Subject: railML 2.3 infrastructure extension proposal - a new "state"
Posted by Torben Brand on Thu, 31 Aug 2017 10:57:29 GMT
View Forum Message <> Reply to Message

There is an existing element <state> in railML2.3 that applies to certain selected elements. Jernbanedirektoratet and Bane NOR use case needs to extend the features of the state and make it more flexible. Unfortunately the current <state> element it is not extendable and the structure as an element seems too complex and rigid for our need. We thus suggest to add a new Norwegian extension attribute to our extension railML2.3nor which can be placed on any element and is valid for that element and all sub elements (inheritance) within the element.

Placing the attribute nor:state
The new attribute @nor:state can be placed on any element. It is valid for that element and all sub elements (inheritance) within the element. Inheritance can be broken by defining a new state on an element.
For instance the attribute @nor:state can be placed on <infrastructure id="" name="" nor:state="operational">
For railML files containing multiple planned states with states spanning over more than one ocp area we suggest to first define an <infrastructure with nor:state="operational" and then one or multiple <infrastrucure with nor:state="planned" with their corresponding operatingPeriodRef="id".
For smaller planned changes within an ocp the nor:change attribute can be placed on a planned track within the same infrastructure. Even smaler changes within one track can be placed on the element(s) itself.
The planned tracks must have connections to the operational tracks over the <connection> element.
If no nor:state is not defined the value is by default "operational"

Values of nor:state
We need the values: operational/planned/disabled/closed/historic.
operational: the described infrastructure is operational
planned: the infrastructure is planned in the future.
disabled: the infrastructure is still in place, but operationally disabled.
removed: vital or all elements of the infrastructure is removed but juridical rights are in place
closed: The infrastructure is still in place but juridical rights have been removed
historic: all elements of the infrastructure is removed and juridical rights have been removed.

The timeframe of the defined states are defined in the existing element <timetable/operationalPeriods/operationalPeriode> @startDate and @endDate with an attribute in the infrastructure @operatingPeriodRef refering to it.

Subject: Re: railML 2.3 infrastructure extension proposal - a new "state"
Posted by christian.rahmig on Fri, 01 Sep 2017 12:32:03 GMT
View Forum Message <> Reply to Message

Dear Torben,

thank's for opening this big topic that needs to be solved with railML v3 for sure...

Am 31.08.2017 um 12:57 schrieb Torben Brand:

> There is an existing element <state> in railML2.3 that
> applies to certain selected elements. Jernbanedirektoratet
> and Bane NOR use case needs to extend the features of the
> state and make it more flexible. Unfortunately the current
> <state> element it is not extendable and the structure as an
> element seems too complex and rigid for our need. We thus
> suggest to add a new Norwegian extension attribute to our
> extension railML2.3nor which can be placed on any element
> and is valid for that element and all sub elements
> (inheritance) within the element.

Currently, railML v2 already provides states (of operation) for all trackElements, ocsElements and additionally for lines, tracks and locallyControlledAreas. The state is modelled as sub element <state> in a container <states> and it is used for marking infrastructure components that are not usable for operation. The current model lacks further details on different states and it is not possible to define the state of the whole infrastructure. With your proposal you want to bridge these gaps, but the solution is not that easy...

> Placing the attribute nor:state
> The new attribute @nor:state can be placed on any element.
> It is valid for that element and all sub elements
> (inheritance) within the element. Inheritance can be broken
> by defining a new state on an element.
> For instance the attribute @nor:state can be placed on
> <infrastructure id="" name="" nor:state="operational">

Using an attribute instead of an object leaves you very little space for further modelling: An attribute provides one single information while the sub element can have multiple further attributes for specifying the state in detail. In particular, the <state> currently allows for referencing a timespan of validity using its attribute @operatingPeriodRef. Or you can define a limited part of a track as being disabled (e.g. for construction) using the <state> sub elements <from> and <to>. All this is not possible, if you have only one single attribute @state.

> For railML files containing multiple planned states with
> states spanning over more than one ocp area we suggest to
> first define an <infrastructure with nor:state="operational"
> and then one or multiple <infrastrucure with
> nor:state="planned" with their corresponding
> operatingPeriodRef="id".

