
Subject: Include track length information in topology
Posted by [Martin Karlsson](#) on Thu, 17 May 2018 08:21:11 GMT
[View Forum Message](#) <> [Reply to Message](#)

The RailTopoModel contains no information about lengths of tracks. It is concerned only with the relations between the network elements, not with their physical dimensions. However, practically all use cases imaginable will need the track length information to calculate travel times, to compare with train lengths, etc. So in the railML model, it has to be included in some way.

This issue has been discussed in a previous forum post (<https://www.railml.org/forum/index.php?t=msg&th=472&start=0& amp; amp; amp;>), and the solution in version 3.1 beta was to introduce a Track object within the FunctionalInfrastructure view. The Track is mapped to a NetElement in the Topology view, but adds a length attribute. A trackside object may be mapped to a position at a distance on a Track through the child elements trackPosition and trackSection, which were added to the RTM originated location elements of spotLocation, linearLocation and areaLocation.

It has been argued that this solution adds unnecessary complexity to the model, by in effect acting as a second layer topology model, within the FunctionalInfrastructure view. This could be avoided by instead including length information in the Topology view.

A proposal discussed in the In2Rail/railML collaboration team is to Add an attribute "length" of type tLengthM to the class NetElement. NetElement is derived from the RTM class LinearElement, which remains unaltered. tLengthM indicates that the length is in meters, with decimals. The value of "length" shall be the full length between agreed reference points in the delimiting crossings, so that the aggregated length over consecutive tracks always reflects the correct total distance Add an attribute "pos" to the spotLocation child element in the Entity class. In order to do this without changing the RTM classes SpotLocation and LocatedNetEntity, new derived classes may need to be created. The value of "pos" shall be the distance from the agreed reference point in the crossing at leg 0 of the net element In the same way, add attributes "posBegin" and "posEnd" of type tLengthM to the associatedElement element used in linearLocation and areaLocation child elements of the Entity class Remove trackPosition and trackSection elements from class Entity Remove class Track Remove abstract base class TrackNode, and reparent its descendants (Switch, Crossing and BufferStop) to Entity. The TrackNode was introduced to allow a Track to refer to the functional assets at its ends, hence it is no longer needed

The new attributes should all be optional in the XSD, but the following application rules should apply:

"length" shall be mandatory in NetElements within a "Micro" description level network (i.e. representing a track), optional on other levels "pos", "posBegin" and "posEnd" may only be used in Entities referring to a NetElement with a "length" When "pos" is defined, "intrinsicCoord" may be omitted, as it's value can be derived from "pos" and "length" (same rule for posBegin/intrinsicCoordBegin and posEnd/intrinsicCoordEnd respectively)

Subject: Re: Include track length information in topology
Posted by [christian.rahmig](#) on Wed, 30 May 2018 08:57:38 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear Martin,

thank you for your extensive conclusion on this discussion. I forwarded the main question to the RTM developing team [1]. We'll see what happens...

[1] <https://www.railml.org/forum/index.php?t=msg&th=579&start=0&>

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

Am 17.05.2018 um 10:21 schrieb Martin Karlsson:

- > The RailTopoModel contains no information about lengths of
- > tracks. It is concerned only with the relations between the
- > network elements, not with their physical dimensions.
- > However, practically all use cases imaginable will need the
- > track length information to calculate travel times, to
- > compare with train lengths, etc. So in the railML model, it
- > has to be included in some way.
- >
- > This issue has been discussed in a previous forum post
- > (<https://www.railml.org/forum/index.php?t=msg&th=472&start=0&>),
- > and the solution in version 3.1 beta was to introduce a
- > Track object within the FunctionalInfrastructure view. The
- > Track is mapped to a NetElement in the Topology view, but
- > adds a length attribute. A trackside object may be mapped to
- > a position at a distance on a Track through the child
- > elements trackPosition and trackSection, which were added to
- > the RTM originated location elements of spotLocation,
- > linearLocation and areaLocation.
- >
- > It has been argued that this solution adds unnecessary
- > complexity to the model, by in effect acting as a second
- > layer topology model, within the FunctionalInfrastructure
- > view. This could be avoided by instead including length
- > information in the Topology view.
- >
- > A proposal discussed in the In2Rail/railML collaboration
- > team is to

