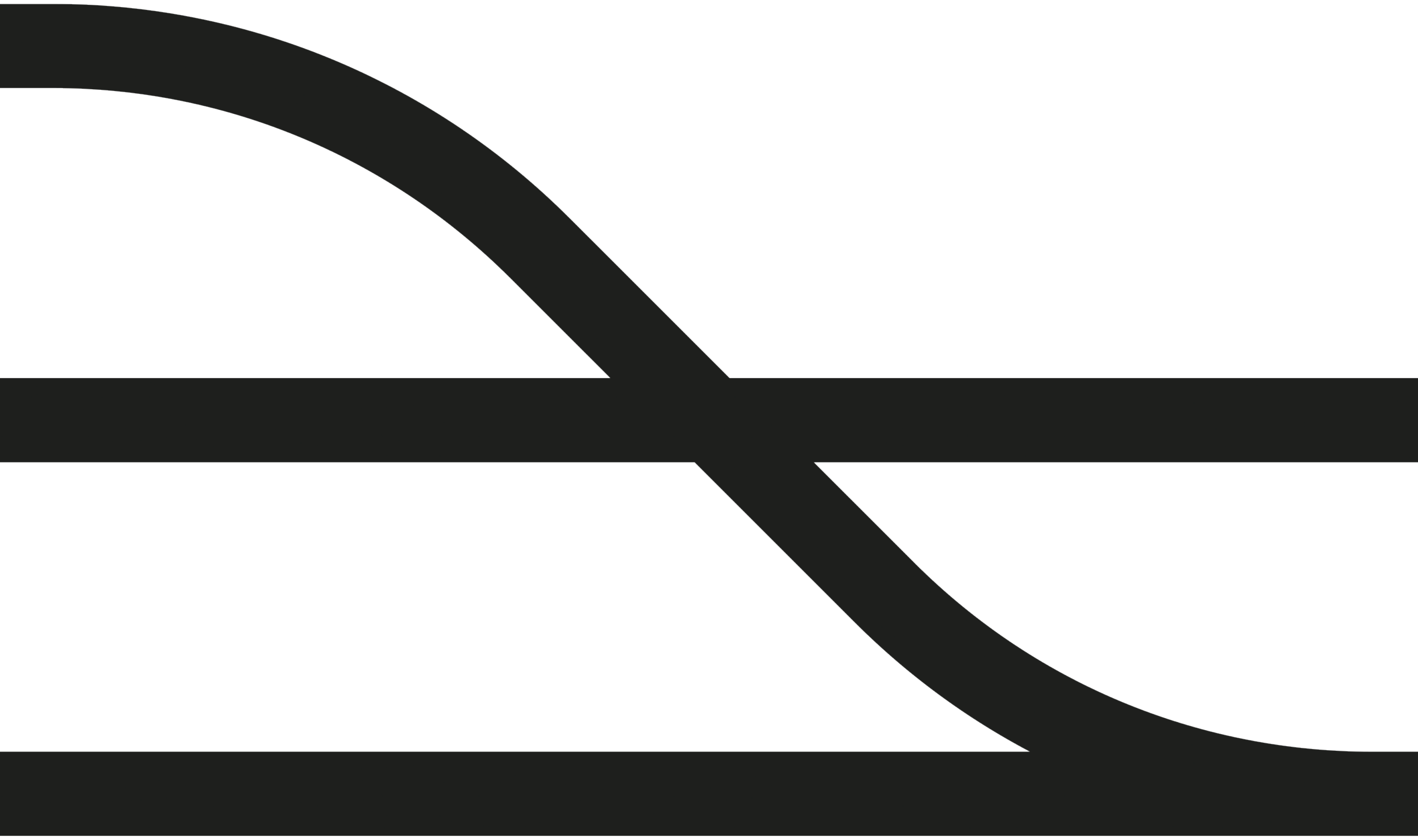
Jernbane-
direktoratet

railML2.4nor Documentation

Subschema Infrastructure



Version: 1.6
Date: 30.03.2023

Summary

The following documents the railML2.4nor Infrastructure schema, especially the Norwegian specialties and resulting extensions to railML2.4. It contains information about the application of railML2.4nor Infrastructure, general modelling rules and an example (track plan visualisation and corresponding railML2.4nor source code).

Content

1 General Information 6
1.1 What is railML2.4nor 6
1.2 Why do we need railML2.4nor 6
1.3 What do we use railML2.4nor Infrastructure for 7
1.4 Reader information 7
1.5 Syntax guide 8
1.6 Versioning of railML2.4nor Infrastructure developments 8
1.7 Versioning of railML2.4nor Infrastructure documentation 9

2 railML2.4nor Infrastructure content on element level 11

3 General Modelling Rules 13
3.1 Main language 13
3.2 Missing information and default values 13
3.3 Conventions 14
3.3.1 New elements and attributes 14
3.3.2 Norwegian extensions of enumerations using other:* 14
3.3.3 Exporting unmappable values in enumerations 14
3.4 Usage of deprecated attributes 15
3.5 Mandatory elements 15
3.6 Container elements 15
3.7 Common attributes 15
3.8 Position of elements 16
3.9 Orientation of elements 17
3.9.1 Extent of elements with a length attribute 17
3.9.2 Directional validity of elements 17
3.9.3 Horizontal placement of element (@side) 18
3.10 Aggregation level of track model 18
3.10.1 Microscopic model 19
3.10.2 Macroscopic modelling 21
3.11 Mandatory placement of "<*Change>" elements 22
3.11.1 Flow of "<*Change>" elements along the track(s) 22
3.12 Inheritance 23
3.13 Project model 24
3.14 Connection of models 24
3.14.1 Connection between models in the same <nor:project> 24
3.14.2 Connection between models in different <nor:project> 24
3.15 Time dimension 25
3.15.1 The @change attribute 25

4 Element Specific Definitions and Extensions 27
4.1 The element <track> 27
4.1.1 Definition extension in accordance to Norwegian usage 27
4.1.2 Code example 29
4.2 The element <ocp> - operational control point 29
4.2.1 Definition extension in accordance to Norwegian usage 29
4.2.2 Sub-element extension 32
4.2.3 Code example 32
4.3 The elements <switch> and <crossing> 32
4.3.1 Definition extension in accordance to Norwegian usage 32
4.3.2 Attribute extension 35
4.3.3 Code example 36
4.4 The element <signal> - modelling of boards and signals 37

4.4.1 Definition extension in accordance to Norwegian usage 37
4.4.2 Attribute extension 38
4.4.3 Code example 38
4.5 The element <controller> 38
4.5.1 Definition extension in accordance to Norwegian usage 38
4.5.2 Attribute extension 39
4.5.3 Code example 39
4.6 The element <baliseGroup> 39
4.6.1 Definition extension in accordance to Norwegian usage 39
4.6.2 Attribute extension 41
4.7 The elements <speedProfile> and <speedChange> - modelling of speed 41
4.7.1 Definition extension in accordance to Norwegian usage 41
4.7.2 Code example 46
4.8 The element <nor:route> 46
4.8.1 Definition in accordance to Norwegian usage 46
4.8.2 Sub-elements of the element 47
4.8.3 Attributes of the element 49
4.8.4 Shortened routes 51
4.8.5 Code example 52
4.9 The element <nor:area> 52
4.9.1 Definition 52
4.9.2 Attributes of the element 53
4.9.3 Sub-elements of the element 53
4.9.4 Code example 54
4.10 The element <trainDetector> 54
4.10.1 Definition extension in accordance to Norwegian usage 54
4.10.2 Definition extension in accordance to Norwegian usage 54
4.10.3 Code example 54
4.11 The element <trackCircuitBorder> 54
4.11.1 Definition extension in accordance to Norwegian usage 54
4.11.2 Code example 54
4.12 The element <border> 54
4.12.1 Definition in accordance of Norwegian usage 54
4.12.2 Attribute extension 54
4.13 The element <nor:project> 55
4.13.1 Definition 55
4.13.2 Attributes of the element 55
4.13.3 Sub-elements of the element 55
4.13.4 Code example 58
4.14 The element <state> 58
4.14.1 Definition in accordance of Norwegian usage 58
4.14.2 Attribute usage and extension 59
4.14.3 Subelement <nor:additionalRunningTime> 60
4.14.4 Code example 60
4.15 The element <gradientChange> 60
4.15.1 Definition extension in accordance to Norwegian usage 60
4.15.2 Code example 61
4.16 The element <tunnel> 61
4.16.1 Definition extension in accordance to Norwegian usage 61
4.16.2 Attribute extension 61
4.16.3 Code example 61
4.17 The element <stopPost> 61
4.17.1 Definition extension in accordance to Norwegian usage 61
4.17.2 Code example 62
4.18 The element <nor:lineSection> 62
4.18.1 Definition 62

4.18.2 Attributes of the element 62
4.18.3 Code example 62
4.19 The element <derailer> 62
4.19.1 Attribute usage and extension 63
4.19.2 Code example 63
4.20 The element <trainProtectionChange> 63
4.20.1 Attribute extension 63
4.21 The element <lock> 63
4.21.1 Definition in accordance to Norwegian usage 63
4.21.2 Attributes of the element 63
4.21.3 Code example 64
4.22 The element <electrificationChange> 64
4.22.1 Definition extension in accordance to Norwegian usage 64
4.23 The element <levelCrossing> 64
4.23.1 Definition extension in accordance to Norwegian usage 64
4.23.2 Attributes of the element 64
4.23.3 Code example 65
4.24 The element <infrastructureVisualizations> 65
4.24.1 Definition in accordance to Norwegian usage 65
4.24.2 Code example 66
4.25 The element <geoMapping> 66
4.25.1 Definition extension in accordance to Norwegian usage 66
4.25.2 Code example 66
4.26 The element <brigde> 66
4.27 The element <crossedElement> 66
4.28 The element <nor:operatingRules> 67
4.28.1 Attributes of the element 67
4.28.2 Code example 67
4.29 The element <uptime> 67
4.29.1 Attribute extension 68
4.29.2 Code example 68
4.30 The element <serviceSection> 68
4.30.1 Code example 69

5 railML2.4nor Examples 70
5.1 “Simplest Example” 70
5.2 “Norwegian Simple Example” 71
5.3 "BB excerpt simple mapping" 71
5.4 Source code of “Simplest Example” 72

6 References 87

1 General Information

1.1 What is railML2.4nor

railML2.4nor is an extension of the railway data exchange schema railML version 2.4, addressing specific Norwegian requirements. railML – Railway Markup Language – is an open-source XML based data exchange format, which shall enable an easy communication of heterogenous railway applications. Today, the connection of various railway software packages is beset with problems. The purpose of the railML.org initiative has been to find, discuss and present systematic, XML-based solutions for simplified and standardized data exchange between railway applications. The schema is developed by railML.org, a registered association in Germany since 2002, in close cooperation with all interested institutions and businesses of the railway sector across Europe.[1] railML2.4 consists of three subschemas: Infrastructure, Rollingstock and Timetable.

railML2.4nor is a microscopic model, which consists of a proportion of railML2.4 and adds extensions to meet Norwegian requirements. railML2.4nor does not contain all railML2.4 elements, as not everything in railML2.4 is of relevance for Norwegian railways. railML2.4nor precisely defines rules and conventions - in accordance to Norwegian requirements - for adopted ambiguous elements/attributes, where existing railML documentation is vague or open to interpretation. In other words: railML2.4nor is the Norwegian usage of railML2.4 plus Norwegian extensions. The following diagram shows the relation of railML2.4 and railML2.4nor.

