

Hi,

(English answer: see below)

- > Neben der topologischen Beschreibung, möchte ich auch den geometrischen
- > Verlauf von Strecken möglichst genau darstellen.

- > Meine Frage ist deshalb, wie Kurven am besten in railML angelegt werden
- > sollten. Momentan nutze ich dafür das <radiusChange>-Element und gebe
- > zusätzlich die Koordinaten an:
- >
- > Beispiel:
- > <radiusChange radius="1000.0" absPos="5500.00" pos="50.00"
- > name="radiusChange_1" id="1">
- > <geoCoord lat="47.0" long="11.0" />
- > </radiusChange>

Das sieht gut aus.

Das "id"-Attribut ist für die Weiterverarbeitung gedacht und somit für externe Referenzierungen wichtig. Es sollte dateiweit eindeutig sein.

Das "radius"-Attribut ist ein Längenwert in Metern (Typ "tRadiusM").
Positiver Radius heißt Linksbogen in Kilometrierungsrichtung,
negativer Radius heißt Rechtsbogen in Kilometrierungsrichtung.

!! Diese Aussage muss ich nochmals überprüfen. Sie basiert auf
Version 1.0 und wurde für 1.1 nicht dokumentiert (weder geändert
noch bestätigt).

- > Was soll die Geo-Koordinate genau beschreiben, die Mitte auf der
- > Kreisbogenlinie oder den Mittelpunkt des Kreises der die Kurve
- > beschreibt (ich verwende momentan die erste Variante)? Wie lässt
- > sich der Anfangs- und Endpunkt einer Kurve ablegen?

Bei railML definiert die Angabe <radiusChange> nur den Anfang des
Bogenwechsels in Kilometrierungsrichtung. Die Radius-Angabe gilt also
von diesem Punkt an bis zum nächsten <radiusChange> bzw. <bufferStop>.
Auf diese Weise legt der Anfang eines neuen Bogens gleichzeitig das
Ende des vorangegangenen Bogens fest.

Das trifft auch auf Geraden zu, die mathematisch als Bögen mit
unendlichem Radius aufgefasst werden können.

Da der Typ "tRadiusM" auf "xs:decimal" zurückgreift, ist der Wert "INF" (unendlich, infinity) nicht zulässig. Deshalb wird für Geraden radius="0" festgelegt.

Zusätzlich existiert ein Attribut "dir" für die Wirkungsrichtung des railML-Elements (none/up/down/both/unknown/). Dieses sollte *nicht* verwendet werden, die Bogenwechsel sollten in Kilometrierungsrichtung beschrieben werden.

Mit <geoCoord> wird der gleiche Punkt absolut beschrieben, wie bereits relativ mittels "pos" bzw. "absPos" in <radiusChange>. Somit ist er ebenfalls der Anfang eines Bogenwechsels. Strenggenommen sollte dieser Punkt auf der Gleisachse (Gleismittellinie) liegen.

Ich hoffe, einiges damit zu erklären.

Ich schicke eine Kopie an den infrastructure-Coordinator (Volker Knollmann). Möglicherweise kann er etwas dazu ergänzen.

Viel Erfolg mit der Arbeit...

Susanne

```
> -----  
>  
> I want do describe the topology and the geometry of the track as acurately  
> as possible.  
>  
> My question is, what's the best way to model curves in railML? I'm currently  
> using the <radiusChange> tag and an additional <geoCoord>.  
>  
> Example:  
> <radiusChange radius="1000.0" absPos="5500.00" pos="50.00"  
> name="radiusChange_1" id="1">  
>   <geoCoord lat="47.0" long="11.0" />  
> </radiusChange>  
>
```

That looks already nice.

"id" attribute is important for external references and should be file wide unique.

"radius" attribute is of type "tRadiusM" and describes the radius of arc in direction of mileage in metres.

"+" = left, "-" = right

!! This fact I should research by asking other railML users. It's

quite undocumented in 1.1, but was defined in 1.0.

- > Should the geo-coordinate describe the center on the arc of the circle or
- > the center of the circle which characterises the curve (I use the first
- > variant)? And how should I describe the beginning and end of a curve?

In railML you only define the beginning of a curve (arc segment) in direction of lineage. The mentioned radius applies up to the next `<radiusChange>` or `<bufferStop>`. This way also straight lines are described, they represent an arc segment with `radius=infinity`.

Type "tRadiusM" is inherited from "xs:decimal". This doesn't allow "INF" values. So straight lines are defined by `radius="0"`.

There is also an attribute "dir" (none/up/down/both/unknown/) for showing the direction of validity für railML elements. This shouldn't be used. `<radiusChange>`s are defined in direction of lineage!

Absolute coordinates in `<geoCoord>` represent the same point as the relative position (pos/absPos) of the `<radiusChange>` element. More precisely, they should reference to the point at the centerline of the track.

I'll send a copy to infrastructure coordinator (Volker Knollmann). Maybe he comments my statements.

I wish you good luck with your paper...

Susanne

--

Susanne Wunsch
Schema Coordinator: railML.common
