

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.

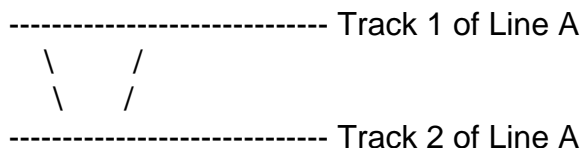
Like Matthias, I don't understand the necessity for <singleCrossOver>. In fact, it is a switch, so why don't we use the <switch>-element? Additionally I found that using the current syntax, we always have to include two <branchConnection>-elements with the same contents (one in each <track> of the branch).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



```
<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