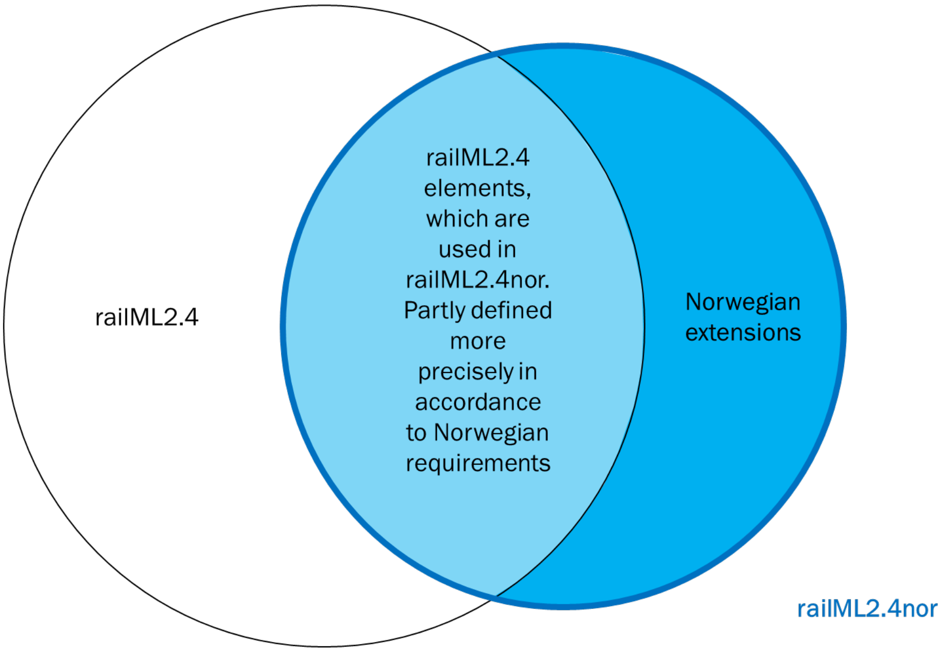

railML2.4
railML2.4 elements, which are used in railML2.4nor. Partly defined more precisely in accordance to Norwegian requirements
Norwegian extensions
railML2.4nor

Figure 1: Relation of railML2.4 and railML2.4nor

1.2 Why do we need railML2.4nor

railML is a multinational development and aims at modelling the reality in the most generic way possible. This requires coordination and discussion with numerous stakeholders and hence is a time-consuming process. Furthermore, its generic characteristic does not allow the implementation of all national requirements. As a result, the railML schema in its original form, currently does not fit Norwegian

[1] (railML.org, 2018)

requirements sufficiently for productive use. Due to this, the Norwegian Sector decided to adjust the schema to meet Norwegian requirements in an extension. The Norwegian sector consists of all companies who work within the railway industry in Norway. Primarily in charge of the development and distribution of railML2.4nor is the Norwegian Railway Directorate Jernbanedirektoratet and the Norwegian infrastructure manager BaneNOR.

The Norwegian sector's aim is to create a data exchange format, which serves as exchange format between various railway applications, tailored to Norwegian requirements.

1.3 What do we use railML2.4nor Infrastructure for

railML2.4nor Infrastructure (IS) shall serve as standard data exchange format between the Norwegian infrastructure data base Trasé and all applied IT applications. It serves both, import and export services. It is developed in accordance to the requirements of the following use cases and shall be used by the Norwegian sector as the general data exchange format in these areas. Information about the scope of the use cases is provided in the railML Wiki (links provided as hyperlink).

- Capacity Planning (simulation)
- Network Statement of an Infrastructure Manager - NEST
- Schematic Track Plan – SCTP
- Routes for Timetable Simulation – RSIM

1.4 Reader information

This document is a documentation of the subschema Infrastructure of the railML2.4nor data exchange format, which is hereafter abbreviated by IS. Unless stated otherwise, the term railML2.4nor stands for railML2.4nor IS in this specification and railML2.4 for railML2.4 IS. The documentation of the schemas railML2.4nor Rollingstock and Timetable are available in a separate document.

This documentation contains general rules and definitions of how ambiguously defined core railML2.4 elements are interpreted and modelled in Norway. Furthermore, it addresses Norwegian extensions that were introduced in order to meet national requirements. The Norwegian extensions are always recognizable in the documentation by their "nor:" prefix, e.g. <nor:area>.

Please note that this document is only a supplement to the documentation of the core railML schema. The documentation of the core railML schema is provided by railML.org and can be found in the following places:

- railML Wiki: Contains information about schema application
- railML Forum: Platform for discussions with railML users and the developer community
- railML Trac Ticket System: Platform to record and track all bugs and model enhancements
- railML.org website: Contains general information about the organisation of railML.org, the development of the schema, download of the schema and example data
- railML XSDs

In addition to the general modelling rules, this document provides an example of railML2.4nor. It consists of track plan visualisation and the corresponding railML2.4nor source code.

Furthermore, the Norwegian sector provides an Excel sheet - railML2.4nor Infrastructure Model - listing all railML2.4nor elements with their attributes, values and specific definitions. This Excel sheet contains all used railML2.4 core elements with their attributes and values, and the extension elements with their attributes and value lists. Please additionally have a look at this Excel sheet, it is the complete overview of all elements, attributes and values of the railML2.4nor schema.

The official XSD of railML2.4nor can be found on the Jernbanedirektoratet website.

1.5 Syntax guide

In the text, railML <elements> are put into XML specific brackets <>. railML @attributes can be recognized via the @ symbol before the attribute name. The combination of element and attribute is notated <element>@code. When specifying a parent- and a child-element, the syntax is <parent><child>. Attribute "values" are framed by quotation marks "".

Source code examples are written in grey boxes:

```
<railml sourcecode="example">
   ...
</railml>
```

Some information in this document is written in *italic letters*. This means that the information is regarded as additional background information.

1.6 Versioning of railML2.4nor Infrastructure developments

railML2.4nor is developed in baselines. The term baseline refers to versions, which are published officially. They consist of further Norwegian specialties, which have been modelled and integrated in railML2.4nor step by step.

A mature version of railML2.4nor is reached with Baseline2. The enhancement work will be stopped after the publication of the official XSD and corresponding documentation. Baseline 3 is seen as a collective wish list for the future. The Norwegian sector's aim is to shift to railML3.x once this standard is more mature and approved. Should a future need be so urgent that it cannot wait for railML3.x, the Norwegian sector might consider developing railML2.4nor Baseline3.

Version	Release date of railML2.4nor version	Based on railML core version	Corresponding railML2.4nor XSD version
railML2.3nor Baseline1	01.11.2018	railML2.3	Version 3.2
railML2.4nor Baseline2 version 1.0	20.12.2018	railML2.4	Version 5.0
railML2.4nor Baseline2 version 1.1	04.03.2019	railML2.4	Version 5.0
railML2.4nor Baseline2 version 1.2	02.07.2019	railML2.4	Version 5.0
railML2.4nor Baseline2 version 1.3	03.07.2020	railML2.4	Version 5.1 and later version 5.2
railML2.4nor Baseline2 version 1.4	17.12.2020	railML2.4	Version 5.3 and later version 5.4 published 11.06.2021
railML2.4nor Baseline2 version 1.5	17.06.2022	railML2.4	Version 5.5
railML2.4nor Baseline2 version 1.6	30.3.2023	railML2.4	Version 5.5
railML2.4nor Baseline3	Is cancelled due to roadmap to switch to railML3.2 or later after railML2.4nor Baseline 2		

As stated in the table above, railML2.3nor Baseline1 was based on railML2.3. As railML.org published a new version – railML2.4 - on 07.10.2018 during the work with the nor extensions, the Norwegian sector decided to migrate its nor-extension standard and use cases to railML2.4. Thus, Baseline2 is based on the current version railML2.4.

Note many of the extensions of railML2.4nor are included in railML2.5 which is also the foundation for the international standard ISO TS/4398:2022 “RailDax”.

1.7 Versioning of railML2.4nor Infrastructure documentation

For comprehensibility of changes in the railML2.4nor documentation the document has an ongoing version number. Current and previous versions are stated in the table below including a short description of the modifications with regard to the previous version.

Version	Release date of railML2.4nor documentation	Comments
1.0	20.12.2018	
1.1	14.03.2019	Version 1.1 of the documentation contains extended and specified descriptions of railML2.4nor. No major changes have been made in this version.
1.2	15.07.2019	Updated with extended "simplest example", "simple example Nor" and "operational RHB" example with full baseline 2 features.
1.3	03.07.2020	Contains minor changes in descriptions. Description of main language added. New extension for <state> element. New element <nor:operatingRule> introduced. Attribute @flankProtection introduced.
1.4	17.12.2020	Extension of <uptime> with @nor:operatingPeriodRef and @nor:endDayOffset. Minor changes in descriptions.
1.5	17.06.2022	More detailed description/specified use of: • Revision of description of ocp types (section 4.2.1) • Mapping of <state>@status values with Banedata (section 4.14.2) • How to describe the type of derailer for the Norwegian railway sector (section 4.19.1) • Use of values for <brigde>@kind (section 4.26) • Use of values for <crossedElement>@type (section 4.27) New attributes: • <derailer>@nor:preferredPosition (section 4.19.1) • <lock>@nor:trackDist (section 4.21.2) • The element <serviceSection> (section 4.30)

Version	Release date of railML2.4nor documentation	Comments
1.6	31.03.2023	More detailed description/specified use of: • 3.3.3 rewritten chapter on extensions for clarification • 3.10.1 revision of the switch model for 2 switches meeting stock rail to stock rail at the same absPos • 3.10.2 added note to contact before use of macro model. • 3.14.2 revision of use of border@type for projects • 4.7 added text explaining ending speed bands • 4.8.2 revision of use of <nor:switchandPosition>@switchPosition with illustration, table and example code Corrected text on <nor:state**s**> • 4.9 Added note how to refer to <border> and how to use @nor:projectRef • 4.9.3 Corrected text on <nor:state**s**> • 4.12.1 added more precise text on border without area. • 4.14.2 revision of use of nor:restrictionID and @status for restriction sections. Added note on difference to <nor:states>

2 railML2.4nor Infrastructure content on element level

The following hierarchy shows all elements railML2.4nor contains. It does not address attributes and values of these elements. Please note that the hierarchy is a simplification of the model and does not show container elements or elements from core railML that are not used in Norway. It is only meant to illustrate the content and does not display the correct syntax. A full list of all elements, attributes and values is given in the Excel sheet – railML2.4nor Infrastructure Model. Additionally, also view the example at the end of this document and the railML2.4nor XSD.

