Subject: infrastructure_V094_13
Posted by Matthias Hengartner on Thu, 27 Nov 2003 14:30:49 GMT
View Forum Message <> Reply to Message

Hello,

on http://www.theband.ch/matthias/railml/infrastructure_V094_13 .zip you can find a new suggestion for the infrastructure-schema.

After a short meeting with Nikolaus Fries last friday, we agreed that Nikolaus will make some considerations about signals and other "operation- and controlsystem elements" (ocs) resp. "command and control system elements", whereas I'll think about switches and connections.

In my scheme, there are some parts adopted from Nikolaus' thesis and from the suggestions of Volker Knollmann.
Some suggestions from Volker are not yet included, because they concern mostly the area of ocs. And there are more elements and attributes in Nikolaus' scheme which could be adopted in a next draft.

Let's begin:

--- begin switches ---
A <switch> can be either a <junction> (Abzweigung) or a <crossover> (direkter Gleiswechsel). A crossover refers to another crossover, while a junction refers to a <connection> element.
For each track, there can be at maximum 2 connection elements. A connection element is meant to be the begin or the end of another track. It refers either to another connection element (to connect 2 tracks) or to a switch element (which has to be a junction in this case).

Additionally, I added the attribute "branchFile" to the switch element to give the possibility to refer another railML infrastructure file in which the branch track (and its superior line) is stored.

If a switch is a crossover, there can be appended one or more <clearTrackContrElements>, which can be <trackCircuitBorders> or <axleCounters>. clearTrackContrElements can also appear as "normal" track elements (in trackData). This idea is fully adopted from the scheme of Nikolaus and covers parts of the suggestions from Volker Knollmann.
--- end switches ---

--- begin unique IDs ---

For a first approach, I've added an attribute named "uniqueId" to the elements <infrastructure>, <line>, <track>, <ocp> and all the elements in

<trackData>. The "old" IDs (lineId, trackId etc.) are kept in the scheme,
because they are intended to correspond to "real-world"-IDs.
If we really introduce these uniqe IDs, it becomes unnecessary to provide
lineId, trackId and elemId to identify an track element uniquely. But we
could leave these attributes to accelerate search in the data structure.
--- end unique IDs ---

--- begin other ---
Finally, I reintroduced the attribute "ocpId" for the element
<crossSection>, which refers to a <ocp> and I adapted the visualisation part
of the scheme according to the changes descripted above.
--- end other ---

That's it for the moment...

Kind regards,
Matthias Hengartner

Subject: Re: infrastructure_V094_13
Posted by nfries on Thu, 27 Nov 2003 15:47:06 GMT
View Forum Message <> Reply to Message

Hello back,

thanks for your effort to integrate these elements!

> --- begin switches ---
> A <switch> can be either a <junction> (Abzweigung) or a <crossover>
> (direkter Gleiswechsel). A crossover refers to another crossover, while a
> junction refers to a <connection> element.
> For each track, there can be at maximum 2 connection elements. A connection
> element is meant to be the begin or the end of another track. It refers
> either to another connection element (to connect 2 tracks) or to a switch
> element (which has to be a junction in this case).

> Additionally, I added the attribute "branchFile" to the switch element to
> give the possibility to refer another railML infrastructure file in which
> the branch track (and its superior line) is stored.

> If a switch is a crossover, there can be appended one or more
> <clearTrackContrElements>, which can be <trackCircuitBorders> or
> <axleCounters>. clearTrackContrElements can also appear as "normal" track
> elements (in trackData). This idea is fully adopted from the scheme of
> Nikolaus and covers parts of the suggestions from Volker Knollmann.
> --- end switches ---

What did you mean by "otherID"? Is it meant to replace the attribute "connSwitchID"?

> --- begin unique IDs ---
> According to the suggestions of Joachim Büchse on sept 25 about unique IDs.
> For a first approach, I've added an attribute named "uniqueId" to the
> elements <infrastructure>, <line>, <track>, <ocp> and all the elements in
> <trackData>. The "old" IDs (lineId, trackId etc.) are kept in the scheme,
> because they are intended to correspond to "real-world"-IDs.
> If we really introduce these uniqe IDs, it becomes unnecessary to provide
> lineId, trackId and elemId to identify an track element uniquely. But we
> could leave these attributes to accelerate search in the data structure.
> --- end unique IDs ---

Here we will have to define how to use them. Keeping the old IDs implies once again the danger of redundant information. Is the "uniqueID" meant to become a required attribute later on?

> --- begin other ---
> Finally, I reintroduced the attribute "ocpId" for the element
> <crossSection>, which refers to a <ocp> and I adapted the visualisation part
> of the scheme according to the changes descripted above.
> --- end other ---

However, I can cope very well with this version.
Best regards,

Nikolaus

Subject: Re: infrastructure_V094_13
Posted by Matthias Hengartner on Fri, 28 Nov 2003 11:15:52 GMT
View Forum Message <> Reply to Message

Hello again...

> What did you mean by "otherID"? Is it meant to replace the attribute
> "connSwitchID"?

