

---

Subject: Modified V94\_12

Posted by [volker.knollmann](#) on Tue, 18 Nov 2003 08:04:47 GMT

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

---

Hello,

based on a suggestion from Matthias Hengartner, I modified the latest version of the scheme (v94\_12) according to the ideas I discussed with Matthias via email. You can download my modified version here:

[http://creature.apm.etc.tu-bs.de/~volker/infra\\_V094\\_12\\_knollmann\\_1.xsd](http://creature.apm.etc.tu-bs.de/~volker/infra_V094_12_knollmann_1.xsd)

The most significant change is the introduction of a element called `<ocsElements>`, which is a child of `<track>`. "ocs" means "operation- and controlsystems" and is meant to be a translation of the German "Leit- und Sicherungstechnik".... err.... if you know a better translation, feel free to tell me :-)

`<ocsElements>` should contain all objects related to operation, control, train protection and similar things. Just to give you an idea, I created sub-elements for Balises, track coupling coils, track circuits, LEUs and EuroLoops. And I moved the `<signals>`-Element into this `<ocsElements>`, of course.

The detailed definition, especially the attributes, of the new elements is not yet complete. I just want you to get an idea of my changes and if you think that I chose a good way, we / I can spend more time on this.

Another change is the introduction of the attributes `<prevTrackID>` and `<nextTrackID>` for the element `<track>`. With this means, you are able to compose a line of multiple tracks similar to a double linked list. Of course, `<nextTrackID>` of one track and `<prevTrackID>` of the next track need to be consistent, otherwise the xml-file would be corrupt.

These two attributes are just a quick & dirty suggestion. Up to know, I'm not really sure whether I really like them myself. To be honest, I'm not even sure whether you really need to compose a line of several tracks. Background: due to your microscopical model, you only need to define special points (swiches, signals, xxxChange, etc) along the track. This can all be done within one track element. This is in contrast to a macroscopical model, where to need to compose a "long" track (line) of several edges. Therefore, I see the need to compose one line of several tracks only for the use of a macroscopical model, not for the use of a microscopical model.

I'm afraid that my description was quite confusing... if you have question or if I wrote complete bullsh\*\*, please tell me. I'd really appreciate a discussion about my ideas here in this newsgroup.

Thanks to everyone who read this long posting! I'm looking forward to your comments!

Best regards from Braunschweig,  
Volker Knollmann

---

---

Subject: Re: Modified V94\_12  
Posted by [nfries](#) on Tue, 18 Nov 2003 19:58:22 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hello,

in my thesis handed in at the TU Dresden in August presenting a new version of the infrastructure scheme some of your points have been discussed. I might try to give some propositions from my point of view:

- > The most significant change is the introduction of a element called
- > <ocsElements>, which is a child of <track>. "ocs" means "operation- and
- > controlsystems" and is meant to be a translation of the German "Leit- und
- > Sicherungstechnik".... err.... if you know a better translation, feel free
- > to tell me :-)

How about "command and control systems"?

- > <ocsElements> should contain all objects related to operation, control, train
- > protection and similar things. Just to give you an idea, I created
- > sub-elements for Balises, track coupling coils, track circuits, LEUs and
- > EuroLoops. And I moved the <signals>-Element into this <ocsElements>, of
- > course.

What does LEU mean?

Why did you leave <trainProtectionChanges> as an own element?

- > The detailed definition, especially the attributes, of the new elements is
- > not yet complete. I just want you to get an idea of my changes and if you
- > think that I chose a good way, we / I can spend more time on this.

It is for sure a possible solution, although the whole element might get rather clumsy if you try to cover all technical solutions of communication between engine and infrastructure. The <signal> element will need further development in order to be practically applicable.

- > Another change is the introduction of the attributes <prevTrackID> and
- > <nextTrackID> for the element <track>. With this means, you are able to
- > compose a line of multiple tracks similar to a double linked list. Of

- > course, <nextTrackID> of one track and <prevTrackID> of the next track
- > need to be consistent, otherwise the xml-file would be corrupt.
  
- > These two attributes are just a quick & dirty suggestion. Up to know, I'm
- > not really sure whether I really like them myself. To be honest, I'm not
- > even sure whether you really need to compose a line of several tracks.
- > Background: due to your microscopical model, you only need to define
- > special points (swiches, signals, xxxChange, etc) along the track. This
- > can all be done within one track element. This is in contrast to a
- > macroscopical model, where to need to compose a "long" track (line) of
- > several edges. Therefore, I see the need to compose one line of several
- > tracks only for the use of a macroscopical model, not for the use of a
- > microscopical model.

