
Subject: simple Infrastructure scheme usage as a base for Timetable reference

Posted by on Tue, 05 Aug 2008 14:26:47 GMT

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

Hello,

I try to visualise a simple line in RailML.Infrastructure scheme. The line should consist of stations only, no elements so far. So it should be only a "rough" usage to create a most rudimentary database for a timetable without detailed information. The aim is to have an infrastructure which can be referred by Timetable scheme (Trains refer to OCPs of infrastructure).

Therefore, I filled the following elements:

- RailML.Infrastructure.Line.Id
- RailML.Infrastructure.OCP.Id,Name,prop... a.s.o.

The problem is, that it seems to me that there is no possibility to create a cross-reference from an OCP to a Line.Id. Also, I did not find any trackElements which can refer to an OCP. There should be a possibility to link OCPs and Lines/Tracks and to create TrackElements as "Fahrzeitmesspunkte" (representing the station lines in a graphic timetable!). From my opinion, such a TrackElement="Fahrzeitmesspunkt" is simply a cross-reference to an OCP.

I would like to fill also the RailML.Infrastructure.Line.Track structure to say whether the line is single or double track (one or two track substructures) and also to fill some TrackElements but this doesn't work because I cannot give a proper "trackTopology.trackBegin" and "trackTopology.trackEnd" element. They should be optional for such a rough usage, or it should be possible to refer to an OCP there.

Can the responsible scheme coordinator please give some hints how to deal with that problem?

Thank you!

Dirk.

--

Erstellt mit Operas revolutionärem E-Mail-Modul: <http://www.opera.com/mail/>

Subject: Re: simple Infrastructure scheme usage as a base for Timetable reference

Posted by [Dr. Volker Knollmann](#) on Mon, 22 Sep 2008 16:36:19 GMT

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

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.

A crossSection is bound to a certain location on the track (identified by lineID, trackID and position on track) and references an ocp via its IP. As far as I know from RailML's history, this element has been introduced with exactly the same thought in mind that you described in your original posting.

Regarding your other two questions (attribute for the number of tracks / empty <track> elements), I'm afraid to have no such easy answers for you. An attribute for the number of regular tracks per line is so far not implemented. Unfortunately, you cannot derive it from the number of <track>-elements belonging to that line, because cross-overs, passing loops, station tracks etc. lead to numerous <track>-elements all inside the same <line>-section. If there really is a common need to add an appropriate attribute for the number of tracks, we can insert it in a future RailML release. For the time being I suggest to use the <anyAttribute>-extension which allows you to add arbitrary attributes to almost all RailML elements.

Your approach to use "empty <track>-elements" in order to indicate the number of operational tracks does definitely not conform with the RailML-architecture. The <track> is a container for the "physical" tracks and all other (physical) elements along that track. With its sub-element <trackTopology> (which owns <trackBegin>, <trackEnd> and <mileageChanges> among others), the coordinate system for subsequent position information is established. So these elements contain crucial data for the interpretation and evaluation of the infrastructure's topology. For this reason I doubt that making <trackBegin> and <trackEnd> optional elements - as you suggested in your posting - is a good move. This would affect the very heart of the infrastructure schema and would lead to much more efforts for parsers and import filters. We should have a certain set of core data which every RailML-capable application can rely on. In my opinion, a consistent track description including length etc. is part of that core data.

I hope that these explanations answer your questions. If not, please don't hesitate to get in touch with me!

Best regards,
Volker

--

Dr. Volker Knollmann
Siemens AG
Industry Sector
I MO RA SPP SM MT 1
Ackerstr. 22
38126 Braunschweig

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.

Subject: Re: simple Infrastructure scheme usage as a base for Timetable reference
Posted by [Dr. Volker Knollmann](#) on Wed, 24 Sep 2008 18:46:27 GMT
[View Forum Message](#) <> [Reply to Message](#)

On 23/09/08 22:37, Dirk Bräuer wrote:

> Where is the real cross-section, i. e. the Lichtraumprofil of the
> track?

We once had an intensive discussion about how to store information about a tracks clearance. This discussion happend during a RailML-meeting approx. 2 years ago. We had a participant with really deep knowledge about the different clearance variants for railway tracks and it was

obvious, that we cannot store all related data in two or three data fields.

I personally don't know enough about that topic to make a sound suggestion for appropriate data structures. I still hope for an expert to had in his / her ideas about clearance profiles.

Besides that, the clearance is nothing physical you can touch. And this was the original criteria for new elements to be added to infrastructure file. But nevertheless I admit that the clearance is something strongly related to the infrastructure and it definitely makes no sense to store it elsewhere than in the infrastructure.

Perhaps we can re-animate that discussion along with the storage of topographical information like clothoids etc. Susanne did some very good preparational work for that!

- > 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.).

But the information inside <trackBegin> and <trackEnd> is vital for all other elements you mentioned (speeds, gradients, curves, gauge a.s.o.). Without begin / end, you cannot give proper location information for all these elements.

