Subject: Reference to PositioningSystem
Posted by christian.rahmig on Mon, 12 Mar 2018 16:55:29 GMT
View Forum Message <> Reply to Message

Dear all,

reviewing the current implementation of RTM within railML 3.1 I discovered the following issue where I would like to get your feedback w.r.t. best way of implementation:

In RTM every PositioningNetElement contains at least one AssociatedPositioningSystem. The AssociatedPositioningSystem references a PositioningSystem. In railML 3.1 the matching XML syntax looks like this:

<netElement>.<associatedPositioningSystem>@positioningSystemRef

Further, an AssociatedPositioningSystem contains at least one IntrinsicCoordinate, which is typically linked with a PositioningSystemCoordinate. Each PositioningSystemCoordinate itself references a PositioningSystem. In railML 3.1 the matching XML syntax looks like this:

<associatedPositioningSystem>.<intrinsicCoordinate>.<*coordinate >@positioningSystemRef

This way of modelling seems to result in some redundancy w.r.t. referenced positioning system. Therefore, my question I would like to answer together with you: Is it really necessary having the first reference (<netElement>.<associatedPositioningSystem>@positioningSystemRef) being mandatory? I think that the second reference (<associatedPositioningSystem>.<intrinsicCoordinate>.<*coordinate >@positioningSystemRef) is sufficient. Therefore, I suggest to either remove the first reference or make it optional in the model. What do you think?

For the sake of completeness, please find here a complete example taken from the railML 3.1 Simple Example [1]:

```
<netElement id="ne_a01">
  <relation ref="nr_a01a02"/>
  <relation ref="nr_a01a03"/>
  <associatedPositioningSystem id="ne_a01_aps01" positioningSystemRef="??">
    <intrinsicCoordinate id="ne_a01_aps01_ic01" intrinsicCoord="0">
      <linearCoordinate positioningSystemRef="lps01" measure="0.0"/>
    </intrinsicCoordinate>
    <intrinsicCoordinate id="ne_a01_aps01_ic02" intrinsicCoord="1">
      <linearCoordinate positioningSystemRef="lps01" measure="500.0"/>
    </intrinsicCoordinate>
  </associatedPositioningSystem>
```

</netElement>

[1] https://www.railml.org/en/user/exampledata.html

Best regards
Christian

--
Christian Rahmig - Infrastructure scheme coordinator
railML.org (Registry of Associations: VR 5750)
Phone Coordinator: +49 173 2714509; railML.org: +49 351 47582911
Altplauen 19h; 01187 Dresden; Germany www.railml.org

Subject: Re: Reference to PositioningSystem
Posted by christian.rahmig on Wed, 04 Jul 2018 04:57:42 GMT
View Forum Message <> Reply to Message

Dear all,

although there has not been an answer on that topic so far, we need to find a solution for the problem, because it is essential for railML 3.1 and related "beta 2" version scheduled for end of August [2].

In particular, I already implemented the required RTM related change in railML 3.1. The latest version of railML 3.1 is available in the railML3 SVN trunk [3]. An overview of all the changes is provided in [4].

In this overview, removing the redundant reference to PositioningSystem from AssociatedPositioningSystem is marked as issue number 2.

[2]
https://www.railml.org/en/public-relations/news/reader/33rd-railml-conference-and-version-roadmap.html
[3] https://svn.railml.org/railML3/trunk
[4]
http://forum.railML.org/userfiles/2018-07-02_railml_railml3- induced-changes-to-rtm12.pdf

Best regards
Christian

Am 12.03.2018 um 17:55 schrieb Christian Rahmig:
> [...]
>
> In RTM every PositioningNetElement contains at least one
> AssociatedPositioningSystem. The AssociatedPositioningSystem references
> a PositioningSystem. In railML 3.1 the matching XML syntax looks like this:

>
> <netElement>.<associatedPositioningSystem>@positioningSystemRef
>
> Further, an AssociatedPositioningSystem contains at least one
> IntrinsicCoordinate, which is typically linked with a
> PositioningSystemCoordinate. Each PositioningSystemCoordinate itself
> references a PositioningSystem. In railML 3.1 the matching XML syntax
> looks like this:
>
> <associatedPositioningSystem>.<intrinsicCoordinate>.<*coordinate >@positioningSystemRef
>
>
> This way of modelling seems to result in some redundancy w.r.t.
> referenced positioning system. Therefore, my question I would like to
> answer together with you: Is it really necessary having the first
> reference
> (<netElement>.<associatedPositioningSystem>@positioningSystemRef) being
> mandatory? I think that the second reference
> (<associatedPositioningSystem>.<intrinsicCoordinate>.<*coordinate >@positioningSystemRef)
> is sufficient. Therefore, I suggest to either remove the first reference
> or make it optional in the model. What do you think?
>
> For the sake of completeness, please find here a complete example taken
> from the railML 3.1 Simple Example [1]:
>
> <netElement id="ne_a01">
> <relation ref="nr_a01a02"/>
> <relation ref="nr_a01a03"/>
> <associatedPositioningSystem id="ne_a01_aps01" positioningSystemRef="??">
> <intrinsicCoordinate id="ne_a01_aps01_ic01" intrinsicCoord="0">
> <linearCoordinate positioningSystemRef="lps01" measure="0.0"/>
> </intrinsicCoordinate>
> <intrinsicCoordinate id="ne_a01_aps01_ic02" intrinsicCoord="1">
> <linearCoordinate positioningSystemRef="lps01" measure="500.0"/>
> </intrinsicCoordinate>
> </associatedPositioningSystem>
> </netElement>
>
> [1] https://www.railml.org/en/user/exampledata.html
>
> Best regards
> Christian
>

