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