Still there might be some occasions where the possibility to connect two <track> elements will be useful, e.g. if you want to combine to sections of infrastructure already existing in RailML (analogy to OpenTrack), there will be the need to refer to the appropriate point in the neighbour model. In that concern one single attribute "nextTrackId" will not be sufficient because the point of reference on the given track is not specified. My version provides a <connection> element containing the attributes "trackID", "pos" and "dir" (sense of mileage).

Greetings from Paris,

Nikolaus Fries

---

---

Subject: Re: Modified V94\_12  
Posted by [volker.knollmann](#) on Wed, 19 Nov 2003 08:41:19 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Nikolaus Fries wrote:

- > What does LEU mean?

LEU is a "Lineside Electronic Unit" and a part of the ETCS System. On one side, a LEU is connected to the lamp circuits of a conventional signal to determine the signal aspect. On the other hand, it is connected to variable balises to transmit arbitrary ETCS-telegrams depending on the signal aspect. With a LEU, existing tracks can be upgraded to ETCS Level 1 without too expensive changes to the infrastructure (signals, cables, interlocking etc. stay the same).

- > Why did you leave <trainProtectionChanges> as an own element?

I interpreted that element as a kind of "instruction", which kind of ATP-system is to be used on lines with multiple systems. So from my point

of view, <trainProtectionChanges> does not describe / define / enumerate elements of "command- and control-systems" (thanks for your suggestion!) and is therefore not a part of <ocsElements>.

>> [... introduction of <ocsElements> ...]

- > It is for sure a possible solution, although the whole element might get
- > rather clumsy if you try to cover all technical solutions of communication
- > between engine and infrastructure.

Yes, I definitely agree to your point. But in a first step, we don't need to include all possible means of communication in this group. I just demonstrated with some examples, how <ocsElements> should be used. Perhaps "LEU" was way too exotic... I should have chosen "<axleCounters>" for example...

The number and kind of "sub-elements" for the container <ocsElements> can be defined depending on the needs of the first applications. But I think it's a good idea to have at least the most important systems (track circuits, axle counters, LZB, ETCS) at hand.

- > The <signal> element will need further
- > development in order to be practically applicable.

YES. The same applies to various other elements (e.g. switches). I'm having a list with notes and comments regarding railML on my desk and this list is growing continuously. So you will have to suffer from some more of my postings in this newsgroup in the future...

>> Therefore, I see the need to compose one line of several  
>> tracks only for the use of a macroscopical model, not for the use of a  
>> microscopical model.

- > Still there might be some occasions where the possibility to connect two
- > <track> elements will be useful, e.g. if you want to combine to sections
- > of infrastructure already existing in RailML (analogy to OpenTrack), there
- > will be the need to refer to the appropriate point in the neighbour model.

You think of distributing one line over several documents (like OpenTrack)? Okay, in this case we need a function to combine two <track>s, that's right.

- > In that concern one single attribute "nextTrackId" will not be sufficient
- > because the point of reference on the given track is not specified.

Yes, I've seen that problem, too. You really need a double-linked list (with all its drawbacks).

- > My version provides a <connection> element containing the attributes

> "trackID", "pos" and "dir" (sense of mileage).

According to Annex B of your thesis, you completely skipped <tracks> and <track>. And you renamed <trackData> to <linaData> and assigned all information directly to the <line>. Apart from the fact that was a useful simplification (in my opinion), I found no <connection>-Element in your infrastructure-scheme :-(

Best regards from Braunschweig,  
Volker Knollmann

---

Subject: Re: Modified V94\_12  
Posted by [nries](#) on Wed, 19 Nov 2003 22:16:47 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hello,

>> It is for sure a possible solution, although the whole element might get  
>> rather clumsy if you try to cover all technical solutions of communication  
>> between engine and infrastructure.

> Yes, I definitely agree to your point. But in a first step, we don't need  
> to include all possible means of communication in this group. I just  
> demonstrated with some examples, how <ocsElements> should be used. Perhaps  
> "LEU" was way too exotic... I should have chosen "<axleCounters>" for  
> example...  
> The number and kind of "sub-elements" for the container <ocsElements> can  
> be defined depending on the needs of the first applications. But I think  
> it's a good idea to have at least the most important systems (track  
> circuits, axle counters, LZB, ETCS) at hand.

Here you are risking an endless discussion about the "most important" systems, because this is the german point of view...

Still you are right that for a version 1.0 we will have to have a limited choice of subelements in the <ocsElements>.

Altogether I agree to integrate all these kinds of elements into the <ocsElement>.

>> The <signal> element will need further  
>> development in order to be practically applicable.

