Subject: Rollingstock - modelling patterns
Posted by Joerg von Lingen on Tue, 20 Nov 2018 03:54:05 GMT
View Forum Message <> Reply to Message

Dear all,

in the misc-forum we have discussed about modelling patterns which shall apply for railML3 development. In the current
RS schema there is one main conflict with the set rules - model hierarchy.

1) What do you think about hierarchy issue concerning a perspective RS in railML3?
2) Shall the modelling split into vehicle components (independent of vehicle) in order to achieve a rather flat hierarchy?
3) What would you see in RS as "view", "container", "object" and "part"?

Refer also https://www.railml.org/forum/index.php?t=msg&th=573& goto=2014&#msg_2014

Regards,
Jörg v.Lingen - Rollingstock coordinator

Subject: Re: Rollingstock - modelling patterns
Posted by   on Thu, 22 Nov 2018 09:23:12 GMT
View Forum Message <> Reply to Message

Dear Jörg,

> 1) What do you think about hierarchy issue concerning a perspective RS in railML3?

I think a flat hierarchy is not more practical especially in a very 'technical' context. I already have often the problem of needing to 'jump' very often in the railML files (when reading manually) to resolve references. Sometimes I loose overview...

So, I would prefer the rather deep but contextual hierarchy we already have.

Additionally, when I made the suggestion of a possible generic model for future <TT> (with a very flat hierarchy), it was widely refused because of too less structure. So, I am probably (obviously) not the only one with this opinion.

Best regards,
Dirk.

Subject: Re: Rollingstock - modelling patterns
Posted by Joerg von Lingen on Thu, 22 Nov 2018 12:30:34 GMT
View Forum Message <> Reply to Message

Thanks Dirk for your input. I fully agree.

Regards,
Jörg v.Lingen - Rollingstock coordinator

Dirk Bräuer wrote on 22.11.2018 10:23:
> Dear Jörg,
>
>> 1) What do you think about hierarchy issue concerning a perspective RS in railML3?
>
> I think a flat hierarchy is not more practical especially in a very 'technical' context. I already
have often the problem of needing to 'jump' very often in the railML files (when reading manually)
to resolve references. Sometimes I loose overview...
>
> So, I would prefer the rather deep but contextual hierarchy we already have.
>
> Additionally, when I made the suggestion of a possible generic model for future <TT> (with a
very flat hierarchy), it was widely refused because of too less structure. So, I am probably
(obviously) not the only one with this opinion.
>
> Best regards,
> Dirk.
>