The colours in which the rectangles in the hierarchy are coloured have the following meaning:

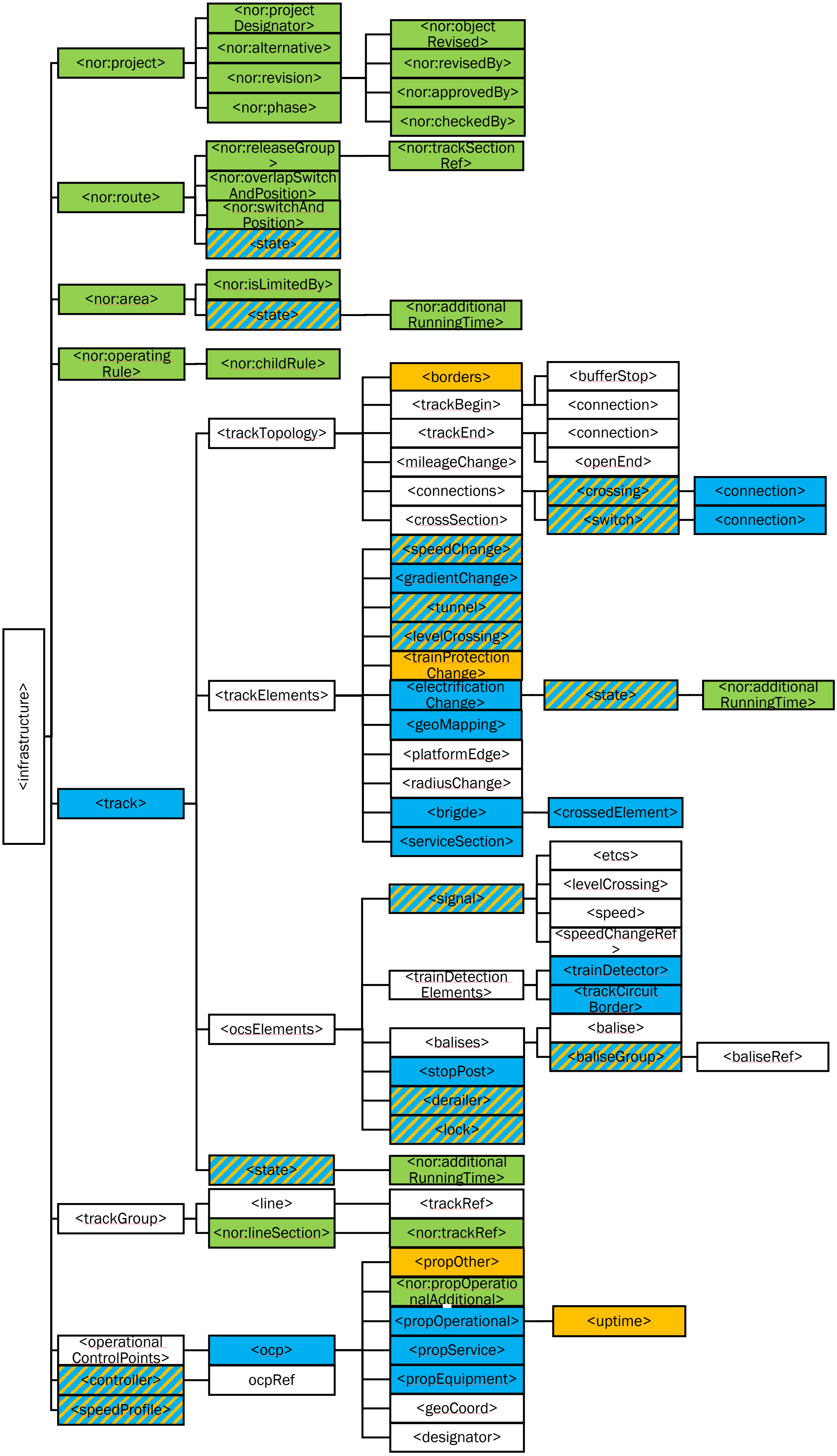
- Green: These are newly introduced extension elements. They can also be identified by their "nor:" namespace and are not part of the railML2.4 core schema.
- Orange: These elements are railML2.4 core elements but are extended by Norway specific attributes or values.
- Blue: These elements are railML2.4 core elements, which are defined ambiguously by railML and thus were defined more precisely in accordance to Norwegian usage.
- Non-coloured: These elements are railML2.4 core elements, which are defined precisely by railML and thus do not need to be defined more precisely for railML2.4nor.

All elements that are coloured in the hierarchy are addressed in this document. A definition of all non-coloured elements is available in the railML 2 Wiki. All elements are listed with their – for railML2.4nor - relevant attributes and value sets in the Excel sheet.

Furthermore, please note that railML2.4nor contains several extensions which are applicable to all elements. This is not illustrated in the hierarchy as it would require every element to be coloured in blue and thus interfere with the actual purpose of illustrating other element specific changes. These extensions are:

- Attribute extension of core raillML2.4 <state> sub-element, see chapter 4.14
- Attribute @nor:change, see chapter 3.6 and 3.15

railML2.4nor Infrastructure content on element level



3 General Modelling Rules

The following chapters explain general modelling rules that must be taken into account when working with railML2.4nor data.

3.1 Main language

The main language used in railML2.4nor IS is Norwegian. That is to be specified in the sub element <dc:language> of the element <metadata> and applies for all @name and @description attributes of children elements of <timetable>. The value of the unique language identifier <dc:language> complies with the language standard IETF BCP 47, which is described in the railML-wiki[2]. In practice this element will be described in railML in the following way:

```
<dc:language>
   no-NO
</dc:language>
```

If the language of any @name or @description attributes of any element differs from the main language, the used language has to be described under @xml:lang of the concerned element.

3.2 Missing information and default values

No railML2.4nor extension attributes have default values. However, core railML2.4 defines default values, for some attributes that are used in railML2.4nor. The following shows the complete list of railML2.4 core attributes that have default values and exist in railML2.4nor:

<element>@attribute	Default Value
<crossing><connection>@passable	“true”
<electrificationChange>@type	“none”
<propOther>@status	“operational”
<propService>@passenger	“false”
<propService>@service	“false”
<propService>@ship	“false”
<propService>@bus	“false”
<propService>@airport	“false”
<propService>@tariffpoint	“false”
<propService>@goodsLoading	“false”
<propService>@goodsSiding	“false”
<propService>@goodsIntermodal	“false”
<propService>@goodsMarshalling	“false”
<signal>@virtual	“false”
<switch><connection>@passable	“true”

Core railML2.4 further defines default values for attributes that are not used in railML2.4nor. However, other attributes of these elements are used in railML2.4nor, which is why systems could falsely interpret default values of not used attributes. Therefore, the following lists all railML2.4 core attributes, which are to be ignored by railML2.4nor reading systems:

[2] https://wiki2.railml.org/index.php?title=TT:dc:language

<element>@attribute	Default Value
<trainProtectionChange>@monitoring	“none”
<electrificationChange>@voltage	“0”
<electrificationChange>@frequency	“0”
<electrificationChange>@isolatedSection	“false”
<ocp><propOperational>@orderChangeable	“false”
<ocp><propOperational>@ensuresTrainSequence	“true”
<ocp><propOther>@frontier	“false”
<ocp><propOther>@chargeFrontier	“false”

Default values for all further railML2.4nor attributes can be defined by user systems. In case a value is not known for a mandatory attribute, the writing system will have to insert a value.

If no information is available for an attribute, optional attributes shall not be written. The procedure for mandatory attributes is explained in chapter 3.5.

If attributes and their values are not written, this denotes that a value, or information to derive it, is unknown in the system producing the railML2.4nor file. This is regardless of whether it is not required there, or the value is missing and can be ignored by the receiving system.

3.3 Conventions

3.3.1 New elements and attributes

In this document, the Norwegian extensions will be marked with a “nor:” prefix:

[nor:]+[attribute/element name]

Example: Extension element <nor:lineSection>

3.3.2 Norwegian extensions of enumerations using other:*

Some railML 2.4 attributes do not contain specific values needed for Norwegian use cases, but support extending the list by using the other:* pattern. The Norwegian sector decided to generally add other:* enumeration values in the following way when the value is specific to Norway:

"other:"+[nor]+[Value]

Example: <balise>@type="other:norFF" for a Norwegian FF balise

Please note: If the value is not specific to Norway, or used for a Norwegian extension element or attribute where the element or attribute is already prefixed with “nor”, it is omitted in the enumeration value.

Examples: <nor:area>@type="trackSection" and <crossedElement>@type="other:primaryRoad"

A special case is that some enumerations have been extended with the value “other:norUnknown”, where the value “unknown” is not already available, but needed for Norwegian use cases.

All the agreed enumeration extensions are found in the Excel sheet “railML2.4nor Infrastructure Model”.

3.3.3 Exporting unmappable values in enumerations

The “other:*” pattern can also be used to export information that cannot be mapped to an available enumeration value in railML2.4nor. The unmappable value is then prefixed by “other:”. When not present in the Excel sheet “railML2.4nor Infrastructure Model” this denotes a value (*) that is not part of or mappable to a value in the fixed railML2.4nor value set. This value can be ignored, mapped or handled by a local "other"-value in the receiving system.

Example: The source system Banedata maps the status value "I DRIFT" to the corresponding railML value "operational". In case of the incorrect registered value "IBRUK" the @status value is mapped to "other:IBRUK". Receiving systems may ignore this value, or choose how it should be handled.

3.4 Usage of deprecated attributes

railML2.4nor uses attributes which have been marked as deprecated in railML2.4. However, they are of great importance and value for railway operations in Norway and are thus used in railML2.4nor. The Norwegian sector requested to revoke the deprecated mark in future railML versions. The following deprecated elements and attributes are used in railML2.4nor:

- <levelCrossing>@protection
- <signal>@switchable
- <signal>@sigSystem

3.5 Mandatory elements

Generally, railML2.4 and railML2.4nor define some elements and attributes as mandatory and others as optional. As railML2.4nor is an extension of railML2.4, elements which are defined as mandatory in railML2.4 can never be optional in railML2.4nor. However, optional core railML2.4 elements can be made mandatory in railML2.4nor.

Further restrictions that cannot be recorded in the XSD exist in railML2.4nor. These are noted in the Excel sheet “railML2.4nor Infrastructure Model”. In order to comply with the standard, these restrictions have to be taken into account.

