Subject: Branches and connections - a neverending story
Posted by Volker Knollmann on Fri, 11 Jun 2004 09:05:36 GMT
View Forum Message <> Reply to Message

Hello,

I just was preparing some RailML-example-code and had a close look at the current schema (0.94_18) when I came across some difficulties related to switches, branches and connection. The problems refer to Matthias' posting from April 13.

Like Matthias, I don't understand the neccesarity for <singleCrossOver>. In fact, it is a switch, so why don't we use the <switch>-element? Additionally I found that using the current syntax, we always have to include two <branchConnection>-elements with the same contents (one in each <track> of the branch). This creates unneccessary redundancy.

My suggestion to reduce the complexity of branches and connections is:
* skip <singleCrossOver>
* use <switch> for every kind of branch
* rename <connections> to <switches> and make it an ordinary container element
* make <branchConnection> a child of <tracks> to include it only once per branch. Introduce an appropriate container element.

I think, the last point makes sense. Since a branch connects TWO tracks, it should NOT be a child of a track. It should be a sibling of <track>.

Here's a very simple example (ASCII-art):

```
------------------------------ Track 1 of Line A
   \        /
    \      /
------------------------------ Track 2 of Line A
```

```
<line lineID="A">
  <tracks>
    <track trackID="1" length="42.000">
      <trackTopology>
        <switches>
          <switch connectionID="SW1" pos="1.000">
          <switch connectionID="SW2" pos="1.500">
        </switches>
      </trackTopology>
    </track>
    <track trackID="2" length="42.000">
      <trackTopology>
        <switches>
```

```
        <switch connectionID="SW3" pos="1.060">
        <switch connectionID="SW4" pos="1.440">
      </switches>
    </trackTopology>
  </track>
  <branchConnections>
    <branchConnection fromElemID="SW1" toElemID="SW3"
     branchDist="0.065" />
    <branchConnection fromElemID="SW2" toElemID="SW4"
     branchDist="0.065" />
  </branchConnections>
 </tracks>
</line>
```

I skipped many attributes which are required "in real life"; my
intention was to show the idea and the structure.
Using the attribute-names "fromElemID" and "toElemID", the branch has an
implicit direction (like a vector), so the usage of the "dir"-attribute
would be possible.

So what do think? I'm looking forward to your suggestions and comments!

Best regards from Braunschweig,
Volker Knollmann

Subject: Re: Branches and connections - a neverending story
Posted by Volker Knollmann on Fri, 11 Jun 2004 09:16:58 GMT
View Forum Message <> Reply to Message

Volker Knollmann wrote:
> My suggestion to reduce the complexity of branches and connections is:
> * skip <singleCrossOver>
> * use <switch> for every kind of branch
> * rename <connections> to <switches> and make it an ordinary container
> element
> * make <branchConnection> a child of <tracks> to include it only once
> per branch. Introduce an appropriate container element.

P.S.: if there is a common interest, I would adapt the current schema to
introduce my suggestions for further discussion...

Wishing you a pleasant weekend,
Volker Knollmann

Subject: Re: Branches and connections - a neverending story
Posted by Matthias Hengartner on Fri, 11 Jun 2004 13:24:12 GMT
View Forum Message <> Reply to Message

Hello,

Thank you for your suggestions.

Some weeks ago, I discussed with Ulrich Linder the topics I mentioned in the newsgroup in early april.
We decided to remove the <singleCrossOver>-element, because it would raise unnecessary work for importing applications. So <switch> will be the only child element of <connections> for the present. Of course we could discuss about the sense and the naming of this <connections>-container.

Additionally we developed the idea of having another optional track attribute like "trackType" or similar, where we can specify what kind of track it is (e.g. "mainTrack", "crossOverTrack" or "holdingTrack").

