Subject: [railML v3.3 wiki] Ambiguous reference direction of track@mainDirection
Posted by Mathias Vanden Auweele on Wed, 29 Jul 2026 21:19:03 GMT
View Forum Message <> Reply to Message

Hello,

I would like to raise a potential inconsistency in the definition of track@mainDirection.

Wiki page:
https://wiki3.railml.org/wiki/IS:track/3.3

The documentation defines @mainDirection as follows:

Quote:mainDirection: predominant direction of operation on this track;
use this attribute to define the default direction of train operations, especially on double track lines
(optional: tExtendedDirection : xs:string)
Possible values:
- both: from intrinsicCoordinate=0 to intrinsicCoordinate=1 AND from intrinsicCoordinate=1 to intrinsicCoordinate=0 of associated <netElement>
- reverse: from intrinsicCoordinate=1 to intrinsicCoordinate=0 of associated <netElement>
- normal: from intrinsicCoordinate=0 to intrinsicCoordinate=1 of associated <netElement>,

The issue is that a <track> is not necessarily associated with one unique <netElement>. A track usually has one or more <linearLocation> elements, and each linear location may contain an ordered sequence of associatedNetElements.

The order of the associatedNetElements defines the direction of the linear location. However, each referenced net element may have its own intrinsic orientation, independently of the direction of the linear location.

For example, consider a linear location that traverses three net elements in this order:

Linear-location direction: A -> B -> C

AssociatedNetElement @seq=1: netelement@id=A, @beginIntrinsicCoordinate = 0 -> @endIntrinsicCoordinate = 1
AssociatedNetElement @seq=2: netelement@id=B, @beginIntrinsicCoordinate = 1 -> @endIntrinsicCoordinate = 0
AssociatedNetElement @seq=3: netelement@id=C, @beginIntrinsicCoordinate = 0 -> @endIntrinsicCoordinate = 1

A train travelling in the normal direction of the linear location would therefore traverse:

A: 0 -> 1

B: 1 -> 0
C: 0 -> 1

In this case, the train's direction cannot consistently be described as either 0 -> 1 or 1 -> 0 on "the associated <netElement>". The interpretation depends on which associated netelement is being considered.

There is a second ambiguity: a <track> may have multiple <linearLocation> elements, for example because it is represented:

- in different positioning systems;
- at different topology levels; or
- by alternative network representations.

Consequently, even if normal and reverse are interpreted relative to the direction of a linear location, it is still necessary to identify which <linearLocation> provides the reference direction.

Semantic requirements

To make @mainDirection unambiguous, its definition should:
- define normal and reverse relative to the direction of a <linearLocation>, rather than relative to the intrinsic orientation of an individual <netElement>; and
- identify the <linearLocation> that provides the reference direction if the track has more than one linear location.

Possible solution

A backward-compatible solution could be to retain @mainDirection and add a reference to the linear location that defines its orientation. Conceptually, this could look like:

```
<track
   id="track01"
   mainDirection="normal"
   mainDirectionLocationRef="linearLocation01">

   <linearLocation id="linearLocation01">
      ...
   </linearLocation>

   <linearLocation id="linearLocation02">
      ...
   </linearLocation>
</track>
```

The values could then be defined as:
- normal: predominant train operation follows the ordered direction of the referenced <linearLocation>;

- reverse: predominant train operation runs against the ordered direction of the referenced <linearLocation>;
- both: neither direction is predominant, or both directions are normally used.

A better approach in my opinion is to use a dedicated child element, for example:

```
<track id="track01">
   <mainDirection
      value="normal"
      linearLocationRef="linearLocation01"/>
   ...
</track>
```

By allowing multiple <mainDirection>, this representation would allow expressing different main directions for different topology levels or positioning systems.

Why linearLocation@applicableDirection is not sufficient

Another possible approach would be to remove track@mainDirection and use linearLocation@applicableDirection instead. However, I do not think these attributes have the same semantics.

applicableDirection appears to describe whether a location or located object applies in a particular direction. In contrast, mainDirection describes the predominant or default direction of train operation.

For example, if a track has:

```
<linearLocation applicableDirection="normal">
   ...
</linearLocation>
```

this may imply that the location applies only in the normal direction. That is different from stating that the track itself exists and is usable in both directions, while normal direction is the predominant direction of operation.

A train travelling in the reverse direction should therefore not make the track or its location inapplicable. It is merely travelling against the track's main direction.

For this reason, I suggest keeping mainDirection as a separate operational characteristic, but defining it relative to an explicitly identified <linearLocation>.

What do you think?