

Dear Christian,

thank you for your answer.

Your answers to my understanding statements tell me that I have misunderstood the principle of <ocpGroups>. The differences are:

- Trains can refer either to <ocps> or <ocpGroups>.
- Properties of <ocps> override the same properties of an associated <ocpGroup> but not the other way around.

> Well, for my understanding it should be possible to refer to an
> <ocpGroup> as well, because it has the same attributes like an <ocp>
> element.

I have assumed a strongly hierarchical structure, that's why I would not have dared to expect that trains being allowed to refer to <ocps> or <ocpGroups>. As far as I can see, it would be the first time that the target of a 'ref' attribute in RailML cannot be read from the attribute's name. (For example, a 'trainRef' always refers to a 'train', an 'operatingPeriodRef' always refers to an 'operatingPeriod'. But now, an 'ocpRef' could refer to an 'ocp' or an 'ocpGroup'? It then should have the name 'ocpOrOcpGroupRef'? ;-)

Anyway, it is all the same to me if only it works. But I would prefer a simple, clear and straight-forward solution.

>> So, my suggestion would be:
>> - to define an optional attribute 'parentOcpRef' (=tGenericRef) of an
>> <ocp>.

> However, your question brings it to the point: Do we need an explicit
> <ocpGroup> or may we reference from one <ocp> to the next/parent <ocp>?
> What do others think about this question regarding their usage of <ocp>?

It seems to me that an optional attribute 'parentOcpRef' (=tGenericRef) of an <ocp> is the simplest way to allow the desired function: No additional structures, no 'ref's to more than one target list, direct (cascaded) cross-references from a train to it's ocp's, more flexibility.

So, if there comes no objection during the next days or weeks, I would ask you to provide a ticket therefore for 2.2 and add it into the schemes. I would like to try it before final release of RailML 2.2 (as agreed), which we assume to be at InnoTrans, so I do need the XSDs latest in August.

Thank you!

With best regards,
Dirk.
