Subject: extension of <levelCrossing> in railML2.4nor
Posted by Janne Möller on Wed, 10 Apr 2019 09:23:15 GMT
View Forum Message <> Reply to Message

Dear all,

the element <levelCrossing> has been extended for norwegian usage. In this forum post I would like to describe the changes that were made like announced in: [1]
The following adjustments have been made which are described underneath in further detail:

- extended definition of @pos and @absPos,
- definition of new attributes @nor:roadStartPos and @nor:roadEndPos and
- no use of the attributes @length and @dir.

The attributes @pos and @absPos for <levelCrossing> are defined in the following way:
@pos: This attribute is used to store the measured centre position of the level crossing along the track. This does not necessarily have to be the actual middle between the road borders.
@absPos: This attribute is used to store the original position of the level crossing that is provided in Banedata (database on norwegian railway network).

Furthermore we introduced to new attributes which define the start and end position of the level crossing precisely. With this information the length of the element can be calculated.
@nor:roadStartPos: This attribute is used to store the measured start position of the road the railway track is crossing. It is measured along the track from <trackBegin>, similar to the attribute @pos.
@nor:roadEndPos: see above, but end position

As stated above, the length of the level crossing can be calculated by using the newly defined attributes making it therefore unnecessary to use @length.

The attribute @dir is also not used for the element. Here I refer to this forum post: [2], in which it us suggested to deprecate @dir for <levelCrossing> among other elements.

Kind regards,
Janne Möller

Subject: Re: extension of <levelCrossing> in railML2.4nor
Posted by Janne Möller on Wed, 10 Apr 2019 09:24:52 GMT
View Forum Message <> Reply to Message

[1] https://www.railml.org/forum/index.php?t=msg&goto=2051&&srch=levelCrossing#msg_2051
[2] https://www.railml.org/forum/index.php?t=msg&th=607

(Unfortunately I could not send the links in the first message due to forum restrictions)

Subject: Re: extension of <levelCrossing> in railML2.4nor
Posted by christian.rahmig on Tue, 16 Apr 2019 18:25:37 GMT
View Forum Message <> Reply to Message

Dear Janne,

Am 10.04.2019 um 11:23 schrieb Janne Möller:
> [...] the element <levelCrossing> has been extended for norwegian
> usage. In this forum post I would like to describe the
> changes that were made like announced in: [1]
> The following adjustments have been made [...]

Thank you for your input w.r.t. <levelCrossing> element. I will comment on the different proposals one by one:

> The attributes @pos and @absPos for <levelCrossing> are
> defined in the following way:
> @pos: This attribute is used to store the measured centre
> position of the level crossing along the track. This does
> not necessarily have to be the actual middle between the
> road borders.
> @absPos: This attribute is used to store the original
> position of the level crossing that is provided in Banedata
> (database on norwegian railway network).

The @pos attribute stores a measurable distance of an object from <trackBegin>. Your proposal to use the level crossing center as reference point sounds reasonable to me. What does the rest of the community think about it?

For the definition of the @absPos attribute we need to be more generic (not mentioning Banedata...). My proposal for @absPos: This is the position of the level crossing (center) in the railway line kilometer reference system (mileage). What do you think?

> Furthermore we introduced to new attributes which define the
> start and end position of the level crossing precisely. With
> this information the length of the element can be
> calculated.
> @nor:roadStartPos: This attribute is used to store the
> measured start position of the road the railway track is
> crossing. It is measured along the track from <trackBegin>,
> similar to the attribute @pos.
> @nor:roadEndPos: see above, but end position
>
> As stated above, the length of the level crossing can be
> calculated by using the newly defined attributes making it
> therefore unnecessary to use @length.

No objections against introducing the extension attributes @nor:roadStartPos and @nor:roadEndPos and their definitions, but the last statement is not correct: in railML you are free to define extensions using the "any anchor points" (element, attribute, enumeration value) as long as the content expressed in this extension is not already included in the existing railML model. Of course you can calculate a length as difference between @nor:roadStartPos and @nor:roadEndPos, but you must not leave the @length attribute empty and refer to the new attributes.

So, here is my proposal: keep your extension attributes, calculate the length from them, and then store this length information in the existing attribute <levelCrossing>@length. It might look like a redundancy to you, but for all others not following your extension proposal, the situation is clear.

From this solution we can derive the following action item w.r.t. documentation of railML element <levelCrossing>:
It has to be mentioned that @pos refers to the level crossing center and that the @length of the level crossing need to be projected 50% before the @pos and 50% behind the @pos value. This is important to mention, because this approach differs from the existing way of modelling elements with a length (e.g. tunnels). So, before we put this into a Trac ticket, let me ask the community for their opinions.

> The attribute @dir is also not used for the element. Here I
> refer to this forum post: [2], in which it us suggested to
> deprecate @dir for <levelCrossing> among other elements.

If we want to mark the @dir attribute DEPRECATED for selected elements, we have to define some rules for this approach.

My proposal: the @dir attribute represents an "application direction" describing a direction of travel, for which the element has to be considered:
* Example 1: a <signal> has a clear application direction as you can see its lights only from one side.
* Example 2: the <levelCrossing> is always there, no matter from which direction you come.
I am very interested in the feedback of the community on this approach...

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

Subject: Re: extension of <levelCrossing> in railML2.4nor
Posted by Thomas Nygreen JBD on Thu, 23 May 2019 12:37:38 GMT
View Forum Message <> Reply to Message

Dear Janne and Christian,

I agree with Christian's reservation regarding @length, and his proposal regarding @dir.

However, I would like to extend Christian's proposal for @length to allow @pos to be anywhere on the level crossing, while keeping the middle as the default. I see the following alternatives:
As today. It is not clearly defined what the reference point on a level crossing is. Any writing or reading system can define it for itself, providing flexibility, but resulting in ambiguity.

Define @pos to refer to the location of the (geometric) middle of the level crossing. Half the @length is before @pos and the other half is after. This gives clarity, but can be too unflexible for some systems. Further specification for systems that need more information is left to extensions.

Introduce one or more new attributes to allow specifying where the linear position of the level crossing. Some sub-alternatives:

A) as in railML2.4nor, with @roadStartPos and @roadEndPos, resulting in a redundancy with @length;

B) add just @roadStartPos and keep @length, so the level crossing extends from @roadStartPos to @roadStartPos + @length;

C) add an @offset (inspired by ocpTT/@offset) specifying how many metres of the @length is before @pos;

D) add a fractional @offset, e.g. 0.6 if 60 % of @length is before @pos, with a default of 0.5.

My favourite is 3C, with 2 as the fallback/default when @offset is not specified. While 3D includes a default that can be coded into the XSD, it does not operate in the same unit as the other alternatives and the sources for the information, resulting in extra calculations (and round-off errors) both when writing and when reading the information.

Best regards,
Thomas Nygreen - Common schema coordinator

Subject: Re: extension of <levelCrossing> in railML2.4nor
Posted by Torben Brand on Thu, 27 Feb 2020 12:14:19 GMT
View Forum Message <> Reply to Message

Dear comunity and expecially Christian and Thomas,

We accept the proposal made by Christian and Thomas and choose option 2 for railML2.4 and reccomend option 3C for railML2.5.

Subject: Re: extension of <levelCrossing> in railML2.4nor
Posted by christian.rahmig on Fri, 28 Feb 2020 16:00:39 GMT
View Forum Message <> Reply to Message

Dear Janne, dear all,

thank you for your feedback. I concluded the discussion in a new Trac ticket [1].

[1] [https://trac.railml.org/ticket/374]

Have a nice weekend and best regards
Christian