- > Add an attribute "length" of type tLengthM to the class
- > NetElement. NetElement is derived from the RTM class
- > LinearElement, which remains unaltered. tLengthM indicates
- > that the length is in meters, with decimals. The value of
- > "length" shall be the full length between agreed reference
- > points in the delimiting crossings, so that the aggregated
- > length over consecutive tracks always reflects the correct
- > total distance Add an attribute "pos" to the spotLocation
- > child element in the Entity class. In order to do this
- > without changing the RTM classes SpotLocation and
- > LocatedNetEntity, new derived classes may need to be
- > created. The value of "pos" shall be the distance from the
- > agreed reference point in the crossing at leg 0 of the net
- > element In the same way, add attributes "posBegin" and
- > "posEnd" of type tLengthM to the associatedElement element
- > used in linearLocation and areaLocation child elements of
- > the Entity class In the same way, add attributes "posBegin"
- > and "posEnd" of type tLengthM to the associatedElement
- > element used in linearLocation and areaLocation child
- > elements of the Entity class Remove trackPosition and
- > trackSection elements from class Entity Remove class Track
- > Remove abstract base class TrackNode, and reparent its
- > descendants (Switch, Crossing and BufferStop) to Entity. The
- > TrackNode was introduced to allow a Track to refer to the
- > functional assets at its ends, hence it is no longer needed
- >
- > The new attributes should all be optional in the XSD, but
- > the following application rules should apply:
- > "length" shall be mandatory in NetElements within a "Micro"
- > description level network (i.e. representing a track),
- > optional on other levels "pos", "posBegin" and "posEnd" may
- > only be used in Entities referring to a NetElement with a
- > "length" When "pos" is defined, "intrinsicCoord" may be
- > omitted, as it's value can be derived from "pos" and
- > "length" (same rule for posBegin/intrinsicCoordBegin and
- > posEnd/intrinsicCoordEnd respectively)
- >
- >

Subject: Re: Include track length information in topology
Posted by [Martin Karlsson](#) on Mon, 15 Oct 2018 12:50:02 GMT
[View Forum Message](#) <> [Reply to Message](#)

I am pleased to see this suggestion included in version 3.1b2. However, it has not been done quite consistently. Some elements introduced in the previous approach towards this problem (as per forum post referred in the main post) have been left in the model, resulting in redundant information.

Classes SpotLocationOnTrack and LinearLocationOnTrack should be removed, along with references from class Entity to them (trackPosition and trackSection respectively).

TrackNode class should also be removed, with the consequences that

- child classes Switch, BufferStop, Border and Crossing need to be reparented from FunctionalInfrastructureEntity
- relations from Track class should be removed, i.e. trackBegin and trackEnd
- "use" relation from Track class should be removed from diagram railML3_IS_FunctionalTypes (the diagram should also be renamed to better reflect the remaining information in it)
- the whole diagram railML3_IS_Functional_Track-and-Tracknode should be removed

Another minor observation is about the naming of the new attributes, related to physical length, in class RTM_AssociatedNetElement. They are now called fromPos and toPos, whereas my suggestion was posBegin and posEnd. The difference may be insignificant, but since the corresponding intrinsic coordinate attributes are called intrinsicCoordBegin and intrinsicCoordEnd, I think my suggestion makes the connection between the two positioning views clearer.

Subject: Re: Include track length information in topology
Posted by [Thomas Nygreen JBD](#) on Tue, 08 Jan 2019 21:37:35 GMT
[View Forum Message](#) <> [Reply to Message](#)

Was there any further development on this?

One concern I have is how to interpret spotLocation/@pos if the optional netElement/@length is missing.

Subject: Re: Include track length information in topology
Posted by [christian.rahmig](#) on Fri, 18 Jan 2019 14:28:06 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,

Am 08.01.2019 um 22:37 schrieb Thomas Nygreen:

- > [...]
- > One concern I have is how to interpret spotLocation/@pos if
- > the optional netElement/@length is missing.

one possible solution is to define the rule, that if the optional attribute <spotLocation>@pos is being used, the other optional attribute <netElement>@length has to be given.

