Subject: Re: [railML3] Extension methods
Posted by Milan Wölke on Thu, 09 Dec 2021 15:58:08 GMT
View Forum Message <> Reply to Message

Hi Thomas,

I meant to reply to this for a long time already. And after you brought it up again during the conference in Sweden, I feel I should add my position on this.
From my point of view we should switch over to the xsi:type based approach of extending existing railML modelling in railML 3. My reasoning would be that it is otherwise impossible to generate code for an extension. This may actually be fine for minimalistic extensions, but for anything that is a bit more complex, I think that it's a necessity. Even when not using code generation it has advantages such as the fact that like this documents can be validated.
I do not agree with the weakness you see with this approach regarding the fact that you need to specify it more than once like this. It is true, you need to specify it for all the places you need an extension. But you also need to implement it as an exporter and an importer at the same places anyway. Like this at least you can be sure that it either is specified or not.

JM2C

Best regards, Milan