

Hi,  
the issue was implemented with ticket:  
<https://trac.assembla.com/railML/ticket/197>

Kind regards,  
Joachim

-----  
Joachim Rubröder  
Schema Coordinator: railML.timetable

Susanne Wunsch wrote:

>  
> coord@timetable.railml.org (Joachim Rubröder) writes:  
>> Dirk Bräuer wrote:  
>>>> > Can we then clarify that for RailML, there is the following rule:  
>>>> > --> No ocpRef is allowed to occur more than one time in the same  
>>>> > <trainPart>.  
>>  
>> A forced splitting of trainParts whenever an ocpTT would occur several  
>> times would be consequent. This would also solve the problem of  
>> referencing the correct ocpTT within a trainPart  
>> ( <http://www.railml.org/forum/ro/?group=2&offset=0&thead=72&id=247>).  
>  
> I would also prefer this way of modeling instead of references to single  
> 'ocpTT's from within a 'trainPart'.  
>  
>>>> > We could now declare "trainReverse" being obsolete since we could  
>>>> > always use "orientationReversed" (also for single MUs by definition)  
>>>> > because we always will have to have a new <trainPart>.  
>>> I'm afraid I have to add one CON: The current 'trainReverse' attribute  
>>> fits to the very common symbol <-> for reversing direction in timetables.  
  
>>> I guess many public information systems have to handle this information.  
>>  
>> I would like to keep the 'trainReverse' attribute, for this purpose which  
>> was also mentioned by T. Kauer (SBB) at the railML meeting. With the  
>> forced splitting of trainParts, the 'trainReverse' would mainly occur at  
>> the first ocpTT of a trainPart if you have any formations referenced.  
>  
> It is some kind of redundancy, but it's a bit tricky to deduce it:  
>  
> 1. Find the commercial train, where this train part is used.

> 2. Look at the train parts at the previous trainPartSequence.  
 > 3. Look if the same formationTT is referred.  
 > -> 'trainReverse' is true.  
 >  
 > But if the formationTT refers some kind of general formation this  
 > deduction may be false.  
 >  
 > +1 for keeping "trainReverse"  
 >  
 > Instead of allowing the 'trainReverse' attribute only in the first  
 > 'ocpTT' we may include it in the 'formationTT' element as this may only  
 > occur once per 'trainPart'. This may be ensured by the XSD, but the  
 > occurrence of the attribute in the first 'ocpTT' element may only be  
 > ensured by Schematron, not XSD.  
 >  
 > That would mean, that both attribute 'trainReverse' and  
 > 'orientationReversed' will be in the same element, but with some kind of  
 > different meaning. I like to explicitly point to it, instead of "hiding"  
 > it in different elements:  
 >  
 > \* 'trainReverse' important for passenger information systems "<->"  
 >  
 > \* 'orientationReversed' referring to the definition of the formation in  
 > the rollingstock subschema  
 >  
 > There may be a trainPart with 'trainReverse=true' but with  
 > 'orientationReversed=false' because of an already reversed formation in  
 > the previous "train part sequence".  
 >  
 >> It should therefore no longer be seen as automatically reversing the  
 >> formation. For a simple timetable information system (without dealing  
 >> with formations) it could still be used within a long trainPart to  
 >> indicate the symbol <->.  
 >  
 > I would be happy if the railML semantics would be covered by all  
 > systems. That would mean, that already today a timetabling information  
 > system has to split train parts if the formation changes, nevertheless  
 > it does not know the formation type at all.  
 >  
 > Kind regards...  
 > Susanne  
 >

--

----- posted via PHP Headliner -----