--
Christian Rahmig - Infrastructure scheme coordinator
railML.org (Registry of Associations: VR 5750)
Phone Coordinator: +49 173 2714509; railML.org: +49 351 47582911

Altplauen 19h; 01187 Dresden; Germany www.railml.org

Subject: Re: Reference to PositioningSystem
Posted by Airy Magnien on Thu, 16 Aug 2018 09:57:10 GMT
View Forum Message <> Reply to Message

Thanks for the synthetic diagram of changes, now under scrutiny.
One minor remark:
I'd not agree with having identifiers starting with '@'. UML models are commonly used for code generation. Code is often Java or Python. Valid identifiers should be alphabetic, or underscores (to take their least common denominator), and I'd even avoid underscores, as they have a conventional meaning in Python (private scope) which is not intended here.

Why is '@' important for railML ? how can it be avoided ?

Subject: Re: Reference to PositioningSystem
Posted by christian.rahmig on Thu, 16 Aug 2018 11:42:40 GMT
View Forum Message <> Reply to Message

Dear Airy,

Am 16.08.2018 um 11:57 schrieb Airy Magnien:
> [...]
> Why is '@' important for railML ? how can it be avoided ?

The '@' is not part of the identifier, but shall just indicate that this
identifier is modelled as attribute, and not as (child) element. So,
'@name' stands for an attribute named 'name'.

Best regards
Christian

--
Christian Rahmig - Infrastructure scheme coordinator
railML.org (Registry of Associations: VR 5750)
Phone Coordinator: +49 173 2714509; railML.org: +49 351 47582911
Altplauen 19h; 01187 Dresden; Germany www.railml.org

Subject: Re: Reference to PositioningSystem
Posted by Airy Magnien on Thu, 16 Aug 2018 12:42:49 GMT
View Forum Message <> Reply to Message

Sorry, in the sketched class diagram, I took these @s "wörtlich"...

Subject: Re: Reference to PositioningSystem
Posted by Airy Magnien on Tue, 04 Sep 2018 09:34:37 GMT
View Forum Message <> Reply to Message

Back to the question. After review with RTM experts and talkback with C. Rahmig, this is where we are:
Role 'positioningSystem' of class AssociatedPositioningSystem shall be made optional (cardinality 0..1 instead of 1). This leaves the possibility to omit redundant info. However it is not clear why this redundancy would cause a genuine problem. Our compromise solution is:

- We insist that the relation shall me maintained, because it embodies the meaning of the AssociatedPositioningSystem class;
- We recommend to use this redundancy for consistency checking;
- We demand not to mix coordinates referring to different positioning systems in one same composition, because this is why the class "AssociatedPositioningSystem" has been introduced in the first place.

Subject: Re: Reference to PositioningSystem
Posted by christian.rahmig on Mon, 17 Sep 2018 11:19:43 GMT
View Forum Message <> Reply to Message

Dear all,

Am 04.09.2018 um 11:34 schrieb Airy Magnien:
> [...] After review with RTM experts and
> talkback with C. Rahmig, this is where we are:
> Role 'positioningSystem' of class
> AssociatedPositioningSystem shall be made optional
> (cardinality 0..1 instead of 1). This leaves the possibility
> to omit redundant info. However it is not clear why this
> redundancy would cause a genuine problem. Our compromise
> solution is:
>
> - We insist that the relation shall me maintained, because
> it embodies the meaning of the AssociatedPositioningSystem
> class;
> - We recommend to use this redundancy for consistency
> checking;
> - We demand not to mix coordinates referring to different
> positioning systems in one same composition, because this is
> why the class "AssociatedPositioningSystem" has been
> introduced in the first place.
>

railML.org appreciates the decision of RTM Expert Group to make the positioningSystem reference in class AssociatedPositioningSystem

optional. The resulting simplified solution has been implemented with railML 3.1 beta 2 that is available in [1].

From modelling point of view, I generally prefer solutions without redundancy in order to limit risk of inconsistencies. So, maybe the question to be answered in future is whether the positioningSystem reference shall be located in class AssociatedPositioningSystem OR in PositioningSystemCoordinate.

[1] https://svn.railml.org/railML3/tags/railML-3.1-beta2/

Best regards
Christian

--
Christian Rahmig - Infrastructure scheme coordinator
railML.org (Registry of Associations: VR 5750)
Phone Coordinator: +49 173 2714509; railML.org: +49 351 47582911
Altplauen 19h; 01187 Dresden; Germany www.railml.org