Here a simple example (I removed attributes which are not relevant for this topic and added the attribute "trackType" to show roughly what we mean).

```
------------------------- Track 1
      \
       \
------------------------- Track 2
```

```
<track trackID="Track1" trackType="mainTrack">
   <trackTopology>
      <trackBegin>
         <bufferStop connectionID="15" pos="0"/>
      </trackBegin>
      <trackEnd>
         <bufferStop connectionID="16" pos="10"/>
      </trackEnd>
      <connections>
         <switch connectionID="20" pos="4.98000" dir="up">
            <branchConnection branchIDRef="25"/>
         </switch>
      </connections>
   </trackTopology>
</track>
<track trackID="Track2" trackType="mainTrack">
   <trackTopology>
      <trackBegin>
         <bufferStop connectionID="18" pos="0"/>
      </trackBegin>
```

```
      <trackEnd>
         <bufferStop connectionID="19" pos="10"/>
      </trackEnd>
      <connections>
         <switch connectionID="21" pos="5.00000" dir="down">
            <branchConnection branchIDRef="26"/>
         </switch>
      </connections>
   </trackTopology>
</track>
<track trackID="Track3" trackType="crossOverTrack">
   <trackTopology>
      <trackBegin>
         <simpleConnection connectionID="25" pos="0">
            <branchConnection branchIDRef="20"/>
         </simpleConnection>
      </trackBegin>
      <trackEnd>
         <simpleConnection connectionID="26" pos="0.025">
            <branchConnection branchIDRef="21"/>
         </simpleConnection>
      </trackEnd>
   </trackTopology>
</track>
```

As you can see, we have an additional third track for the crossover.

There is still some redundancy, but nevertheless, the consistency is better for the following reasons:
- if there is a switch, we have a <switch> element
- each track has a beginning and an end which is either a <bufferStop> or a <simpleConnection>
- a <simpleConnection> can have a branchConnection, which refers either to another <simpleConnection> of another track (to simply connect 2 tracks) or to a <switch> of another track (which of course means that this track begins as a branch of the other track)

So parts of your suggestions are integrated in the current development of the scheme. Your idea of simplifying crossovers with the attributes "fromElemID" and "toElemID" is good, but as I already mentioned above, we decided not to have a "special treatment" for crossover tracks but to have an separate track.

Do you agree with this idea?

Have a nice weekend,
Matthias

----- Original Message -----
From: "Volker Knollmann" <volker.knollmann@dlr.de>
Newsgroups: xml.line
Sent: Friday, June 11, 2004 11:05 AM
Subject: Branches and connections - a neverending story

> Hello,
>
> I just was preparing some RailML-example-code and had a close look at
> the current schema (0.94_18) when I came across some difficulties
> related to switches, branches and connection. The problems refer to
> Matthias' posting from April 13.
>
> Like Matthias, I don't understand the neccesarity for <singleCrossOver>.
> In fact, it is a switch, so why don't we use the <switch>-element?
> Additionally I found that using the current syntax, we always have to
> include two <branchConnection>-elements with the same contents (one in
> each <track> of the branch). This creates unneccessary redundancy.
>
> My suggestion to reduce the complexity of branches and connections is:
> * skip <singleCrossOver>
> * use <switch> for every kind of branch
> * rename <connections> to <switches> and make it an ordinary container
> element
> * make <branchConnection> a child of <tracks> to include it only once
> per branch. Introduce an appropriate container element.
>
> I think, the last point makes sense. Since a branch connects TWO tracks,
> it should NOT be a child of a track. It should be a sibling of <track>.
>
> Here's a very simple example (ASCII-art):
>
> ------------------------------ Track 1 of Line A
> \ /
> \ /
> ------------------------------ Track 2 of Line A
>
> <line lineID="A">
> <tracks>
> <track trackID="1" length="42.000">
> <trackTopology>
> <switches>

```
>           <switch connectionID="SW1" pos="1.000">
>           <switch connectionID="SW2" pos="1.500">
>         </switches>
>       </trackTopology>
>     </track>
>     <track trackID="2" length="42.000">
>       <trackTopology>
>         <switches>
>           <switch connectionID="SW3" pos="1.060">
>           <switch connectionID="SW4" pos="1.440">
>         </switches>
>       </trackTopology>
>     </track>
>     <branchConnections>
>       <branchConnection fromElemID="SW1" toElemID="SW3"
>        branchDist="0.065" />
>       <branchConnection fromElemID="SW2" toElemID="SW4"
>        branchDist="0.065" />
>     </branchConnections>
>   </tracks>
> </line>
>
> I skipped many attributes which are required "in real life"; my
> intention was to show the idea and the structure.
> Using the attribute-names "fromElemID" and "toElemID", the branch has an
> implicit direction (like a vector), so the usage of the "dir"-attribute
> would be possible.
>
>
> So what do think? I'm looking forward to your suggestions and comments!
>
> Best regards from Braunschweig,
> Volker Knollmann
```