- > 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.

Hmm... first of all, I don't like the idea of "virtual" starts and ends of tracks. This would introduce new semantics to core RailML-elements which would definitely confuse existing parsers.

But I think there is a solution to your problem, if I understood you correctly. Time for some ASCII-art:

```
]----T1-----|-----T2-----|---- -T3----[
           xxxxxxxx           xxxxxxxx
]----T4-----|-----T5-----|---- -T6----[
```

"]--" denotes a buffer stop, "|" the joint between two <track>s and "xxx" a platform.

If you compose your infrastructure like this, you gain tracks T2 / T5 with a relative coordinate of value "zero" in the middle of the platform. All elements placed along these tracks are located relative to this reference point (e. g. a signal at 1.234 km distance from the platform center).

Using RailML this way is not exactly how it is intended (T1, T2 and T3 as a single <track>), but it is often used this way and all parses should be able to handle it.

>> If there really is a common need to add an appropriate attribute for
>> the number of tracks we can insert it in a future RailML release.

> I want to ask you to do that if there is no existing possibility.

I'm afraid that - as usual - things aren't as easy as they seem to be. In my opinion, a simple "numberOfTracks"-attribute for a <line> is NOT the solution. Look at your own example in the JPG-file: you have single- and double-line sections ON THE SAME LINE. So what to store in the <line>-header? Double track? Single track? 1,5 tracks? "Mostly single track"?

A better solution could be to store references to parallel <tracks>. Imagine something like this:

```
<parallelTracks>
  <parallelTrack ref1="T1" ref2="T4"/>
  <parallelTrack ref1="T2" ref2="T5"/>
  <parallelTrack ref1="T3" ref2="T6"/>
</parallelTracks>
```

.... with up to ref3 or ref6 or ref42 or whatever attributes. But this brings in new semantic constraints which are difficult to handle: what if two tracks marked as parallel do not have the same length? Do two parallel tracks need to share the same start- / end-coordinates? What if the two tracks do not have the same alignment? In the latter case, the line may have double tracks but the routing (and therefore the infrastructure data) can be completely different. Do you have any ideas how to handle that?

> What I try to develop is a new generation of RailML timetable scheme.

Uuuuhh... especially the timetable file format is heavily used and in my opinion the most mature part of RailML. And if you talk about "a new generation" this sounds to me more like a "revolution" and not like "little enhancements". Without knowing what your ideas and changes are

about my first impression is that "a new generation" is no good signal to the users of our file format. Especially if we think of the time table part.

I'd like to strongly encourage you to get in touch with Joachim Rubröder. He is the coordinator for the time table schema. Additionally, please post details to your changes and ideas in the newsgroup.

- > 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.

Honestly, I really appreciate your efforts to improve RailML! And I'd be happy to discuss your ideas together with all the other participants on Friday. But please understand that we cannot decide about major changes on Friday without having had the chance to see them in advance.

If time isn't too short, please prepare a presentation for Friday in which first of all explain

- * WHAT you want to store / change (==> content)

and then, optionally,

- * HOW you want to store it (data types, structure).

Please inform Vasco if you want to give a presentation on Friday so that he can adapt the schedule.

I'm curious about your ideas and comments! See you on Friday!

Best regards,
Volker

--

Dr. Volker Knollmann
RailML Infrastructure Coordinator

Business contact:

Siemens AG
Industry Sector
I MO RA SPP SM MT 1
Ackerstr. 22
38126 Braunschweig

Subject: Re: simple Infrastructure scheme usage as a base for Timetable reference
Posted by on Wed, 24 Sep 2008 22:10:58 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hallo Volker,

Am 24.09.2008, 20:46 Uhr, schrieb Dr. Volker Knollmann <coord@infrastructure.railml.org>:

- > But the information inside <trackBegin> and <trackEnd> is vital for all
- > other elements you mentioned (speeds, gradients, curves, gauge a.s.o.)..
- > Without begin / end, you cannot give proper location information for all
- > these elements.

I disagree. I do not know what exactly you mean with "proper location information". All I have and all I need is mileage/kilometres of the trackside elements. It is possible to describe the kilometres of the line with only the kilometres of its elements and the "mileageChange" element.

- > But I think there is a solution to your problem, if I understood you
- > correctly. Time for some ASCII-art:

```
>
> ]----T1-----|-----T2-----|---- -T3----[
>           xxxxxxx          xxxxxxx
> ]----T4-----|-----T5-----|---- -T6----[
```

- >
- >
- > "]"--" denotes a buffer stop, "|" the joint between two <track>s and
- > "xxx" a platform.

Please look at the graphic timetable header I sent: As you can see, there is no trace of a buffer stop at the start nor the end of the line. The graphic timetable starts at the first | in your ASCII art, NOT at any buffer stop before. So again: My problem is the T1, T3 T4, T6 sections because a graphic timetable consists only of T2 and T5. I want to start my track at | and end at | and do not want to have any buffer stops!

