Hello,

christoph.jobmann@deutschebahn.com (Christoph Jobmann) writes:
> from my point of view trainPart elements can no longer be used more than
> once within the same trainPartSequence if rostering is included. In this
> case you will need several almost identitical trainParts - the only
> difference will be the id-attribute.

I go with your statement. You can't deploy reservation info when using
the same train part several times in a commercial train. All the other
'formationTT' elements could also differ for train parts running in the
same train through the same 'ocpTT's.

> Furthermore I don't like the thought of multiply referring to the same
> trainPart when working with commercial and operational trains since you
> will not be able to distinguish which references to trainParts in the
> operational train are included in which commercial train.

That's a really good point. Thanks for mentioning.

> I do realize that this leads to many identical trainPart-Elements; yet I
> find it neccessary to transmit the needed information without extending
> the standard.

railML should stay as redundancy-free as possible in the context of the
current model.

If it would be appreciated, we could separate the "ocpsTT" element for
referencing it from each train part. But I think that there are some
cases where some deep attributes of ocpTT differ between the coupled
train parts. Yes, it's a very seldom use case that should be covered,
too.

railML 2.1 is even just released, so I'm not very happy in changing
some core syntax. On the other hand, if that change would bring even
more users to railML, it would be worth.

Let's go on with this topic in this thread. Maybe it's good for the
next major release.

Any ideas, comments, questions appreciated...
Susanne

--

Susanne Wunsch
Schema Coordinator: railML.common