Subject: Re: Branches and connections - a neverending story
Posted by Volker Knollmann on Mon, 14 Jun 2004 08:03:25 GMT
View Forum Message <> Reply to Message

Matthias Hengartner wrote:
> We decided to remove the <singleCrossOver>-element, because it would raise
> unnecessary work for importing applications. So <switch> will be the only
> child element of <connections> for the present. Of course we could discuss
> about the sense and the naming of this <connections>-container.

Fine! I totally agree with you that <singleCrossOver> caused only
confusion and overhead.

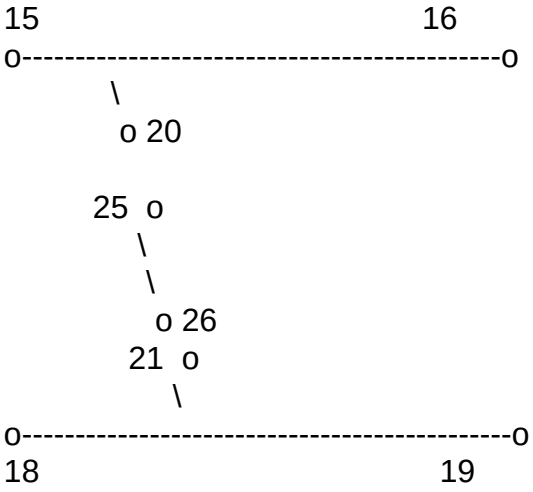
> Here a simple example (I removed attributes which are not relevant for this
> topic and added the attribute "trackType" to show roughly what we mean).
>
> [....]
>
> As you can see, we have an additional third track for the crossover.
>

Yes, this structure is much easier to understand and to implement.
And it is closer to reality: a switch is a <switch> and a track is a
<track>. So far, so good.

But this way of describing the infrastructure is very close to a
vertex-edge-graph and if I remember correctly, a
vertex-egde-representation of the track was rejected in the early stage
of the development of the schema.

Just think of edges as <track> elements and nodes as
connectionID-attributes. The connections between nodes are made via
<branchConnection>, which effectively connects to vertices (the
connectionID of the parent element and its own branchIDRef).

So here is your example with "vertices" (o), their IDs and egdes (-----):



Personally, I like vertex-edge-representations and therefore I can live
with this structure without problems (the only tricky thing about this
graph is, that a switch has only ONE vertex and not three vertices like
in normal graphs).

Additionally, some attributes of <branchConnection> make no sense
anymore (e. g. branchDist, which is given by the track length).
And: Is <branchConnection> still neccessary if the branch is a separate

track? Perhaps the information which is now stored in <branchConnection> can be merged into the parent element (either <switch> or <simpleConnection>). As an alternative, we could think of a pure <conenction>-element, which combines the functions of <branchConnection> and <simpleConnection>; this is possible, since both elements now just connect to nodes. Or we can only use <simpleConnection> and remove <branchConnection>, since effectively we only have connections between <track>-elements which should originally be handled by <simpleConnection>. Or.....

So these are my suggestions for today... don't kill me if they are too blasphemic... ;)

Best regards,
Volker