- > If you compose your infrastructure like this, you gain tracks T2 / T5
- > with a relative coordinate of value "zero" in the middle of the
- > platform. All elements placed along these tracks are located relative to
- > this reference point (e. g. a signal at 1.234 km distance from the
- > platform center).

I am aware of the difference between relative and absolute kilometres. I do not have any problem with kilometres of the line, changes of kilometres and I am able to use the element type "mileageChange".

- > I'm afraid that - as usual - things aren't as easy as they seem to be.
- > In my opinion, a simple "numberOfTracks"-attribute for a <line> is NOT
- > the solution.

But I did never write that I want to have a "numberOfTracks"-attribute. Never. I do not want it!

What I did ask you is to make "trackBegin" and "trackEnd" optional. Please tell me "yes" or "no, because...".

- > Look at your own example in the JPG-file: you have single- and double-line sections ON THE SAME LINE. So what to store in
- > the <line>-header? Double track? Single track? 1,5 tracks? "Mostly single track"?

This is not objective. Who said anything about a RailML line header???

Concerning my graphic example, in RailML would look like:

```
infrastructure
  lines
    line id=5102
      tracks
        track id=5102.0
          trackTopology
            crossSections
              crossSection id=OR
              crossSection id=ONPNU
              crossSection id=OSTB
              crossSection id=OTHA
              crossSection id=OCB
              crossSection id=OC
            /crossSections
          /trackTopology
        /track
        track id=5102.1
          trackTopology
            crossSections
              crossSection id=OC
              crossSection id=OCS
              crossSection id=OEES
              crossSection id=OWSK
            /crossSections
          /trackTopology
        /track
        track id=5102.2
          trackTopology
            crossSections
              crossSection id=OC
              crossSection id=OCS
```

```

        crossSection id=OESS
        crossSection id=OWSK
    /crossSections
  /trackTopology
/track
track id=5102.3
  trackTopology
    crossSections
      crossSection id=OWSK
      crossSection id=OMOLL
    ... a. s. o.

```

As you can see: The line 5102 has two tracks (=is double-track) between OC and OWSK. There is only one track between OR and OC. There are no trackBegin and trackEnd elements. The kilometres will be defined by the crossSections and by mileageChange elements.

I AGAIN REPEAT MY WISH: Please tell me whether my example of usage of infrastructure scheme will be possible (by making trackBegin / trackEnd optional) OR please make trackBegin / trackEnd usable in such a kind that it needs not to be a buffer stop nor a point nor a head-on junction. THANK YOU.

> Uuuhh... especially the timetable file format is heavily used and in my
 > opinion the most mature part of RailML. And if you talk about "a new
 > generation" this sounds to me more like a "revolution" and not like
 > "little enhancements".

You can believe me that I am aware of the usage and limits of the current RailML timetable scheme. At some earlier meeting in Dresden I demonstrated some of the reasons why there are needs for a new timetable scheme. Joachim knows about this and is already working on this new scheme. My work should be only a support to him.

But here, we should discuss technical and technological problems.

> But please understand that we cannot decide about major changes on Friday without having had the chance to see them in advance.

??? I do not want to have major changes! What I want is simply to have trackBegin and trackEnd to be optional!

I was told that this forum is the right way to get such little improvements.

> If time isn't too short, please prepare a presentation for Friday in which first of all explain
 > * WHAT you want to store / change (==> content)

??? I did sent you a message describig exactly that and I did send you a graphic timetable header. This is what I want to store!

I think that the time at our fairground stand at InnoTrans needs my presence and therefore I will not be present at the RailML meeting on Friday.. Let's see. Since the infrastructure of a graphic timetable (what I want to store in RailML) should be a very common thing and I want only two minor changes (optional elements) I wonder why it is necessary therefor to prepare a speech.

It seems that there are major misunderstandings and I think that a personal talk would be good. If you can find the time to visit our stand at InnoTrans on Thursday or Friday (hall 2.1 stand 220) it would be fine. If not, I try to come to the meeting at least for a short talk.

Thank you,
with best regards,
Dirk.

Subject: Re: simple Infrastructure scheme usage as a base for Timetable reference
Posted by [Dr. Volker Knollmann](#) on Sun, 28 Sep 2008 08:48:25 GMT

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

Hi Dirk,

I think we have some severe misunderstandings in our discussion. I doubt that we can solve just by exchanging newsgroup posts.

Since you didn't attend the meeting on Friday I ask you to call me on the phone in order to discuss your questions. This will be the easiest solution.

Best regards,
Volker

--

Dr. Volker Knollmann
RailML Infrastructure Coordinator
EMail: coord@infrastructure.railml.org

Business contact:

Siemens AG
Industry Sector
I MO RA SPP SM MT 1
Ackerstr. 22
38126 Braunschweig
Tel.: +49 (531) 226-2592
<mailto:volker.knollmann@siemens.com>
