
Subject: Re: simple Infrastructure scheme usage as a base for Timetable reference
Posted by on Tue, 23 Sep 2008 20:37:16 GMT

[View Forum Message](#) <> [Reply to Message](#)

Am 22.09.2008, 18:36 Uhr, schrieb Dr. Volker Knollmann <volker.knollmann@siemens.com>:

- > Hello Dirk,
- >
- > thank you for your message and please excuse my late reply. Regarding
- > your first question about defining special positions as reference points
- > for time table purposes ("Fahrzeitmesspunkt") I think I've good news.
- > You requested a link between a track and an ocp and I can confirm that
- > exactly such a link already existis. The associated element is called
- > "crossSection" an dis located in the infrastructure part.

Thank you. I indeed did not recognise "crossSection" as a cross-link from tracks to OCPs. I thought (w/o looking in its attributes) that it is what its name says: Just the cross-section of the track, what in German would be described with the word "Lichtraumprofil". Now that I know that "crossSection" means cross-link (or reference) it is clear what my next question is: Where is the real cross-section, i. e. the Lichtraumprofil of the track?

Anyway, I now know the link from tracks to OCPs I was looking for. We do not have links from OCP to tracks so a parsing programme which needs this has to scan all tracks for links to a certain OCP. It's ok for me; I am also not a big friend of redundancies.

- > Your approach to use "empty <track>-elements" in order to indicate the
- > number of operational tracks does definitely not conform with the
- > RailML-architecture.

Here you misunderstood me. I never intended to create empty track elements and I am certainly aware that this would not be a proper use. What I asked for was to get tracks w/o "trackBegin" / "trackEnd" elements. These tracks should not be empty at all! They should contain the crossSections-OCP-Refs you've shown me above and also nearly all trackElements (speeds, gradients, curves, gauge a.s.o.). Of course these tracks have a virtual start (the mileage of the first crossSection-OCP-Ref) and a virtual end (the mileage of the last crossSection-OCP-Ref). But I cannot declare these places as the trackStart and trackElements in the meaning of the detailed infrastructure architecture where this means the real start/end (buffer stop or point) of the track. I do not know whether the real track starts on my first OCP-Ref or whether it "comes from somewhere". In fact, I do know that the track does not start exactly at the place of the first OCP-Ref because an OCP-Ref is always (defined as) at the middle of the station/platform. The distance between first OCP-Ref and start of the track is at least half of a train/platform length.

- > If there really is a common need to add an appropriate attribute for the number of tracks...

There is. What I try to describe in a RailML file is what you can see in the picture http://www.irfp.de/download/example_graphic_timetable_header.jpg. Everything in it is infrastructure, and it is a common graphic timetable header. You can see the no. of line tracks and

the "crossSection"-OCP-Refs as the intersection of horizontal station lines and vertical track lines.

> ... we can insert it in a future RailML release.

I want to ask you to do that if there is no existing possibility. What I try to develop is a new generation of RailML timetable scheme. Parallel to this it would be good to have the extended infrastructure scheme where necessary. My expectation is to have them ready at the end of the year. May be you can talk and decide about it at Innotrans meeting on Friday.

> For this reason I doubt that making <trackBegin> and
> <trackEnd> optional elements - as you suggested in your posting - is a
> good move.

I think that it should be possible to describe a common graphic timetable header with its typical information (and not more!) in RailML. This means, having no detailed information about trackside elements but about the stations (OCPs) only. Therefore, I again ask to make trackBegin / trackEnd optional to create a track whose topology consists of "crossSection"-OCP-Refs only. I do not and never did opt for an "empty" track.

If not, there is a possibility to make a trackBegin directly at the same mileage as the first OCP-Ref and a trackEnd directly at the same mileage as the last OCP-Ref. But then, it would be necessary to have a new type of trackBegin/trackEnd element because it would not be a simpleConnection (point or head-on junction) and not a buffer stop. We would need something which makes the trackBegin/trackEnd virtual (saying that there is no real trackBegin/trackEnd), without any reference. But I would prefer a solution without trackBegin/trackEnd because I think a trackBegin/trackEnd at the same mileage as the first/last entry has not new information and no use and is a little like bureaucracy.

With best regards,
Dirk.