How do you say "jein" in English? "Yo" or "Nes"?!
But seriously: I named this attribute "otherID" because it refers either to a connection or to another switch element. I thougt if I named it "connId" oder "connSwitchId", it would be a little bit confusing. But if we rename it, it'll be fine with me.
Alternatively, we could remove this attribute and introduce an attribute "connId" for junctions and "connSwitchId" (or "otherSwitchId") for crossovers.

By the way, what do you think about the "location" of the new <connection>-element in the scheme?
Alternatively, we could also place it in <trackData>.

> Here we will have to define how to use them. Keeping the old IDs implies
> once again the danger of redundant information. Is the "uniqueID" meant to
> become a required attribute later on?

Yes, you're right; using two different types of IDs implies redundance. But the old IDs are meant to refer to reality and are probably not globally unique (e.g. the ID of a operation control point is probably "only" unique within one specific country). Nevertheless we should keep them in the scheme for informational purposes (like other attributes, e.g. the name).
I think, "uniqueId" will become a required attribute later on.

25.9.03:

--- begin quotation
> Hence my suggestion is:
>
> RailML should use IDs (attribute with the name ID) for main elements
> like track, line etc. IDs MUST be of type string. IDs SHOULD have a
> minimal length of 8 and a maximal length of 32 symbols. Applications
> SHOULD create IDs that are globally unique. Applications SHOULD preserve
> IDs when importing and reexporting a data set with RailML. The content
> of IDs MAY be arbitrarily choosen but SHOULD NOT be semanticly
> interpreted by an application. IDs SHOULD NOT be used to order elements.
>
> Please note: I do not suggest the IDs should be used to replace
> attributes like lineId, trackId, etc in the current schema [except where
> thoose are only used to reference elements].
--- end quotation

The remaining question is, which of those two types of IDs should be used in which cases. If there should be the possibility to enter an ID (for reference, which might be the case for "ocpId" of <crossSection>), it surely would not be practical to use "uniqueId". "uniqueId" should stay an attribute for datastructure-interal use only.

Best regards,
Matthias Hengartner

Subject: Re: infrastructure_V094_13
Posted by nfries on Fri, 28 Nov 2003 14:41:36 GMT
View Forum Message <> Reply to Message

Hi Matthias,

>> What did you mean by "otherID"? Is it meant to replace the attribute
>> "connSwitchID"?

> How do you say "jein" in English? "Yo" or "Nes"?!
> But seriously: I named this attribute "otherID" because it refers either to
> a connection or to another switch element. I thougt if I named it "connId"
> oder "connSwitchId", it would be a little bit confusing. But if we rename
> it, it'll be fine with me.
> Alternatively, we could remove this attribute and introduce an attribute
> "connId" for junctions and "connSwitchId" (or "otherSwitchId") for
> crossovers.

Would it not be best to unify all possibilities of connecting two tracks
i.e. always use the <connection> element? I propose to leave it like that
except add the attribute "connDir" and change the type of "connID" to
"uniqueIdType". As well, we do not really need both - "connId" and
"uniqueID" - do we?
Additionally it will be linked to <junction> and <crossover> which allows
us to cancel "branchLineId", "otherID", "branchTrackID", "branchPos" and
"branchDir" inside the <switch> element.

> By the way, what do you think about the "location" of the new
> <connection>-element in the scheme?
> Alternatively, we could also place it in <trackData>.

No, it should be all right here.

>> Here we will have to define how to use them. Keeping the old IDs implies
>> once again the danger of redundant information. Is the "uniqueID" meant to
>> become a required attribute later on?

> Yes, you're right; using two different types of IDs implies redundance. But
> the old IDs are meant to refer to reality and are probably not globally
> unique (e.g. the ID of a operation control point is probably "only" unique
> within one specific country). Nevertheless we should keep them in the scheme
> for informational purposes (like other attributes, e.g. the name).
> I think, "uniqueId" will become a required attribute later on.

> Please have also a look on the following quotation from Joachim Büchse,
> 25.9.03:

> --- begin quotation

>> Hence my suggestion is:
>>
>> RailML should use IDs (attribute with the name ID) for main elements
>> like track, line etc. IDs MUST be of type string. IDs SHOULD have a
>> minimal length of 8 and a maximal length of 32 symbols. Applications
>> SHOULD create IDs that are globally unique. Applications SHOULD preserve
>> IDs when importing and reexporting a data set with RailML. The content
>> of IDs MAY be arbitrarily choosen but SHOULD NOT be semanticly
>> interpreted by an application. IDs SHOULD NOT be used to order elements.
>>
>> Please note: I do not suggest the IDs should be used to replace
>> attributes like lineId, trackId, etc in the current schema [except where
>> thoose are only used to reference elements].
> --- end quotation

> The remaining question is, which of those two types of IDs should be used in
> which cases. If there should be the possibility to enter an ID (for
> reference, which might be the case for "ocpId" of <crossSection>), it surely
> would not be practical to use "uniqueId". "uniqueId" should stay an
> attribute for datastructure-interal use only.

I agree. Logical references should not be replaced by structural
references ("uniqueId"), although from the structural point of view it
might define the same relation.
Have a nice weekend,

Nikolaus