

---

Subject: [railML3] Positioning approach

Posted by [christian.rahmig](#) on Mon, 04 Sep 2023 10:49:50 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Dear all,

Positioning is an important topic, also in railML 3. However, the railML 3.2 data model and schema basically provides different options for specifying a position in a certain coordinate reference system and/or the topology network. This variety of options results from the schema syntax where most elements and attributes are optional. While this variety is very comfortable for a railML export interface (because you can export whatever suits you best), it causes problems on the side of the railML importer, which need to be able to understand all the different options in the same manner and be able to convert one solution into the other one without any ambiguities.

Considering this challenge, we discussed the positioning topic in our different use case working groups (SCTP, NEST, ETCS) and agreed on a proposal for a ruling framework that limits the variety of how to model positions in railML 3.x. This proposal is documented in the slides [1]. This presentation is in a working draft stage and we would like to collect feedback of the community.

The suggested approach summarized in three points:

(1) Every NetElement has to have an associatedPositioningSystem linking to a so-called "intrinsic positioning system" usable as measuring tape. This associatedPositioningSystem realizes the mapping between the intrinsic coordinates (0..1) and the measuring tape.

(2) Every located functional infrastructure element need to have at least one spotLocation or linearLocation or areaLocation.

(3a) For a spotLocation: define the position either with @intrinsicCoord or linearCoordinate@measure referencing the intrinsic positioning system.

(3b) For a linearLocation: define the position either with @intrinsicCoordBegin / @intrinsicCoordEnd or linearCoordinateBegin@measure / linearCoordinateEnd@measure referencing the intrinsic positioning system.

So, after we already got first feedback from the railML use case working groups, I want to ask the whole railML community to provide us with their thoughts, comments and ideas, directly here in the forum. Please focus on:

- \* Is the approach understandable (also in its documentation)?
- \* Do you agree with the approach?
- \* Is there something missing?

As usual, any kind of feedback is highly appreciated...

[1] <https://cloud.railml.org/s/m8JGaBsARjmoD2m>

Best regards  
Christian

---