Subject: Re: problems with <train>s: uniqueness constraints, scope
Posted by on Tue, 12 Mar 2013 17:12:03 GMT
View Forum Message <> Reply to Message

Dear Andreas,

Am 02.01.2013, 11:27 Uhr, schrieb Andreas Tanner <ata@ivu.de>:

> 1. If I read the standard correctly, trainPartRefs within a trainPart
> sequence model coupled trains. The wiki formulation
>
> " Therefore all referenced elements trainPart of a trainPartSequence
> should have the same starting point and end point"
>
> hints to that but is not really strict enough. The formulation should be
> changed to
> "The <ocpsTT> of all <trainPart>s within a <trainPartSequence> should
> contain the same sequence of <ocp>s with the same arrival and departure
> times."

It would be more restricting than today but it would also be easier for
parsing the files. So I would agree if the others do.

> 2. What variation of the trainPartSequences is allowed within one train?
> The case
>
> train x runs daily from A to B, and mon-fr a trainPart is added with
> position 2, and a trainPartSequence from B to C
>
> is apparently intended to be legal, while
>
> train y runs daily from A to B, and mo, tue, wed it continues to C but
> thu, fr, sat to D
>
> is not.

Your example is not clear enough to answer this. If C and D are at the
same route, it may be allowed. If it is a Y-like arrangement it is not.

> If this is so, I suggest adding the following -hopefully clarifying-
> text to the documentation:
>
> "For any <train>, there is a sequence of ocpTT without locational or
> temporal breaks, such that
> - for any <trainPartSequence>, there is a section of that sequence such
> that the ocpTTs of all referred trainParts of that trainPartSequence
> correspond with that section
> - the sections of subsequent trainPartSequences are subsequent to each

> other
> - for any operatingPeriod, the trainPartSequences spanned by the
> trainParts effective on that operatingPerid has no gaps."

I would not agree with the last item. I think I can imagine what you mean but also I think that writing of “gaps” in conjunction with operatingPeriods is not clarifying.

In general, it was not common in RailML the past to make such far-going restrictions. Rather, the philosophy of RailML was to more allow than restrict. I understand the practical advantage of such clarifications but since we do not have them in RailML at other themes, I think it is better to stay consequently.

Please consider that RailML should not be bound to the German philosophy of trains and train number usage. So I am afraid this would be left to bilateral agreements superset on RailML.

Andreas, please remember your own suggestions concerning a more wider definition of timetable periods in another discussion “thread”. There you see your own interest in not making things more restrictive.

But if there will be an agreement now or in future to change the philosophy to more restring usage of attributes and elements, I of course would agree and I would have some suggestions…

> 3. The scope construct is intended to model variations in the path of a
> train on different days. The wiki states the constraint
>
> "The compound of the attributes trainNumber, additionalTrainNumber and
> scope has to be unique for all <train> elements. If some of these
> attributes is absent the others have to be unique. The code attribute is
> used for some unique string identifying the train regardless of the
> unique attribute triple.".
>
> So a variation of a train path at an intermediate section should be
> modelled with scope "secondaryInner" - but what if you have _two_
> variations on different days?

For instance: Make two trains with scope = "secondaryInner", same trainNumber, different additionalTrainNumber. The additionalTrainNumber is used to distinguish between trains with the same trainNumber and the same scope.

> Also, the differentiation between "secondaryStart / secondaryEnd /
> secondaryInner" is redundant with the train path.

Yes and no.

Yes: All primary keys are a kind of redundant: They do only describe that there is a difference between two elements (of which they are primary keys) and you could always find it out yourself by comparing the contents of the elements…

No: Think on importing data from RailML a second time: You already have a train with trainNumber=xyz in your data and there is another train with trainNumber=xyz in the new RailML file. The scope+additionalTrainNumber tell you whether it is a new version of the same train or an additional train. Of course you could compare the routes but from that you would not now whether the route of the one and only train has changed or whether there is an additional (secondary) train with the same number.

> The constraint should be relaxed to allow variation of the train path on
> disjoint operatingPeriods.

Ah, we are now back at "relaxing" constraints. So quickly things can change…

> If a designated "primary" path is needed, the constraint should at least
> be relaxed to allow multiple trains with scope secondaryXXX.

This is already the case with additionalTrainNumber.

> In that case, I also propose to introduce a new scope value "secondary"
> and to mark secondaryStart / secondaryEnd / secondaryInner as deprecated.

We could do so, but why changing things already working? Of course the reading of the "secondaries" come from the terminology of Deutsche Bahn ("primary" = "Stammfahrplan", "secondaryInner" = "Doppelfahrplan" a. s.

o.), and of course this is arbitrary and not very "international". But at the time it was invented in RailML, nobody has had objections.
The solution to use the "additionalTrainNumber" for the German "Nummer des Ergänzungsfahrplans" came from Joachim and also was agreed then.

Now that we already have it I hope that there is a little bit "investment protection".

See it from the positive side: The secondary scopes are an additional (partly redundant) information on the intention of the originator of the trains and on how to "link" primary and secondary route sections. You do not need to scan and compare the routes/OCPs; you can expect that they "fit" at the place described by the type of "secondary".

Best regards,
Dirk