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?