> YES. The same applies to various other elements (e.g. switches). I'm  
> having a list with notes and comments regarding railML on my desk and this  
> list is growing continuously. So you will have to suffer from some more of  
> my postings in this newsgroup in the future...

You will probably have taken notice of my propositions concerning the

signal element. What do you think of them? Unfortunately Matthias has not integrated them into his scheme so that now we are still working with its first version.

>> Still there might be some occasions where the possibility to connect two  
>> <track> elements will be useful, e.g. if you want to combine to sections  
>> of infrastructure already existing in RailML (analogy to OpenTrack), there  
>> will be the need to refer to the appropriate point in the neighbour model.

> You think of distributing one line over several documents (like  
> OpenTrack)? Okay, in this case we need a function to combine two <track>s,  
> that's right.

>> In that concern one single attribute "nextTrackId" will not be sufficient  
>> because the point of reference on the given track is not specified.

> Yes, I've seen that problem, too. You really need a double-linked list  
> (with all its drawbacks).

>> My version provides a <connection> element containing the attributes  
>> "trackID", "pos" and "dir" (sense of mileage).

> According to Annex B of your thesis, you completely skipped <tracks> and  
> <track>. And you renamed <trackData> to <linaData> and assigned all  
> information directly to the <line>. Apart from the fact that was a useful  
> simplification (in my opinion), I found no <connection>-Element in your  
> infrastructure-scheme :-(

In fact with my thesis I was still based on an older version. As far as I know the elements <tracks> and <track> have been introduced by Ulrich Linder whose scheme formed the base for Matthias (please correct me if I'm wrong).

My <connection> element ist derived from <kilometer> if that might help you to find it. To fit it into your version it might be best made a subelement of <track> as an alternative to your two new attributes.

Best regards from Paris,

Nikolaus Fries

---

---

Subject: Re: Modified V94\_12  
Posted by [volker.knollmann](#) on Thu, 20 Nov 2003 10:23:56 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Nikolaus Fries wrote:

- > But I think
- >> it's a good idea to have at least the most important systems (track circuits, axle counters, LZB, ETCS) at hand.
  
- > Here you are risking an endless discussion about the "most important" systems, because this is the german point of view...

Okay, so I officially declare open this discussion!! :)  
Anybody who wants to contribute?

IIRC, Switzerland for example uses track magnets (or track beacons) for their widely spread ATP-systems SIGNUM and ZUB 121. And ZSI 27 / ZSI 127 uses Balises.

Austria uses LZB and PZB (INDUSI) just like Germany.

All these systems could be defined im RailML using the elements <trackCouplingCoil>, <trackCircuit> and <Balise>, couldn't they?

Unfortunately, I know nothing about other European Countries. Who can help?

- > You will probably have taken notice of my propositions concerning the signal element. What do you think of them?

Sorry, but when you spoke of "your propositions", I thought you were refering to your thesis. Obviously, I was wrong. Which other proposals have you made? I found nothing here in this newsgroup, but perhaps I'm just blind... can you give me a hint?

- > My <connection> element ist derived from <kilometer> if that might help you to find it. To fit it into your version it might be best made a subelement of <track> as an alternative to your two new attributes.

I will have a look at your scheme as soon as I found it (see above). I promise!

- > Best regards from Paris

^v^v^v^v^v^v^v^v^v^v

<kidding>

So you are the ideal candidate to add French ATP-components to <ocsElements>, hmm?

</kidding>

As usual, best regards from Braunschweig!  
Volker Knollmann

---

---

Subject: Re: Modified V94\_12

Posted by [Matthias Hengartner](#) on Thu, 20 Nov 2003 11:05:03 GMT

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

---

Hello,

> You will probably have taken notice of my propositions concerning the  
> signal element. What do you think of them? Unfortunately Matthias has not  
> integrated them into his scheme so that now we are still working with its  
> first version.

and

> In fact with my thesis I was still based on an older version. As far as I  
> know the elements <tracks> and <track> have been introduced by Ulrich  
> Linder whose scheme formed the base for Matthias (please correct me if I'm  
> wrong).

Yes, my work based on the scheme 0.94 from Ulrich Linder and so I didn't integrate the propositions from Nikolaus (I asked some months ago, if those propositions will be integrated in the "official schema", but I didn't receive an answer. Furthermore, this was not the core topic of my thesis, and I'm not an traffic engineer but only a computer scientist ;-))  
But I'm glad that the discussion is resumed now and we can talk about the integrations Nikolaus' propositions, not only concerning signal elements but also the new line/track-elements, etc.

Best regards from foggy Zurich :-)  
Matthias Hengartner

---