Having multiple <infrastructure> elements of different states sounds good at first sight. However, this approach runs into a major problem: Currently, the railML schema does not allow to having several <infrastructure> elements in one file. If you want to distinguish between different states of infrastructure on top level, you have to have multiple railML files. We can see these different files as different "scenarios" for the infrastructure network.

Thinking about this top level state information, I assume that it may be useful to have the <state> element also available directly under <infrastructure>. This would us allow to recognize the difference between the different railML files and their included "infrastructure scenarios" without having to look in detail into them. So, probably a ticket for railML 2.4?

> For smaller planned changes within an ocp the nor:change
> attribute can be placed on a planned track within the same
> infrastructure. Even smaler changes within one track can be
> placed on the element(s) itself. The planned tracks must have
> connections to the operational
> tracks over the <connection> element.

That is another point to think about in detail: A complete operational railway network will have tracks that are connected with each other and that have defined borders (like buffer stops). It is not connected with possible planned tracks (since there is no physical combination). So, currently the only option is to define multiple railML files with different infrastructure scenarios that are valid at different points in time. A logical linking between both scenarios is currently (railML v2) not possible. In my opinion, railML v3 should solve this problem.

> If no nor:state is not defined the value is by default
> "operational"

That's how railML works already right now. Maybe, we should add this clarification in the railML Wiki.

> Values of nor:state
> We need the values:
> operational/planned/disabled/closed/historic. operational: the described
> infrastructure is operational
> planned: the infrastructure is planned in the future. disabled: the
> infrastructure is still in place, but
> operationally disabled. removed: vital or all elements of the
> infrastructure is
> removed but juridical rights are in place
> closed: The infrastructure is still in place but juridical

> rights have been removed
> historic: all elements of the infrastructure is removed and
> juridical rights have been removed.

I cross-checked your proposed values with OpenStreetMap. There, they define the following states:

- construction (corresponds to your "planned")
- disused (corresponds to your "disabled" and "closed")
- abandoned (corresponds to your "historic" and "removed")

There is no specific state for "operational" since all elements that are not explicitly defined differently, are seen as "operational".

However, there exists also a concept for a more detailed description of infrastructure lifecycle, see [1]

Concluding the various approaches, I want to ask you, if you would be happy if the following values would be available:

- plan
- construction
- operation
- disabled
- removed
- historic

(Do you see "closed" really as a longer lasting state or isn't it a transition state from "disabled" to "historic"?)

Once, we could agree on a final list of states, I would take it into a ticket for railML v3.

> The timeframe of the defined states are defined in the
> existing element
> <timetable/operationalPeriods/operationalPeriode> @startDate
> and @endDate with an attribute in the infrastructure
> @operatingPeriodRef refering to it.

Yes, this part of the solution exists already.

To conclude: the problem of infrastructure states can be fully solved only with railML v3. For railML v2 (2.4) we may think of updating the list of different states and of adding the <state> element on top level <infrastructure>.

@all: Please let me know your ideas on the topic, especially when you are interested as well in an improved infrastructure state modelling on the basis of railML v2.4.

Thank you very much and best regards

Christian

[1] https://wiki.openstreetmap.org/wiki/Lifecycle_prefix

--
Christian Rahmig - Infrastructure scheme coordinator
railML.org (Registry of Associations: VR 5750)
Phone Coordinator: +49 173 2714509; railML.org: +49 351 47582911
Altplauen 19h; 01187 Dresden; Germany   www.railml.org

Subject: Re: railML 2.3 infrastructure extension proposal - a new "state"
Posted by Torben Brand on Thu, 07 Sep 2017 08:29:58 GMT

View Forum Message <> Reply to Message

Thank you Christian for an extremely quick answer

To summarize (as I read you answer):
1. time dimension only possible partially in railML 2.4 and fully in railML3
2. suggested railML 2.3nor extension is limited
3. not possible to use more than one <infrastructure> in railML 2.3

Torbens reply:
1. I see a limited, but sufficient possibility for time dimension within railML2.3nor extension. Put's big responsibility on source and import application for data consistency.
2. I agree. But it fullfils our use case.
3. OK

I have made some solution scenarios for railML2.3nor/NorRailView/Trase below:

