Subject: [railML 3] Areas in railML 3
Posted by Thomas Langkamm on Tue, 21 Sep 2021 10:09:15 GMT
View Forum Message <> Reply to Message

Dear all,

we have discussed the introduction of areas in railML 2 (https://www.railml.org/forum/index.php?t=msg&th=813, https://trac.railml.org/ticket/393). In the IS working group we had some discussions on whether we should introduce a similar concept in railML 3. More precisely:

railML 3 has 2 existing concepts which could be used to define an area, which are (1) IS:operationalPoint and (2) aggregation if netElements in higher level topologies (that is, a netElement in a macroscopic topology is an aggregation of microscopic netElements). However, the netElement aggregation is probably not a good tool to model areas, because we may have many different types of areas (see the list in the ticket linked above) and we would have to subdivide netElements whenever an area does not contain all of the netElement. This could lead to a combinatorial explosion with a huge number of netElements.

So basically, we have the option to use operationalPoint or to introduce a new concept similar to the one we use in railML 2.5. Personally, I think it would be a good idea to mirror the changes from railML 2.5 in railML 3 and introduce a new area object, for the following reasons:

operationalPoint is a very generic concept. Although this gives us a high degree of flexibility, in my experience it's better to have concepts that are a bit less generic as it makes the model more descriptive and consistent. Basically a very generic concept will probably lead to a large number of very different uses and perhaps different interpretations on how to use it.

Many of the areas that we discussed are well defined, for example the areas used by an interlocking to display train occupations, or the part of a network that is controlled by one interlocking. I believe it would help the schema if we would use more specific types here instead of a very generic type.

Some other areas are only references to external software systems, whose data we don't want to model in railML. An example would be maintainance systems, where we have track areas allowing certain types of work or access to certain machines. Here we need to define the part of the rail network belonging to these areas, but the only other additional data would be a reference to the external software system.

Looking at the schema, a minimal and generic definition of an area would be basically the use of the location types (areaLocation, linearLocation and spotLocation) plus the optional "external" reference, without most of the other attributes that we can add to operationalPoint.

I don't have a strong preference on whether we should introduce additional types for specific areas (like information area, track sections used in interlockings or the parts of the network controlled by an interlocking), or if we should add a type attribute to the area definition. What do you think?

Subject: Re: [railML 3] Areas in railML 3
Posted by Fabiana Diotallevi on Mon, 27 Sep 2021 10:12:51 GMT
View Forum Message <> Reply to Message

Dear Thomas,

I agree with you that there is the need to define, from a topological point of view, the "area" concept, that could be used and referred to from many different "tags".

As you said, to identify an area we could use the <areaLocation> tag, because it is basically a collection of netElements. However, I think that we should improve such definition for two main reasons:

1. In the current <areaLocation> definition, you don't have the possibility to define multiple intervals spanning the same netElement (if you want for example to define a "U"-shaped area), since you only have the possibility to specify a starting and a ending position on one netelement.

2. I think that it is important to define areas in the <topology> section (i.e. at the same level of the neteElements) so that any infrastructure or interlocking element could refer just the areaId without re-definig its location.

If we stick just to the topological definition of an area, we don't need to define a "type" attribute. However, in this case we may have to introduce a more general "zone" object, with a "type" attribute, an "external" reference and an "areaId" reference.

Subject: Re: [railML 3] Areas in railML 3
Posted by christian.rahmig on Mon, 11 Oct 2021 10:37:23 GMT
View Forum Message <> Reply to Message

Dear Fabiana, dear Thomas,

thank you for your input and ideas regarding modelling specific areas with railML 3.x. I agree with you to have a look at railML 2.5 implementation of <genericArea> first [1] and evaluate how to adapt it for railML 3.x.

Here comes my proposal:
* a new "view" is added to <infrastructure>: it is named <genericLocations>
* within <genericLocations> an arbitrary number of <genericArea> elements may exist
* a <genericArea> has an ID, name and designator attributes / child elements
* a <genericArea> contains child elements for specifying a location, e.g. as <circle> or <polygon>
* like any functional infrastructure element, the <genericArea> shall have child elements to be located within the topology network using <spotLocation>, <linearLocation> and/or <areaLocation>
* the <genericArea> may reference bounding elements using <isLimitedBy> references pointing to

functional infrastructure elements
* the specific purpose of the generic area comes from outside via the specific elements pointing to the <genericArea>, e.g. an <etcsArea>

What does the community think about this approach?