Alternatively, the attribute <netElement>@length could be declared as being mandatory.

What do you prefer?

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: Include track length information in topology
Posted by [christian.rahmig](#) on Fri, 18 Jan 2019 15:18:27 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear Martin,

Am 15.10.2018 um 14:50 schrieb Martin Karlsson:

- > [...]
- > Classes SpotLocationOnTrack and LinearLocationOnTrack should
- > be removed, along with references from class Entity to them
- > (trackPosition and trackSection respectively).

Done.

- > TrackNode class should also be removed, with the
- > consequences that
- > - child classes Switch, BufferStop, Border and Crossing need
- > to be reparented from FunctionalInfrastructureEntity
- > - relations from Track class should be removed, i.e.
- > trackBegin and trackEnd
- > - "use" relation from Track class should be removed from
- > diagram railML3_IS_FunctionalTypes (the diagram should also
- > be renamed to better reflect the remaining information in
- > it)
- > - the whole diagram
- > railML3_IS_Functional_Track-and-Tracknode should be removed

Indeed, TrackNode is an abstract class that does not provide any further content considering the derived classes. The only purpose of this class was to limit the possible references in `<track>@trackBegin` and `<track>@trackEnd`.

The parameters `<track>@trackBegin` and `<track>@trackEnd` provide a possibility to explicitly state the borders of the track (like it is done for the `<line>` segments with `@beginsInOP` and `@endsInOP`). Alternatively, the `<track>` is located like any other NetEntity derived

object and thus implicitly defines its borders.

Question to all: Do you want to have an explicit reference from the <track> to functional infrastructure elements limiting this <track>?

- > Another minor observation is about the naming of the new
- > attributes, related to physical length, in class
- > RTM_AssociatedNetElement. They are now called fromPos and
- > toPos, whereas my suggestion was posBegin and posEnd. The
- > difference may be insignificant, but since the corresponding
- > intrinsic coordinate attributes are called
- > intrinsicCoordBegin and intrinsicCoordEnd, I think my
- > suggestion makes the connection between the two positioning
- > views clearer.

Done (implemented with railML 3.1 RC2).

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: Include track length information in topology
Posted by [Thomas Nygreen JBD](#) on Mon, 21 Jan 2019 14:34:33 GMT
[View Forum Message](#) <> [Reply to Message](#)

christian.rahmig wrote on Fri, 18 January 2019 15:28
Am 08.01.2019 um 22:37 schrieb Thomas Nygreen:

- > [...]
- > One concern I have is how to interpret spotLocation/@pos if
- > the optional netElement/@length is missing.

one possible solution is to define the rule, that if the optional attribute <spotLocation>@pos is being used, the other optional attribute <netElement>@length has to be given.

Alternatively, the attribute <netElement>@length could be declared as being mandatory.

What do you prefer?

I think there could be (strategic or template) use cases where netElement length is not needed and @intrinsicCoord would still be used. Therefore, I am in favour of keeping

<netElement>@length optional and only require it when a <spotLocation>@pos is not used on that <netElement>.

There is also a possible conflict if both @intrinsicCoord and @pos are used and ($@intrinsicCoord * <netElement>@length$) calculates to something else than @pos. Should there be a rule that only one of the two attributes can be used on the same element, or should they be interpreted differently?

Subject: Re: Include track length information in topology
Posted by [christian.rahmig](#) on Mon, 21 Jan 2019 14:49:36 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear Thomas,

Am 21.01.2019 um 15:34 schrieb Thomas Nygreen:

> [...]
> There is also a possible conflict if both @intrinsicCoord
> and @pos are used and ($@intrinsicCoord * <netElement>@length$) calculates to something else than @pos.
> Should there be a rule that only one of the two attributes
> can be used on the same element, or should they be
> interpreted differently?

I suggest an order of priority:

If both attributes (@pos and @intrinsicCoord) are given, @pos is the leading one and @intrinsicCoord shall be calculated using the ratio of @pos and <netElement>@length.

Any comments from the user community?

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