3.6 Container elements

railML2.4nor contains various container elements. Some are part of core railML2.4, e.g. <signals>. Others were introduced as extension elements in railML2.4nor, e.g. <nor:routes>. Please note that there must never be empty container elements in a railML2.4nor file. In other words: All container elements must have individual elements. This is reflected in the Excel sheet, so for instance the <signals> container is optional, but if it exists, it must have at least one <signal> element. Therefore, the <signal> element is marked as mandatory. If a track has no signals, the <signals> container is skipped.

```
<nor:routes>
   <nor:route […]>
        <nor:switchAndPosition […]/>
   </nor:route>
</nor:routes>
```

3.7 Common attributes

In core railML2.4 and in railML2.4nor most elements have a set of attributes, which apply to a common set of semantics and value definitions. Nevertheless, some elements are exceptions to this rule. A complete overview over the elements that have or do not have those common attributes is given in the Excel sheet.

The following table lists the common set of attributes:

Attribute	Definition extension to Norwegian requirements, if applicable
@id	A unique ID string within the file's scope.
@name	(Human-readable) short string instance name. Avoid repeating information provided in the computer-readable attributes.
@description	Human-readable description giving added information to the name. *Remarks/comments related to the workflow must also be placed in the description until a dedicated element is in place.*
@nor:change	Specifies the type of change for a planned infrastructure element that is pending approval. For further information see chapter 3.15.1

Attribute	Definition extension to Norwegian requirements, if applicable
@nor:ruleRef	Reference to the operational rule that applies to the element. For further information see chapter 4.26. (optional)
@code	This is a UID for elements/objects. For <state>@status = "planned", "operational", "disabled" or "closed" this is a global Bane NOR ID for elements/objects. It is defined in the Bane NOR infrastructure asset management register (BaneData), the system of record for infrastructure information for Norwegian railways. New objects are to be registered in the asset management register before being used in the railML file. Only Bane NOR is eligible to write or edit values for this attribute. For <state>@status ="conceptual" and "other:norUnknown" the values should be generated from a stable repository. This attribute is: • mandatory for <state>@status = "planned", "operational", "disabled" or "closed" • optional for <state>@status ="conceptual" and "other:norUnknown"
@pos	See chapter 3.8
@absPos	See chapter 3.8

Please note: A a new <remark> element has been proposed to be available in railML2.5.

3.8 Position of elements

In railML2.4nor, the two positioning attributes @pos and @absPos are not strictly connected to one another. In accordance to Norwegian usage they are used to model the following aspects in railML2.4nor:

Attribute	Description in accordance to Norwegian usage
@pos	Signifies the precise position on the track from the beginning of the <track> element, given by the sub element <trackBegin>, to the relevant element. A <trackBegin> element always has the value @pos="0". The unit is meters. The precise position means that the measured value should not deviate more than 20 cm from the real position. *The @pos value will either be a distance measured from the position of the <trackBegin> element or if not available be derived by the difference of the @absPos attributes of an object and the @absPos of the track's <trackBegin> element.*
@absPos	This is a reference in accordance to the official national mileage referencing system in Norway (kilometrering). It is a more unprecise measurement of an item’s location than @pos. The unit is meters. This attribute is: • mandatory for <state>@status = "planned", "operational", "disabled" or "closed" • optional for <state>@status ="conceptual" or "other:norUnknown" *The @absPos attribute shall provide a continuous frame of reference for infrastructure information of other data sources and the real world. The mileage number shall directly or indirectly be inherited from the infrastructure data inventory (BaneData) as a master data system of record. As a consequence, an @absPos value has to be derived when modelling new infrastructure objects. This is done by adding the length between a new infrastructure object and the @absPos value of the closest existing object.*

Example:
Two boards, one on each track, can have the same @absPos values but two different @pos values.

Please note that in signal drawings the two signals would be aligned, as they have the same absPos value.

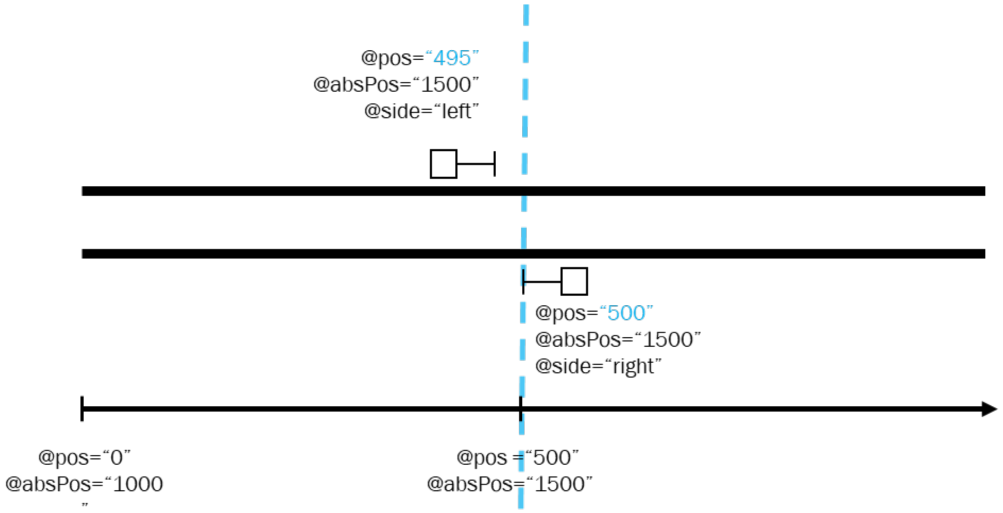


Figure 3: Illustration of @pos / @absPos example

3.9 Orientation of elements

With the exception of <speedChange>@vMax, the value of an attribute in railML2.4nor is always interpreted in track direction. For instance, the <gradient>@slope value is seen in track direction and thus the <gradient> element does not need to use the @dir attribute under normal circumstances.

3.9.1 Extent of elements with a length attribute

Elements that have a @length attribute, e.g. <tunnel> or <platformEdge>, are always placed on the track at the beginning of the object. This is always seen in track direction, i.e. usually the lowest mileage. Thus, they extend from @pos to (@pos + @length) along the track. This is illustrated underneath.

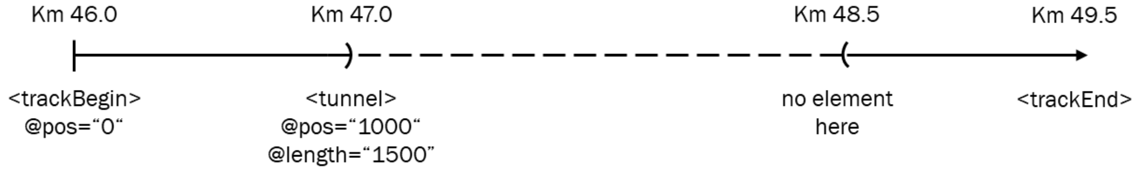


Figure 4: Positioning of elements with @length attribute

If a switch is placed within the length of the object, the element does not branch out to the track connected to the **switch** connection. If applicable for the diverging track, you must then place another element of same type and name on <trackBegin> of the track connected to the switch connection (the diverging track).

```
<trackBegin absPos="46000.0" pos="0.0"/>
<trackEnd absPos="49500.0" pos="3500.0"/>
<tunnels>
   <tunnel absPos="47000.0" length="1500.0" pos="1000.0"/>
</tunnels>
```

3.9.2 Directional validity of elements

In railML2.4nor some elements are only valid in one direction (of travel for the train). These objects use the @dir attribute, which defines the direction in which the element is valid. The values of the attribute can be:

- "up" (the element is only valid in the direction of the track) or
- "down" (the element is only valid against the direction of the track)

If the element can only be valid in both directions the @dir attribute shall be omitted.

Most elements in railML2.4nor are valid in both directions and therefore do not use the @dir attribute. In railML2.4nor, only the following elements have a defined direction of validity. Therefore, it is mandatory to use the @dir attribute:

- <balise>
- <derailer>
- <signal>
- <stopPost>
- <speedChange>

Please note that railML2.4nor does not use the <infraAttributes> element.

3.9.3 Horizontal placement of element (@side)

In railML2.4nor some elements have @side attributes that refer to the position of the element in accordance to the track, e.g. <signal>@side. This element is always interpreted in track direction in railML2.4nor.

Example: A signal or board that is valid in @dir=”down” direction will be declared with @side=”left” if the driver sees it on the right-hand side. If, on the other hand, the signal is on the right-hand side of the driver and valid in @dir=”up” direction it will be declared with @side=”right”. This is illustrated in Figure 3.

3.10 Aggregation level of track model

In railML2.4nor the railway network is modelled as an emulated node-edge style network. This can be done in a microscopic level or in a macroscopic level. Note the levels are not interchangeable (as in railML3).

Edges are generally modelled as one or more <track> elements, in railML2.4nor.

railML2.4 does not provide aggregation of microscopic elements into macroscopic elements contained in the same model. However, the Norwegian sector requires this and thus solves the problem in the following way:

railML2.4nor provides two modelling approaches, which correspond to four modelling levels:

- Microscopic model that supports the microscopic detail level
- Macroscopic model that supports the mesoscopic, macroscopic and network aggregation detail levels
- It is always possible to convert a model to an aggregated model in a higher modelling level, but it is not possible to disaggregate a model into a more detailed one without making assumptions or adding information. It is possible, but not recommended to include both a primary and a secondary model in the same file. The defined modelling levels are:

Level of modelling	Detail of modelling	Description	Usual timeframe
Microscopic	Microscopic	This refers to the standard model for railML2.4nor and is described in detail in this document.	Historic, Existing, planned (+2years)

Level of modelling	Detail of modelling	Description	Usual timeframe
Macroscopic	Mesoscopic	In addition to macroscopic line tracks as described in “macroscopic detail and modelling” row below and in section 3.10.2, all mesoscopic tracks from an operations/timetable planning perspective within the station are modelled. The lengths of station tracks are defined by the @pos in <trackEnd> and must correspond to the tracks’ operational lengths. If the length is unknown, use a length of 0m. The order of the tracks can be indicated in propEquipment/trackRef@sequence. See figure 7 for illustration.	Planned (2-6 years)
	Macroscopic	Only macroscopic line tracks (main track from ocp crossSection to neighbour ocp crossSection) that connect ocps are modelled. The line tracks have <trackElements> with mandatory use of <speedChange> and <gradientChange> and can additionally have <ocsElements> placed on them. Station tracks are not used. The primary purpose for the macro model is to exclusively calculate the runtime of main tracks for timetable planning.	Study (6-15 years)
	Network	Only line tracks that connect ocps with no <trackElements> or <ocsElements> placed on the track. The primary purpose of the network model aggregation is simply to incorporate information about the order of the stations on the line and the connection between lines in the network.	Study (+15 years)

3.10.1 Microscopic model

As railML2.x does not support proper node-edge modelling, the elements representing the nodes are positioned at the beginning (with @pos="0") or at the end (with @pos equal to the @length of the <track>) of a <track>. The node elements <switch> and <crossing> are always positioned at the beginning of the track. The node element <bufferStop> can occur in either <track><trackBegin> or <track><trackEnd>, depending on which end that is not connected to the network.