How to integrate different states in one data set
There are different ways to create a data set with different states with a time dimension in railML2.3nor. I would like to hear you opinions.
Alternative 1 "no" time dimension time dimension in the metadata
A railML data set only handles one time dimension at a time (snap-shot).
There can be complete snap shots (1A) or delta snap shots (1B). Complete snap shots could for instance be a data set with a complete national network today and another data set with a complete national network planned at a certain date.
Delta snap shots would be first a data set with a complete national network today and then delta data sets of individual planned projects.
The challenge with the first is that the distinctions of the changes disappear together with their the time dimension. The project distinctions could somewhat be kept with the use of <border> with @name and @description for the project name and validity date.
The challenge with the latter is to merge the projects into the existing network. The tracks cannot be connected as the id's could be inconsistent in the different data sets. But implicit information is available to place the project infrastructure on the existing infrastructure through the Norwegian consistent use of the values <line>@name,<absPos> and <track>@name

Alternative 2 - with time dimension
Use time dimension as suggested with use @nor:state and @operatingPeriodRef. There exist different sub alternatives for placement.

Alternative 2A - Integrated
As suggested use @nor:state and @operatingPeriodRef and place them on individual elements in the structure for individual element (planned) states. Place them on a separate track with (planned) state if the @state spans an entire track. All elements on the track inherit the change. For states that span more than one ocp the state info is placed on the <infrastructure> element. There is a challenge using more than one <infrastructure>. The wiki says "An <infrastructure> defines the header container for various infrastructure versions." So multiple <infrastructures> should be allowed. But the XSD defines maximum of one occurrence of <infrastructure>. Can we ignore this?
The reason we would like to use <infrastrucure> for different states is that having multiple time dimensions in one <infrastructure> under either separate <track>s with @nor:state and @operatingPeriodRef or even as loose elements in the structure itself can possible create inconsistencies if not kept completely cleanly structured.
Alternative 2A1 multiple <infrastructure> Alternative 2A2 with everything in one <infrastructure>

Alternative 2B - Semi integrated time dimension time dimension in the metadata
A work around would be to create one data set from one source application with multiple railML files zipped together. The individual railML files would contain <infrastrucutres> for each time dimension. The id's and the track connections between the infrastuctures would have to be created consistent. Loading the files together as one zip creating one model with time dimension.

Based on what Christian writes and previous feedback I suggest the following:

Alternatives 2A1 is not valid.
We use alternative 1A for simple transfer between the systems. More complex systems with DB handling of scenarios can export either 1A, 1B or 2B depending on user choice.
Alternative 2A2 seems quite complex... And should be left for railML3
I have made a simple illustration:

What are your opinions?

Subject: Re: railML 2.3 infrastructure extension proposal - a new "state"
Posted by christian.rahmig on Mon, 11 Sep 2017 07:36:03 GMT

View Forum Message <> Reply to Message

Dear Torben,

let me briefly comment on your remarks and questions:

Am 07.09.2017 um 10:29 schrieb Torben Brand:

> To summarize (as I read you answer):
> 1. time dimension only possible partially in railML 2.4 and
> fully in railML3
> 2. suggested railML 2.3nor extension is limited
> 3. not possible to use more than one <infrastructure> in
> railML 2.3

Correct.

> [...]
>
> How to integrate different states in one data set
> There are different ways to create a data set with different
> states with a time dimension in railML2.3nor. I would like
> to hear you opinions.
> Alternative 1 "no" time dimension time dimension in the
> metadata
> A railML data set only handles one time dimension at a time
> (snap-shot). There can be complete snap shots (1A) or delta snap shots
> (1B). Complete snap shots could for instance be a data set
> with a complete national network today and another data set
> with a complete national network planned at a certain date.
>
> Delta snap shots would be first a data set with a complete
> national network today and then delta data sets of
> individual planned projects. The challenge with the first is that the
> distinctions of the
> changes disappear together with their the time dimension.
> The project distinctions could somewhat be kept with the use
> of <border> with @name and @description for the project name
> and validity date. The challenge with the latter is to merge the
> projects into
> the existing network. The tracks cannot be connected as the
> id's could be inconsistent in the different data sets. But
> implicit information is available to place the project
> infrastructure on the existing infrastructure through the
> Norwegian consistent use of the values <line>@name,<absPos>
> and <track>@name

I prefer alternative 1A: complete snap-shots of the infrastructure for
different points in time. Since then all the related "scenario
information" need to be put into the metadata, I suggest to add the
already existing <state> sub element also to <infrastructure>. This
would help to have the basic information about the infrastructure
directly connected to the element.

