringschedule

Posted by [tuomas.tiihonen](#) on Thu, 12 May 2011 13:22:08 GMT

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

Hi Joachim,

Joachim Rubröder wrote:

> Your train drivers question sounds like a taxi driver asking about
> possible further rides. Maybe it is the train asking "where shall we go
> on?" by providing a list of possibilities (-> operational trains) with a
> filter function. Then the train driver would choose the correct one,
> because he knows it. But I would assume that a train driver would type in
> the train number and the train searches in a given in a timetable where to
> go next.

Heureka! Sorry for misleading you. I think that is closest so far. That the `_train_` not the driver is asking "where shall we go on?" And the driver then answers, by either inputting trainnumber, or choose from given `_list of possibilities_` or search. After the train gets its answer (Departure), it will use that information to display e.g. route correctly in the displays.

This matches our concept, except this is thought in commercial perspective rather than operational.

So how the commercial train would know the next commercial train that leaves the end station? Is that only by crawling through list of trains or is there ordered list (under rostering, circulation?)..

Br,
Tuomas

--

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

Subject: Re: RailML semantics, nextdeparture, recurringschedule
Posted by [Joachim Rubröder railML](#) on Fri, 20 May 2011 08:08:59 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Tuomas,

> Heureka! Sorry for misleading you. I think that is closest so far. That
> the `_train_` not the driver is asking "where shall we go on?" And the
> driver then answers, by either inputting trainnumber, or choose from given
> `_list of possibilities_` or search. After the train gets its answer
> (Departure), it will use that information to display e.g. route correctly
> in the displays.

OK, let us assume there is a railML timetable (on the system in the train) and the train knows where it is (station xy). The train needs to know where to go for displaying (->commercial perspective).

- the train driver types in the trainNumber (train searches the related commercial "train" according to the trainNumber)
- train lists all "trainParts" starting at xy and train driver chooses one then the train searches the related commercial train according to the trainPart references)

The timetable gives a full overview over all trains. There is no special list sorted by stations. So it is necessary to search the list of trains or trainParts.

Kind regards,
Joachim

Tuomas Tiihonen wrote:

>
> Hi Joachim,
>
> Joachim RubrÄ¶der wrote:
>> Your train drivers question sounds like a taxi driver asking about
>> possible further rides. Maybe it is the train asking "where shall we go
>> on?" by providing a list of possibilities (-> operational trains) with a
>> filter function. Then the train driver would choose the correct one,
>> because he knows it. But I would assume that a train driver would type in
>> the train number and the train searches in a given in a timetable where to
>> go next.
>
> Heureka! Sorry for misleading you. I think that is closest so far. That
> the _train_ not the driver is asking "where shall we go on?" And the
> driver then answers, by either inputting trainnumber, or choose from given
> _list of possibilities_ or search. After the train gets its answer
> (Departure), it will use that information to display e.g. route correctly
> in the displays.
>
> This matches our concept, except this is thought in commercial perspective
> rather than operational.
>
> So how the commercial train would know the next commercial train that
> leaves the end station? Is that only by crawling through list of trains or
> is there ordered list (under rostering, circulation?)..
>
> Br,
> Tuomas
>
>
>

--

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

Subject: Re: RailML semantics, nextdeparture, recurringschedule
Posted by [tuomas.tiihonen](#) on Mon, 23 May 2011 05:46:19 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Joachim,

I think this answers my question, thank you! Thank you also for the patience :)

Br,
Tuomas

>> Heureka! Sorry for misleading you. I think that is closest so far. That
>> the _train_ not the driver is asking "where shall we go on?" And the
>> driver then answers, by either inputting trainnumber, or choose from given
>> _list of possibilities_ or search. After the train gets its answer
>> (Departure), it will use that information to display e.g. route correctly
>> in the displays.

>

> OK, let us assume there is a railML timetable (on the system in the train)
> and the train knows where it is (station xy). The train needs to know
> where to go for displaying (->commercial perspective).
> - the train driver types in the trainNumber (train searches the related
> commercial "train" according to the trainNumber)
> - train lists all "trainParts" starting at xy and train driver chooses one
> then the train searches the related commercial train according to the
> trainPart references)

>

> The timetable gives a full overview over all trains. There is no special
> list sorted by stations. So it is necessary to search the list of trains
> or trainParts.

>

> Kind regards,
> Joachim

--

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