In core railML the track direction is defined from <trackBegin> to <trackEnd>. railML2.4nor further defines that <trackBegin> and <trackEnd> must be placed so that the track direction is the same as the direction of increasing mileage. This rule cannot be expressed in the XML schema language. In Norwegian schematic drawings the increasing mileage direction is usually drawn from left to right.

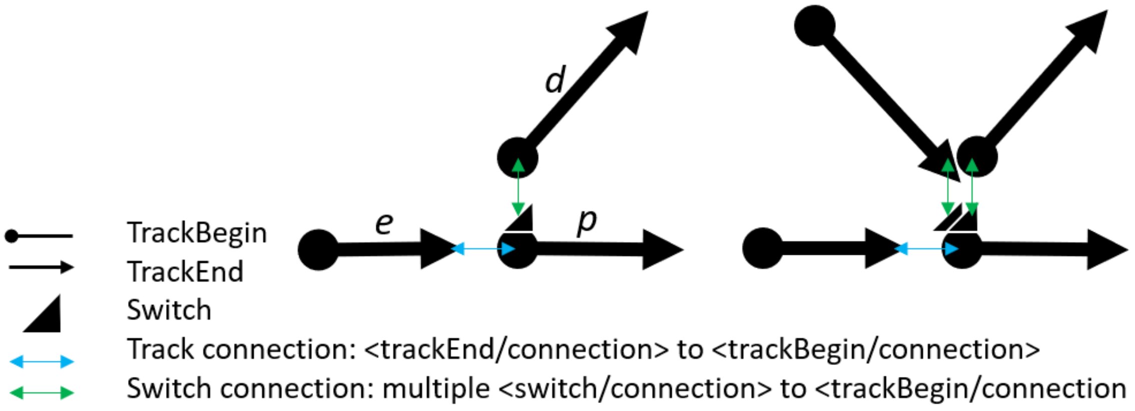

d
e
p
TrackBegin
TrackEnd
Switch
Track connection: <trackEnd/connection> to <trackBegin/connection>
Switch connection: multiple <switch/connection> to <trackBegin/connection

Figure 5a: Illustration of split track model

The Norwegian sector decided to exclusively use a split track model for railML2.4nor. I.e. the track is split into new segments at every switch. Tracks are connected by a <connection> element between the <trackBegin> of one track element and the <trackEnd> of another. As shown in Figure 5a the switch is always placed on the <trackBegin>.

Note in case an element is positioned directly in the node (mileage of the switch), it can be placed in three different track positions. For a branching switch as illustrated in Figure 5a this can either be the end of track "e", the beginning of track "p" or the beginning of track "d". The placement will influence the application of the element for a train traversing the tracks. Placing an element in the stated example on the end of track "e" makes the element applicable for movements over both legs of the switch. Placing the element on one of the legs ("p" or "d") makes it applicable only for movement over that leg.

In the case of two switches meeting stock rail to stock rail, both switches are placed on the same <trackBegin>, as illustrated in Figure 5a. Alternatively, for legacy reasons, railML2.4nor also allows this scenario to be modelled with a track of 0 meters (or next to 0) between the switches, as illustrated in Figure 5b.

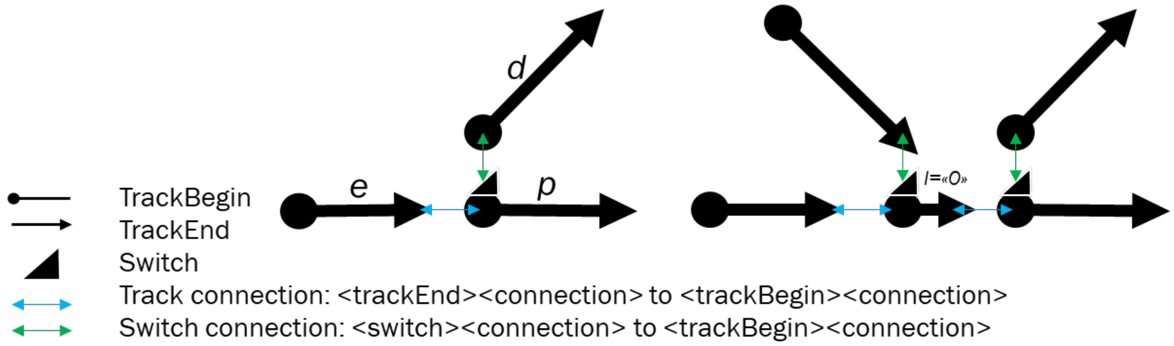

d
e
p
l=«0»
TrackBegin
TrackEnd
Switch
Track connection: <trackEnd><connection> to <trackBegin><connection>
Switch connection: <switch><connection> to <trackBegin><connection>

Figure 6b: Illustration of alternative split track model for 2 switches at same mileage.

Nodes are represented by a <switch>, <crossing>, <openEnd>, <bufferStop> or (home) <signal> element. A track name change represents an abstract node. All OCPs are, according to core railML2 superimposed, using <crossSection>s (reference from track to ocp) and <propEquipment> (reference from ocp to track).

In the case of a change of direction of the mileage, two <trackBegin> or two <trackEnd> elements are connected. To avoid duplicate mileages on a line, mileage direction changes should be placed on station borders. As a switch or crossing cannot be placed on a line border, it follows that a switch or crossing cannot be placed on the same position as a mileage direction change.

●— TrackBegin
→ TrackEnd
↔ Track connection: <trackEnd><connection> to <trackBegin><connection>

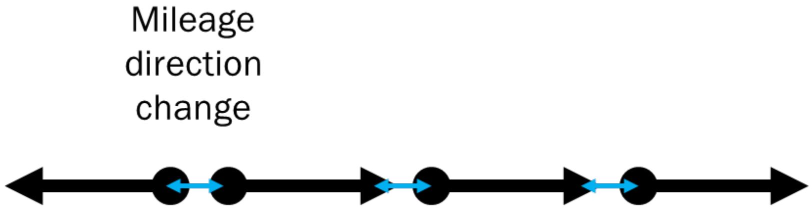


Figure 7: Illustration of mileage direction change

3.10.2 Macroscopic modelling

In the macroscopic model the tracks connect ocp to ocp via the element <macroscopicNode>. Station tracks are superimposed (including the main track). Macro line tracks (main track from ocp crossSection to neighbour ocp crossSection) connect between neighboring ocps. Station tracks connect on both <trackBegin> and <trackEnd> to the same ocp.

> As the precise modelling for macroscopic models has not been decided, due to missing experience and defined use case, we ask to reach out to Jernbanedirektoratet railML stakeholder before implementing a macroscopic railML2.4nor model.

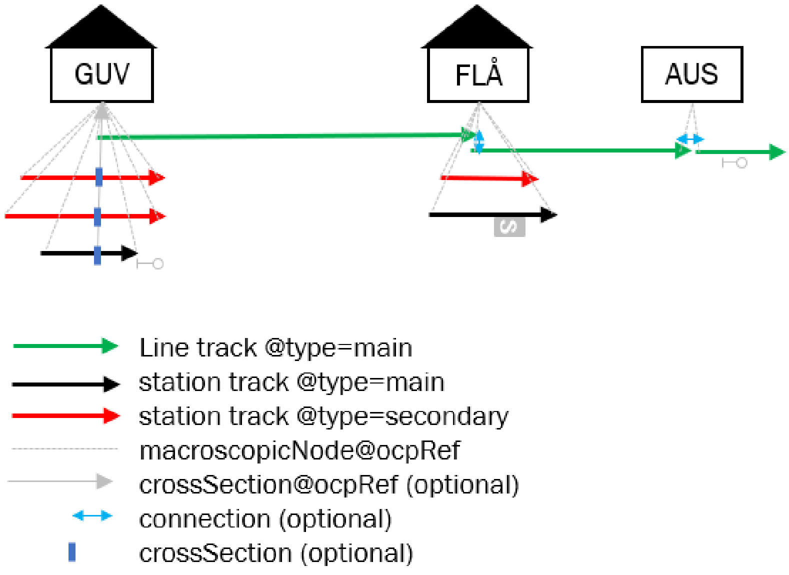


Figure 8: Illustration of the mesoscopic infrastrucure

Se railML2.4nor timetable documentation general example for an infrastructure example on a macroscopic level.

3.11 Mandatory placement of "<*Change>" elements

railML2.4nor currently uses the following different <*Change> elements:

- <speedChange>
- <gradientChange>
- <electrificationChange>
- <trainProtectionChange>

The value of these "*Change" elements must always be defined for the entire model to ensure that values for all different elements exist at every location. Thus, at least one of each type must exist in every railML2.4nor model.

Where tracks which have no connection (with either a <openEnd> or <bufferStop>) begin, all four *change elements must be placed. Where tracks which have no connection (with either a <openEnd> or <bufferStop>) end, a <speedChange> element must be placed. See also table below and figure 8 as illustration.

The following table explains where to place the different elements:

Element	Mandatory placement at unconnected track begin	Mandatory placement at unconnected track end	Mandatory placement at mileage direction change	Mandatory placement when value changes
<speedChange>	Yes, placement with dir="up"	Yes, placement with dir="down"	Once for each direction	yes
<gradientChange>	yes	No	Once for each <trackBegin>	yes
<electrificationChange>	yes	No		yes
<trainProtectionChange>	yes	No		yes

To indicate that a <speedChange> element is placed only for informational purposes about the current speed limit at a model's entry point, use the attribute <speedChange>@signalised="false". As there is no @signalised attribute for <gradientChange>, <electrificationChange> and <trainProtectionChange> this does not apply for these elements.

3.11.1 Flow of "<*Change>" elements along the track(s)

*Change elements are valid from the @pos value of the track they are placed on, in the direction they are applicable, until the next *Change element of the same type. *Change values flow over connections in the following way:

Track connection

If two tracks are connected via a track connection (<trackTopology><trackBegin><connection> or <trackTopology><trackEnd><connection>) the *Change values flows over the track connections.

Switch connection

*Change values also branch out over switches i.e. they transfer the value to all connected tracks, in the drivable direction. As tracks are always split at a switch or crossing in the railML2.4nor model (see chapter 3.9), switches will have both one track connection and one switch connection (<trackTopology><connections><switch><connection>) and crossings will have one track connection and two crossing connections (<trackTopology><connections><crossing><connection>). As mentioned, the *Change values always flow over the track connection in its applicable direction. So, from the entering track to the principal track ((see section 4.3.1) and vice-versa. *Change values flow over the drivable direction over the switch connections. So, from the entering track to the deflecting track. This means that a *Change element value with a @dir attribute with an "up" value or with no @dir attribute transfers to the connected track in a switch with @orientation="outgoing" (branching switch in track direction). The same occurs for a *Change value with dir="down" over a switch with @orientation="incoming" (merging switch in track direction).

