Subject: dayOffset vs. arrival/departureDay
Posted by on Mon, 12 Nov 2012 12:04:09 GMT
View Forum Message <> Reply to Message

Dear all,

in March 2012 we have created the <operatingPeriod>.dayOffset attribute. The original thought was to allow bitmasks with one or more digits than the there are days in the period. This was to describe a midnight-overrun before the station where the bitmask relates to.

Anyway, the longer bitmasks were not agreed. Instead, the new <operatingPeriod>.dayOffset attribute was created.

Since then, I have written some strange explanations at [1] and elsewhere but I am not satisfied with the redundancy which comes with <operatingPeriod>.dayOffset. With implementation, it becomes once more clear that it is always possible to avoid <operatingPeriod>.dayOffset>0 by using the already existing arrival/departureDay even at the first <ocpTT> of a <trainPart>. Even more worst, dayOffset leads by trend to define every <operatingPeriod> several times, one with dayOffset=0 and one with dayOffset=1 a.s.o.

See last sentence of my writings:

“It seams as if it is redundant whether a <trainPart> starts with departureDay=1 or refers to an <operatingPeriod> with dayOffset=1. It is not, since a train shall always start with departureDay=0 at its fist <ocpTT> in its first section; departureDay>0 is intended to happen only in first <ocpTT>s in further sections.”

I think we should throw it away before it becomes valid for the sake of less redundancy. Instead, we should turn that sentence around and say:

“Always when we thought we have to use dayOffset=1 we should use departureDay=1 instead.”

Therefore, I plead for deleting <operatingPeriod>.dayOffset before it ever became valid with RailML 2.2.

If the others agree, I would simplify the Wiki in that way.

Dirk.

[1] http://wiki.railml.org/index.php?title=TT:times#notes

Subject: Re: dayOffset vs. arrival/departureDay
Posted by Susanne Wunsch railML on Mon, 12 Nov 2012 15:50:29 GMT
View Forum Message <> Reply to Message

Dear Dirk and all,

Dirk Bräuer <dirk.braeuer@irfp.de> writes:

> in March 2012 we have created the <operatingPeriod>.dayOffset
> attribute. The original thought was to allow bitmasks with one or more
> digits than the there are days in the period. This was to describe a
> midnight-overrun before the station where the bitmask relates to.
>
> Since then, I have written some strange explanations at [1] and
> elsewhere but I am not satisfied with the redundancy which comes with
> <operatingPeriod>.dayOffset. With implementation, it becomes once more
> clear that it is always possible to avoid
> <operatingPeriod>.dayOffset>0 by using the already existing
> arrival/departureDay even at the first <ocpTT> of a <trainPart>. Even
> more worst, dayOffset leads by trend to define every
> <operatingPeriod> several times, one with dayOffset=0 and one with
> dayOffset=1 a.s.o.

I try to recover why we introduced that attribute although the above
mentioned "redundancy". Please correct me!

* If there are several train parts running long distances (over one or
  more midnights) and are coupled with other train parts for building
  some trains, how to define which 'departureDay' should one train part
  take? Do all train parts start with departureDay="0" never mind when
  another train parts started with whom it is coupled on the way?

  (If not, we would have huge problems, I mean.)

* If the train part goes over midnight at the last day of its operating
  period. It would operate on a day where it is not allowed by its
  operating period.

  The operating period gives the dates when the train part starts, the
  'dayOffset' gives the count of days which the train runs over
  midnight. It only shifts the planned operation days.

  Of course, the departureDay/arrivalDay attributes of the ocpTTs during
  the run should count from cero up to the 'dayOffset' value.

  Why to start a train part with departureDay="1"?

> [1] http://wiki.railml.org/index.php?title=TT:times#notes

Thanks for the wiki documentation. :-)

Kind regards...
Susanne

--
Susanne Wunsch
Schema Coordinator: railML.common

Subject: Re: dayOffset vs. arrival/departureDay
Posted by on Mon, 12 Nov 2012 19:10:49 GMT
View Forum Message <> Reply to Message

Dear Susanne,

> I try to recover why we introduced that attribute although the above
> mentioned "redundancy". Please correct me!

> * If the train part goes over midnight at the last day of its operating
> period. It would operate on a day where it is not allowed by its
> operating period.

That was the original thought behind it, as I explained in my sentence:

>> The original thought was to allow bitmasks with one or more
>> digits than the there are days in the period.

> The operating period gives the dates when the train part starts, the
> 'dayOffset' gives the count of days which the train runs over
> midnight. It only shifts the planned operation days.

Yes. departureDay>0 would do the same.

Well, I did not wanted to start a big discussion again. I only wanted to
tell that <operatingPeriod>.dayOffset _can_ be avoided by using
departureDay="1" at the first <ocpTT> - believe me.

If you want to keep both attributes anyway - no problem, just redundancy.

Best regards,
Dirk.