> Alternative 2 - with time dimension
> Use time dimension as suggested with use @nor:state and

> @operatingPeriodRef. There exist different sub alternatives
> for placement.
>
> Alternative 2A - Integrated
> As suggested use @nor:state and @operatingPeriodRef and
> place them on individual elements in the structure for
> individual element (planned) states. Place them on a
> separate track with (planned) state if the @state spans an
> entire track. All elements on the track inherit the change.
> For states that span more than one ocp the state info is
> placed on the <infrastructure> element.
> There is a challenge using more than one <infrastructure>.
> The wiki says "An <infrastructure> defines the header
> container for various infrastructure versions." So multiple
> <infrastructures> should be allowed. But the XSD defines
> maximum of one occurrence of <infrastructure>. Can we ignore
> this?

The wiki comment is misleading (I corrected it now). It is not planned to allow for having more than one <infrastructure> element. However, of course you are free to have infrastructure elements of different states included in this container, but the <states> element is currently limited in usage.

> The reason we would like to use <infrastrucure> for
> different states is that having multiple time dimensions in
> one <infrastructure> under either separate <track>s with
> @nor:state and @operatingPeriodRef or even as loose elements
> in the structure itself can possible create inconsistencies
> if not kept completely cleanly structured.
> Alternative 2A1 multiple <infrastructure> Alternative 2A2
> with everything in one <infrastructure>
>
> Alternative 2B - Semi integrated time dimension time
> dimension in the metadata
> A work around would be to create one data set from one
> source application with multiple railML files zipped
> together. The individual railML files would contain
> <infrastrucutres> for each time dimension. The id's and the
> track connections between the infrastuctures would have to
> be created consistent. Loading the files together as one zip
> creating one model with time dimension.
> Based on what Christian writes and previous feedback I
> suggest the following:
>
> Alternatives 2A1 is not valid.
> We use alternative 1A for simple transfer between the
> systems. More complex systems with DB handling of scenarios

> can export either 1A, 1B or 2B depending on user choice.
> Alternative 2A2 seems quite complex... And should be left
> for railML3 I have made a simple illustration:
>
> What are your opinions?

Alternatives 1B and 2B are currently not possible, because of the very static structure of the railML schema. This means that in order to stay valid, a lot of information have to be considered in an export, even when they did not change. railML3 may solve this problem by reducing element hierarchy and limiting mandatory elements.

The complete "time dimension problem" is best to be solved with alternative 2A, but you are right: it is also the most complex one and it can only be solved with railML v3.

Other opinions and preferences?

Best regards
Christian

--
Christian Rahmig - Infrastructure scheme coordinator
railML.org (Registry of Associations: VR 5750)
Phone Coordinator: +49 173 2714509; railML.org: +49 351 47582911
Altplauen 19h; 01187 Dresden; Germany   www.railml.org

Subject: <state> for complete <infrastructure>
Posted by christian.rahmig on Thu, 16 Aug 2018 12:12:32 GMT
View Forum Message <> Reply to Message

Dear all,

Am 01.09.2017 um 14:32 schrieb Christian Rahmig:
> [...]
> Thinking about this top level state information, I assume that it may be
> useful to have the <state> element also available directly under
> <infrastructure>. This would us allow to recognize the difference
> between the different railML files and their included "infrastructure
> scenarios" without having to look in detail into them. So, probably a
> ticket for railML 2.4?

I created the Trac ticket #337 [1] that addresses the issue of having a <states> element for the root <infrastructure> element. Any comments and remarks on the proposal are very welcome...

[1] https://trac.railml.org/ticket/337

Best regards
Christian

--
Christian Rahmig - Infrastructure scheme coordinator
railML.org (Registry of Associations: VR 5750)
Phone Coordinator: +49 173 2714509; railML.org: +49 351 47582911
Altplauen 19h; 01187 Dresden; Germany   www.railml.org