Crossing connections

Note that this branching does not occur on crossings. Here the *Change value from one track axis crosses the *Change value of the other track axis.

See figure 8 for an example.

Exemptions for switch and crossing connections

In case the *Change value shall not be transferred to a branching track, a new *Change element with the applicable value needs to be placed on the beginning of the track that connects to the switch connection – i.e. the diverging track.

In case a speed restriction only applies while passing through the diverging track of a switch or crossing, the <connection>@maxSpeed attribute must be used (see example in figure 8). The speed restriction is only valid for the length of the switch/crossing (see chapter 4.3). If the speed restriction is also valid after the length of the switch/crossing, a <speedChange>@vMax value must be set on the beginning of the diverging track.

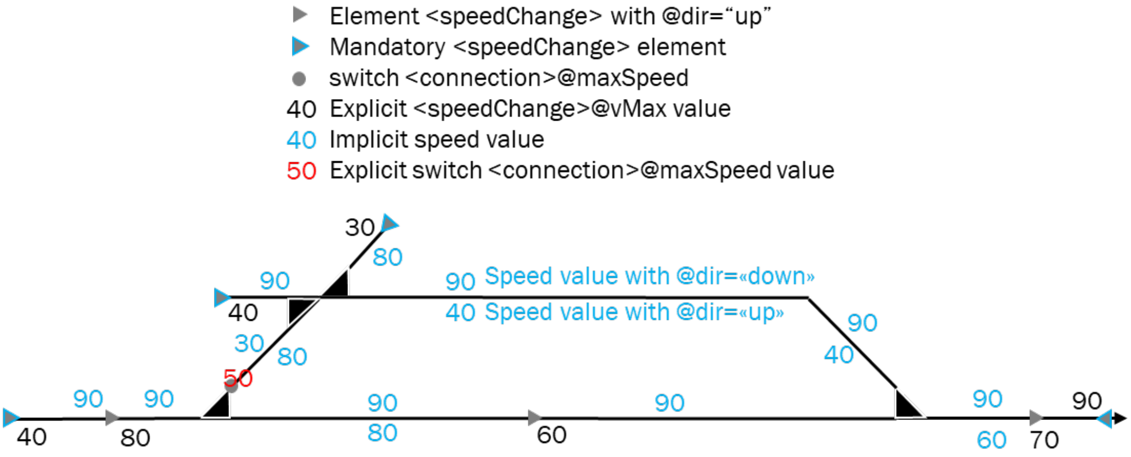

▶ Element <speedChange> with @dir="up"
▶ Mandatory <speedChange> element
● switch <connection>@maxSpeed
40 Explicit <speedChange>@vMax value
40 Implicit speed value
50 Explicit switch <connection>@maxSpeed value
30
80
90
40
30
80
50
90 Speed value with @dir=«down»
40 Speed value with @dir=«up»
90
40
90
90
40
80
90
80
60
90
90
90
60
70

Figure 9: Flow of <speedChange> elements along the tracks

3.12 Inheritance

In railML2.4nor attribute values of the <state> element and the @nor:change attribute values are handed down to elements in a hierarchical order. All elements under the mother element inherit the attribute value. The inheritance is broken whenever a new <state> element or @nor:change attribute is placed on a child element.

I.e., if a value is placed on a mother element (e.g. <infrastructure>), the same value shall not be placed on a child element (e.g. <track>) again. The child element inherits the value. If different values are placed on mother and child, the child element's value takes precedence. The hierarchy follows the XML structure:

1. <infrastructure>
2. <track>
3. Individual elements/objects (e.g. <signal>)

Example: If a signal will be removed, @nor:change="removed" is placed on the signal element. If a whole track is removed, @nor:change="removed" is placed on the track element, and also applies to all elements defined as children of the track element. So, the @nor:change="removed" placed on the signal shall be omitted.

3.13 Project model

This chapter only describes the modelling concept of the project model in railML2.4nor. For the description of the element <nor:project> with its attributes and definitions see chapter 4.13.

In railML2.4nor a <nor:project> describes the project to which the infrastructure in the file belongs. The element describes the metadata of the project - e.g. the name, code, alternative, revision, etc. Please note that the information provided in <nor:project> is not the same as the information in the <metadata> element. The <metadata> describes the data source.

The infrastructure of one project can be divided into separate models, in separate railML files. The data which is stored in <nor:project> will be duplicated for each model of the same project. The connection of different models is further described in chapter 3.14.

3.14 Connection of models

The project model described in chapter 3.13 allows projects to be in more than one model and requires different projects to be in different models. Each model is represented by one <infrastructure>. As there is only one <infrastructure> in one railML file in railML2.x, each model will be in a separate railML file. These models then need to be able to be connected to each other in a precise and stable manner.

3.14.1 Connection between models in the same <nor:project>

For connections between adjoining models in the same project (e.g. lines in separate files) the models are linked by matching <openEnd> elements on both models. As <openEnd> elements can only be placed on <trackBegin> or <trackEnd> elements, models shall only be broken on these. According to chapter 3.10 they are only to be placed on the nodes of the models. The @code value of the <openEnd> of model A shall have the same value as the @code value of the corresponding <openEnd> element of model B.

Example:

```
<openEnd id="oe35"  code=" BB-1-8000"/>
```

3.14.2 Connection between models in different <nor:project>

Connections between replacing/overlapping models in different projects, e.g. conceptual development of an operational infrastructure, are defined by a border of type "area". A project border is placed on the edge of a project, where the initial (unchanged) model and the new (changed) model meet. All project borders must have @absPos (mileage) values and belong to a track with a line and a track name. Subsequently these three values are used to connect models in different projects:

- <border>@absPos
- <line>@name
- <track>@name

Please note that generally no physical objects are used, e.g. signals or balises, to connect models, as these may change their position over time.

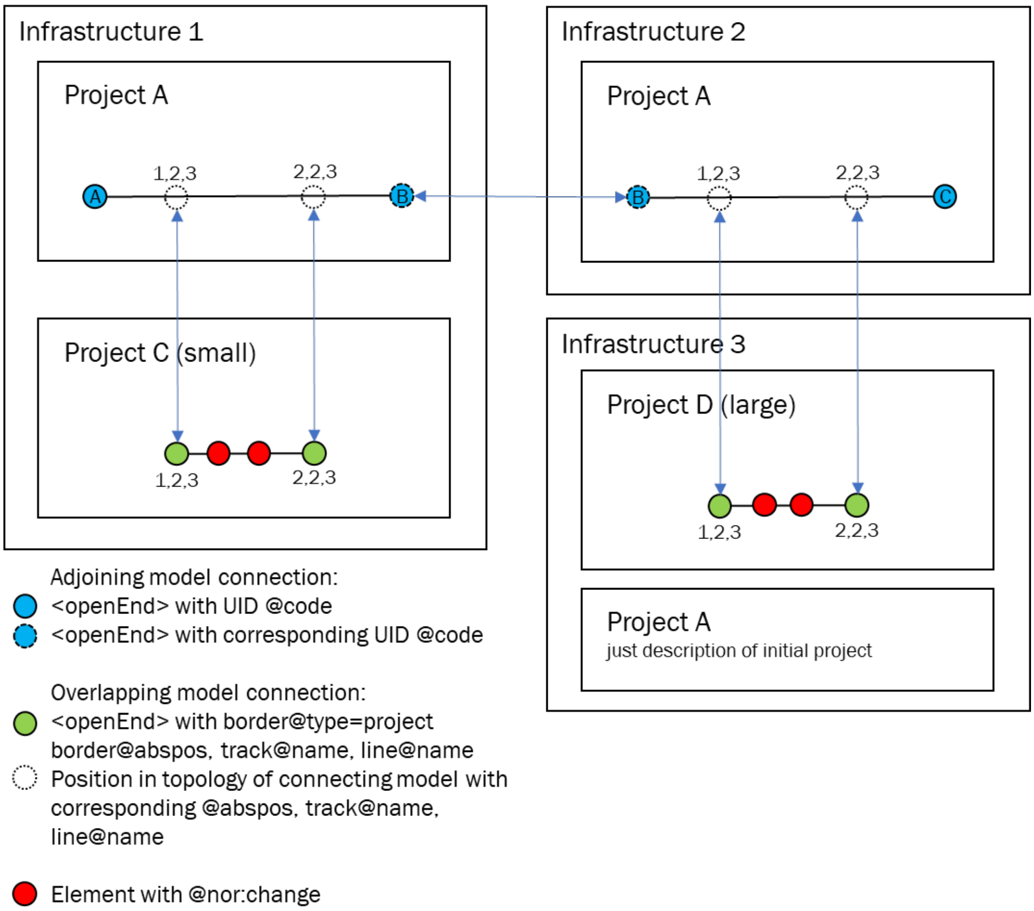

Infrastructure 1
Project A
A
1,2,3
2,2,3
B
Project C (small)
1,2,3
2,2,3
Infrastructure 2
Project A
B
1,2,3
2,2,3
C
Infrastructure 3
Project D (large)
1,2,3
2,2,3
Project A
just description of initial project
Adjoining model connection:
<openEnd> with UID @code
<openEnd> with corresponding UID @code
Overlapping model connection:
<openEnd> with border@type=project border@abspos, track@name, line@name
Position in topology of connecting model with corresponding @abspos, track@name, line@name
Element with @nor:change

Figure 10: connection between nor:project and infrastructure model

3.15 Time dimension

In core railML2.4 the infrastructure objects time dimension is defined as snapshots. Those snapshots have one or more state(s) with a status and a timeframe. The timeframe is defined by the operating period. However, in railML2.4, timeframes that are defined by the <operatingPeriod> element always have to be closed. In Norway, infrastructure objects can also have open timeframes. Therefore, new attributes - @nor:startDate and @nor:endDate – have been added to the <state> element (see chapter 4.14).

In railML2.4nor the time dimension can be defined explicit with dates in the <state> element or it can be dateless. Dateless time dimension uses either consecutive phases in the project model (phase 40 is after phase 30) or the <state>@status values ("planned" comes after "operational" in time).

3.15.1 The @change attribute

The initial model - baseline project - and the final model - the deadline project - are both described or referenced (see chapter 3.14). All changes between the initial and the final model are stated with the attribute @nor:change. It has the following values:

Value of @nor:change	Description
“removed”	Element proposed to be removed from the model. In this case, it is mandatory to additionally use the core railML <state>@disabled=”true”, so that receiving systems cannot interpret this element as usable. *In track plans these infrastructure elements are rendered red.*
“modified”	Element proposed to be modified from its original form. *In track plans these infrastructure elements are rendered blue.*

Value of @nor:change	Description
“new”	Element proposed to be added. *In track plans these infrastructure elements are rendered green.*

Note that this attribute is not extendable via the “other:*” pattern.