[1] https://trac.railml.org/ticket/393

Best regards
Christian

Subject: Re: [railML 3] Areas in railML 3
Posted by Thomas Langkamm on Tue, 12 Oct 2021 12:42:20 GMT
View Forum Message <> Reply to Message

I like the idea. The only change I would like to suggest is a typization, as this would allow the users to distinguish different area types.

In general, if we reference areas required by other software systems (for example, work zones for a maintainance system, that defines which track areas allow certain types of work), we would use the designator as a way to pass references to this other software system. But it might make the structure easier (for a human to read, and for a machine to analyze) if we had an optional "areaType" string attribute, perhaps with some standard entries (as enum that could be extended by the user).

Some area types that I would consider as fairly common, so we might want to define a standard type:

Interlocking areas. This would be strictly part of the rail network as follows:
Influence area, the area controlled by one interlocking
Track sections, the "atoms" of the interlocking configuration where the interlocking shows/tracks occupations (a train currently in the region, or a train scheduled to enter the region). In many cases these will be identical to tvdSections, but there may be instances (and interlockings) where we group several tvdSections to one track section, or split a tvdSection into two or more track section (on account of having another train detection element in the tvdSection).
Areas that can be used in interlocking operations, for example an area that could be controlled by one "shutdown" button (used to set all signals to red, for example used if some unauthorized humans are observed in the track area).
Other interlocking-derived areas listed in the ticket https://trac.railml.org/ticket/393

Track regions belonging to different states, federal provinces or counties. These are typically needed to track commercial differences (like different penalties for delays or cancelled train journeys) or different legal requirements. In duty scheduling we might have different duty rules depending on the state.

Areas that allow a certain type of maintainance work, often as part of a maintainance system.

Areas grouped for a specific use in train operations. For example, we may have a group of tracks designated for overnight parking, while another group of tracks might be used as temporary storage for trains that need to be shunted to a different location (like a workshop) before their next journey starts.

GPS areas. For example, we could identify a train as "being near a station" if its GPS coordinates are in a certain circle or rectangle.

Subject: Re: [railML 3] Areas in railML 3
Posted by Jörg von Lingen on Sun, 17 Oct 2021 04:07:23 GMT
View Forum Message <> Reply to Message

Dear all,

it looks a good idea to bring area objects on the topology level defined by netElements and combing them with general attributes. However, our railML model is built up that IL elements link to IS elements which are linking to netElements in topology but not vice versa. If we now have only the netElements for defining the constrains of an area, we would need a clever way to search for the functional elements contained within. Especially for (conventional) interlocking purpose the location of elements in terms of km x.y or even coordinates are less important.

Interlocking areas: It is not that easy to bring them all together as they have different functions behind and therefore different needs to be considered. Quite often there is the issue of defining the protection from the inside and the outside. Sometimes we need to define the status of elements within, e.g. released for local operation or not. These are all functional needs, which cannot be defined in the infrastructure itself.

ETCS is a kind of Janus head. It uses interlocking functions without really considering the normal interlocking elements. Thus it is more focused on some infrastructure features like exact location.

We may have a similar way of defining the basics of an area but I don't think we can press them in one <genericArea> element.

Best regards,
Joerg v. Lingen - Interlocking Coordinator

Subject: Re: [railML 3] Areas in railML 3
Posted by christian.rahmig on Thu, 18 Nov 2021 13:03:14 GMT

Dear Thomas, Fabiana and Jörg,
dear all,

thank you for your feedback so far. I created an issue for the topic in our railML 3 development environment Git [1].

Further, I would like to encourage the community to provide further feedback; especially on the topic of area typization as described by Thomas and Jörg.

[1] https://development.railml.org/railml/version3/-/issues/479

Thank you very much and best regards
Christian

Subject: Re: [railML 3] Areas in railML 3
Posted by Morten Johansen on Wed, 26 Apr 2023 13:30:25 GMT

Dear all.

I strongly support Thomas' view and arguments, especially regarding the handling of areas in systems recieving the railML file, for having the possibility to give areas type information on the genericArea element.

We have in addition, when trying to express our needs for areas in our ETCS project in railML3.2, identified a need to be able to link a detector element to the genericArea element representing the area that is made not approachable when the detector is triggered.

Best regards
Morten Johansen
Enterprise architect, information architecture
Bane NOR