4 Element Specific Definitions and Extensions

4.1 The element <track>

4.1.1 Definition extension in accordance to Norwegian usage

railML2.4nor uses the <track> element as defined in railML2.4. Three aspects are defined more precisely in accordance to Norwegian usage:

Track types

Track types in Norway differ slightly inside stations and outside of stations. A simplified definition of stations is an area between two main home signals. Please see TJN chapter 1.11 for a complete definition.

In accordance to the Norwegian legislation TJN chapter 1.11 the existing railML2.4 values of the attribute @type are interpreted in railML2.4nor as follows:

- Track types at a station; modelled as operational control point (<ocp><propOperational>@operationalType="station”):

Value of <track>@type	Definition in accordance to Norwegian usage
“mainTrack”	Track to drive into the station with all switches in their normal position.
“secondaryTrack”	Other tracks that have routes going over them.
“sidingTrack”	Residual tracks without main signal routes, like shunting routes or no routes, like shunting areas.
“connectingTrack”	Tracks with no other function than to connect two tracks. *This track type is not defined in the Norwegian operating regulations (TJN) but is in use in Bane NORs Infrastructure asset management system. It is also part of core railML2.4 and is deemed useful for the use cases. Thus, the Norwegian sector decided to include the track type "connectingTrack".*

- Track types outside a station; modelled as open section (nor:lineSection@type=”openSection”):

Value of <track>@type	Definition in accordance to Norwegian usage
“mainTrack”	Predominant track on the open section. A single track line has exactly one main track. A double track line has two main tracks.
“sidingTrack”	Track of an <ocp>@operationalType="siding" at an open section that is not a station. Sidings in Norway are usually factory tracks on the open section for loading and unloading industrial products and timber. The switch leading to the siding track needs to be locked.
“connectingTrack”	Track connecting the two main tracks on a double tracked line, which does not have a route that goes over it. Thus, the switches are locked.

The (driving) direction on a track

In railML2.4nor all tracks of type "mainTrack" have a mandatory attribute @mainDir. For secondary tracks the use of @mainDir is optional. This defines the predominant operational travelling direction over the tracks of a double track line:

- Single track line: <track>@mainDir="none"
- Double track line:
  - <track>@mainDir="up" for trains predominantly travelling in the up direction
  - <track>@mainDir="down" for trains travelling in the down direction.

The name of a track
In Norway, tracks are named by their track number according to the Norwegian technical design rules (teknisk regelverk – TRV530-14-3).

A track extends towards either a home signal or a buffer stop following the track type through switches. If two identical track types merge in a switch the track number follows the principal track after the switch (see definition of switches).

Special case
A special scenario occurs when a single-track line converts into a double track line at an OCP of <propOperational>@operationalType="junction" or "station". This scenario is illustrated in the OCP of type "junction" in the middle top part of Figure 10.

A scenario where no main track continues unbroken throughout the station, is illustrated in the OCP station C in the right down part of Figure 10. Here the main track is broken and extends in parallel. They end at the exit signal on the other side of the OCP cross section. The track continues as type “secondaryTrack” until they merge again with the other main track.

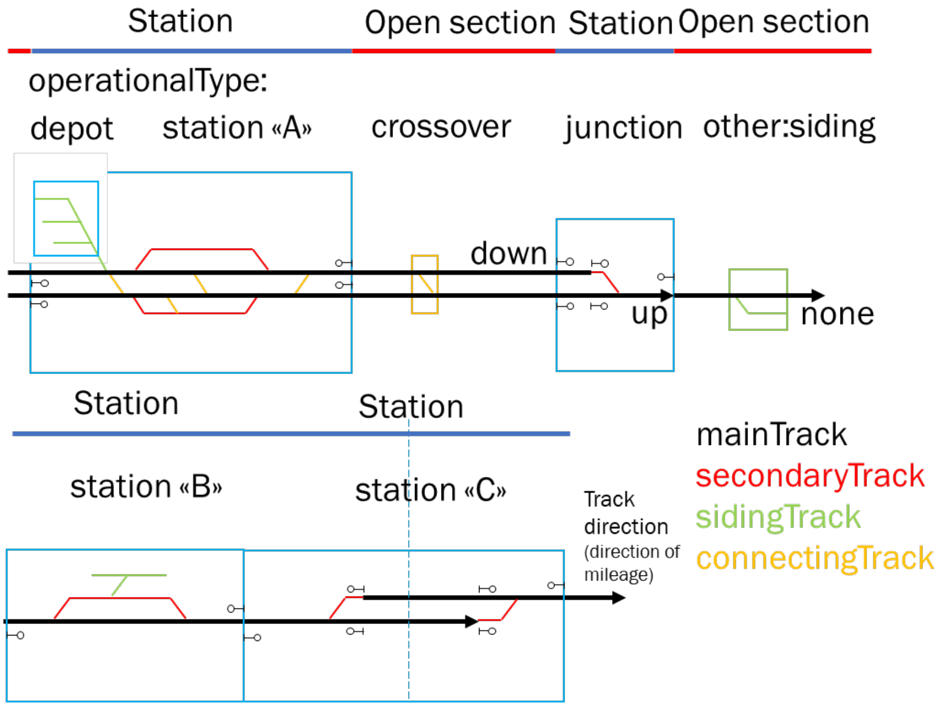


Figure 11: Track types

4.1.2 Code example

```
<track code="KO-SPO-800685" description="naming rule: [arabic numbers]"
id="tr32" mainDir="none" name="3" type="sidingTrack"
...
<track/>
```

4.2 The element <ocp> - operational control point

4.2.1 Definition extension in accordance to Norwegian usage

railML2.4nor uses the railML2.4 concept of describing operational control points (OCPs). Nevertheless, some specific definitions in accordance to Norwegian usage are added for different sub-elements/attributes of the element <ocp>.

Types of OCPs

An OCP's primary purpose is defined by the @operationalType attribute of the <ocp> sub-element <propOperational>. Norwegian railways differentiate between operational control points of type station and of type open section in their regulation TJN. Note only OCPs of operational type "station", "stoppingPoint", "blockSignal" and "siding" are declared in TJN regulations. Definitions of "junction", "depot" and "crossover" are according to this document for the Norwegian railway sector. Note that TJN regulations differentiates on "stoppingPoint" inside a station and on the open section (between stations). Which values the Norwegian OCPs have in railML2.4nor is listed underneath:

- OCPs which are declared as stations:

Value of <ocp><propOperational> @operationalType	Definition in accordance with Norwegian usage
"station"	OCP with a home signal. It is not distinguished between the facilities on a station. This is defined under <propService>.
"junction" *Currently this type is not mapped as a separate OCP operational type in Norway, so use operational type "station".*	A station with no service that branches into another main track (single to double track) or another line, e.g. Sandbukta stasjon.
"depot" *Currently this type is not mapped as a separate OCP operational type in Norway, so use operational type "station".*	A station, which has depot (maintenance/parking), but no (passenger/goods) services.
"crossover" *Currently this type is not mapped as a separate OCP operational type in Norway, so use operational type "station".*	A station, which has connecting tracks, but no service, e.g. Hellerud stasjon.

- OCPs which are declared as **within** stations:
  (It is mandatory to reference the parent station with @parentOcpRef)

Value of <ocp><propOperational> @operationalType	Definition in accordance with Norwegian usage
"junction"	Reference points (usually switches) within stations used to define mesoscopic (branching) paths through a station. E.g. "Hafslund nord/syd/øst" inside parent OCP Sarpsborg stasjon or "SPO» and «249» inside parent OCP Oslo S station.
"depot"	Designated area within a station for service of type "service" e.g. maintaining trains. E.g. Ski driftsbanegård inside parent OCP Ski stasjon.
"stoppingPoint"	Secondary stop contained within a station. E.g. Skansen Stoppested with Trondheim S as parent ocp.

- OCPs which are declared as open sections in TJN:

Value of <ocp><propOperational> @operationalType	Definition in accordance to Norwegian usage
"stoppingPoint"	Halt on the open section for disembarking and/or embarking of passengers
"blockSignal"	Segmentation of the open section into block sections for multiple trains to follow in the same direction.
"siding"	This refers to one or more tracks on the open section that are not the main track. The purpose usually is accessing the network for the loading and unloading of goods (most common at a factory or a timber loading area) without a full interlocking. Thus, the siding does not have signals, e.g. Holmen sidespor.

The <propEquipment><trackRef> sub-element
In railML2.4nor it is mandatory to reference all tracks belonging to an OCP. For OCPs that have a defined extent according to TJN - being the TJN stations listed above – this is easy. As tracks in the primary (micro) model (see chapter 3.10.2) are to be segmented in TJN station borders (home signal node) this reference will be exact. However, according to TJN definition, OCPs in the open section are point objects. Therefore, in railML2.4nor, open section type OCPs reference all tracks that are part of the open section, which the OCP is located in. Tools that demand OCPs to have defined areas, need to deduct this themselves from the open section, which the OCP is part of.

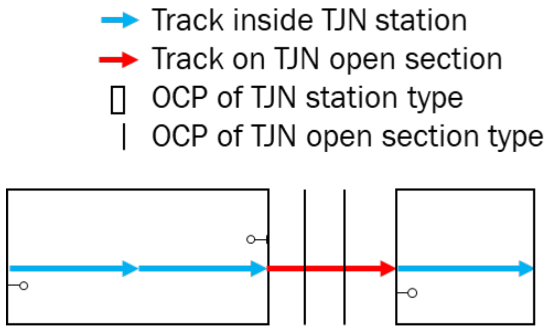


Figure 12: Illustration of TJN open section OCP

In a mesoscopic model also use the attribute <trackRef>@sequence with the following definition extension.

Attribute of <trackRef>	Definition in accordance to Norwegian usage
@sequence	For parallel tracks the order is from right to left as seen in track direction of the main track from lowest to highest @sequence value.

The <propOther> sub-element
In core railML2.4 this sub-element is used to indicate the status of an OCP with a further sub-element <state>@status. The value list of the @status attribute is extended by “other:norUnknown”.

The <propService> sub-element
As the <propService> element is not clearly defined in railML2.4 the Norwegian sector includes an extended definition for all ambiguous attributes according to Norwegian usage:

Attribute of <propService>	Definition in accordance to Norwegian usage
@passenger	If this value is true, the OCP offers passenger services. This means the OCP has at least one platform edge. As <platformEdge> is not mandatory under <trackElements> this does not need to be mapped. An OCP of <propOperational><operationalType>="station" with @passenger="true" is a passenger station.
@service	If this value is true, the OCP offers maintenance service. This means the OCP has at least one service section. As <serviceSection> under <trackElements> is not mandatory this does not need to be mapped. An ocp of <propOpational>@operationalType="station" with @service="true" is a depot. *Please note that there is a redefinition towards this semantic ongoing in core railML.*
@bus	If this value is true, the OCP offers connections to other means of public transport services except for airport and ship (e. g. passenger changing to a bus, tram, subway etc.) *Please note that there is a redefinition towards this semantic ongoing in core railML.*
@goodsLoading	If this value is true, the OCP is capable of loading/unloading goods directly on to freightwagons as part of trains. This usually is achieved through a loading platform (aka. a ramp).
@goodsIntermodal	If this value is true, the OCP offers intermodal goods exchange, e. g. transferring trucks or containers on to flat wagons as part of trains. This usually involves a crane or reach stackers.
@goodsMarshalling	If this value is true, the OCP offers marshalling services (e. g. composing freight trains).

Please note: An OCP of <propOperational><operationalType>="station" with <propService>@goodsLoading="true" and/or @goodsIntermodal="true" is a freight terminal.

4.2.2 Sub-element extension

railML2.4nor extends the core railML2.4 functionality of <ocp> by the sub-element <nor:propOperationalAdditional> to specify additional attributes for any OCP. These are:

Attribute of <nor:propOperationalAdditional>	Description	Type
@remoteControlled	Determines whether an OCP is remote-controlled by a Traffic Management System (TMS) or Centralized Traffic Control (CTC)	xs:boolean
@simultaneousEntry	Enumerates several modes based on Norwegian TRV standards that govern the pattern of simultaneous entry into a station. See TRV for definitions.	xs:enumeration

4.2.3 Code example

```
<operationControlPoints>
   <ocp code="8750-85435" description="desc" id="id4" name="Oslo S"
nor:change="modified">
      <nor:propOperationalAdditional remoteControlled="true"
simultaneousEntry="partial"/>
      <propOperational operationalType="station" trafficType="passenger"/>
      <propService bus="true" passenger="true"/>
      <propEquipment>
        <trackRef ref="tr26"/>
        <trackRef ref="tr32"/>
        <trackRef ref="tr43"/>
        <trackRef ref="tr68"/>
        <trackRef ref="tr40"/>
      </propEquipment>
      <propOther status="operational"/>
      <geoCoord coord="52.0 15.0"/>
      <designator entry="OSL" register="SJN"/>
   </ocp>
</operationControlPoints>
```

4.3 The elements <switch> and <crossing>

In the following the two elements <switch> and <crossing> are described. Both, switches and crossings, are physical infrastructure elements, which connect tracks and are modelled in the same way in railML2.4nor. Thus, they are described together in this chapter.

4.3.1 Definition extension in accordance to Norwegian usage

To ensure a common understanding of the existing railML2.4 elements, the following determinations have been made for railML2.4nor:

Placement on the track

As defined in chapter 3.10 Network model, the track is always broken when we place a switch or crossing, and the node elements <switch> and <crossing> are always positioned at the beginning of a <track>.

We define here the track leading into the tongues of the switch as the entering track. All other tracks are the switch's track legs. One of these legs is the principal track and the other legs are the diverging track(s). As switches are to be placed at the begin of a <track> an outgoing switch is to be placed on the <track> representing the switch's principal track. An incoming switch is to be placed on the <track> representing the entering track.

In a crossing there are two axes (tracks continuing through the crossing). It is up to the producing system to choose which axis to place the crossing on, but it is preferable to place it on the track with the higher ranking @type. The entering track and principal track will both be on the chosen main axis. The <crossing> is to be placed on the <track> on the main axis that begins at the crossing. This <track> is also the entering track for the incoming connection. For the outgoing connection, the entering track is the <track> on the main axis that ends at the crossing.

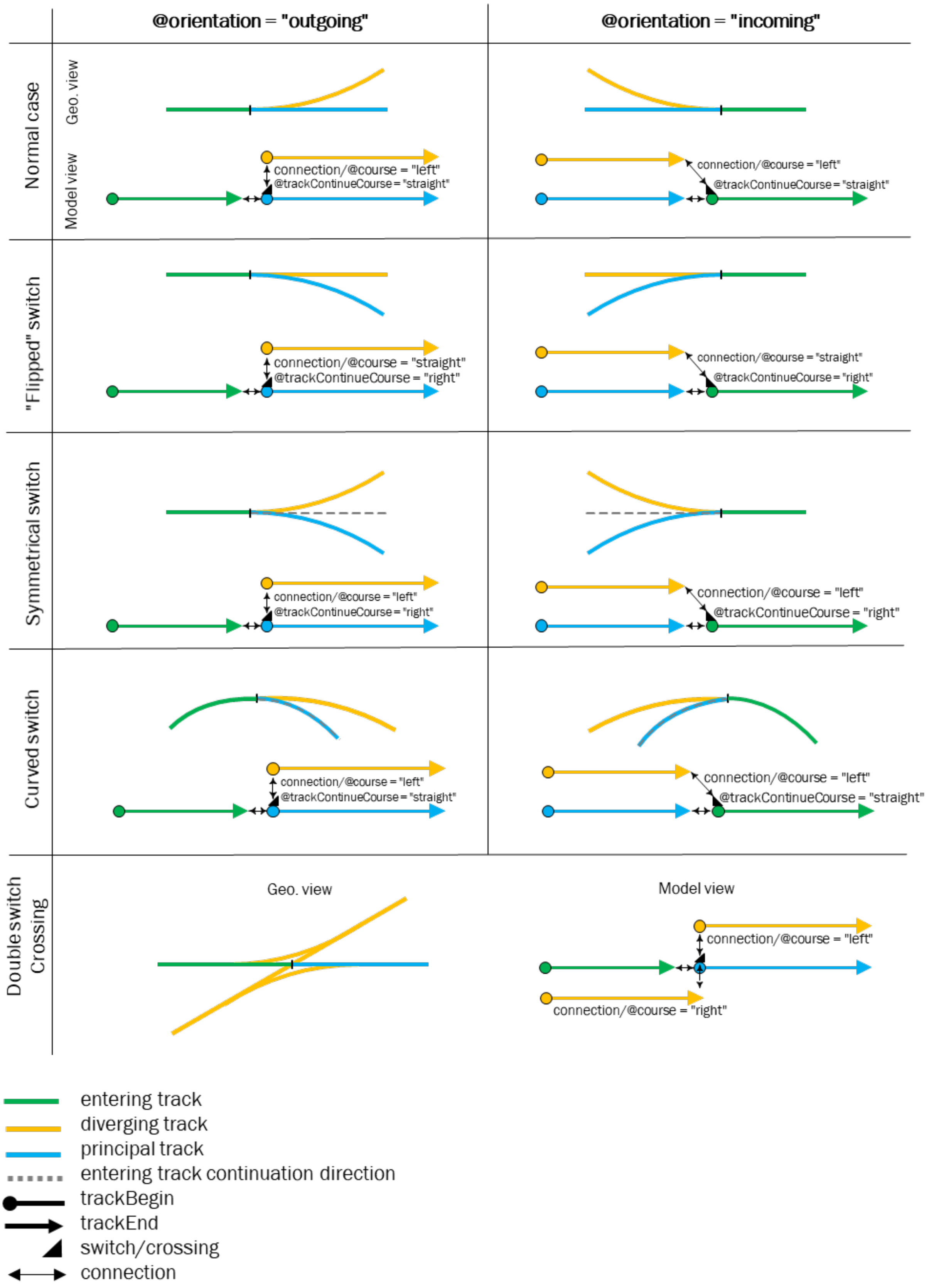

@orientation = "outgoing"
@orientation = "incoming"
Normal case
Geo. view
Model view
connection/@course = "left"
@trackContinueCourse = "straight"
connection/@course = "left"
@trackContinueCourse = "straight"
"Flipped" switch
connection/@course = "straight"
@trackContinueCourse = "right"
connection/@course = "straight"
@trackContinueCourse = "right"
Symmetrical switch
connection/@course = "left"
@trackContinueCourse = "right"
connection/@course = "left"
@trackContinueCourse = "right"
Curved switch
connection/@course = "left"
@trackContinueCourse = "straight"
connection/@course = "left"
@trackContinueCourse = "straight"
Double switch
Crossing
Geo. view
Model view
connection/@course = "left"
connection/@course = "right"
entering track
diverging track
principal track
entering track continuation direction
trackBegin
trackEnd
switch/crossing
connection

Figure 13: Example of a switch and a crossings placement on the track and it's relation to the connecting tracks in both geographic and model view

The principal track is determined by the track leg that has the value, in the following order:

1. The track leg with the same <track>@type as the entering track
2. The track leg with the same <track>@name as the entering track.
3. The track leg with the <connection>@radius closest to the entering track <radiusChange>@radius. Where no <radiusChange>@radius value is available, assume the entering track is straight (radius="0"/∞).

If extracting data from a schematic track plan, not knowing the radii, assume that the principal track according to the rules above is the track drawn in a straight line from the entering track.

If the information available is either not conclusive (in the case where multiple legs have the same value) or the information is not available move down the order to define the principal track.

Attribute of <switch>/<crossing>	Definition in accordance to Norwegian usage
@pos	The position of the switch is the beginning of the switch. The position of a crossing is the theoretical crossing (approximately in the middle of the crossing). The beginning of the switch is preferably the tongue tip. If this position is not available, the position of the stock rail joint should be used. There can either be no information, information in the metadata if identically applicable for all switch/crossing elements in the file, or information in the @description for each element in the railML as to which system is used.
@trackContinueCourse	Defines the course of the principal track in relation to the entering track, as seen in track direction. If the principal track has the same radius as the entering track, the value is "straight". If the principal track curves more to the left (or less to the right) than the entering track, the value is "left", and vice versa for "right". Example: if the entering track is in a −300m radius left sided curve and the principal track is also in a −300m radius left sided curve the value is "straight".
@trackContinueRadius	Radius, in meters, of the curve with which the principal track continues through the crossing. This is a signed value, wherein negative values indicate a center of an arch left of the principal track, and positive values indicate center of an arch right of the principal track, relative to track heading ("up" direction).
@normalPosition	Refers to the baseline position of a switch for route setting. Defines the main track in a station.
@length	Build length of the switch/crossing from its build beginning (see @pos) to its build end in mm. Defined according to TRV from stock rail joint to the joint behind the cross at the end of the switch/crossing.
@model	Declaration for the type of switch/crossing. Bane NOR refers to the switch drawings as UID, e.g. "Sk 3470". Use the main drawing type (for the complete switch). PS. The distances between the stock rail joint, the tongue blades and the theoretical cross is known for each model type. Therefore, a mapping between different positions of the pos is possible.

Furthermore, the <switch> and <crossing> elements have a sub-element <connection> for each diverging track with the following attributes, which have a Norwegian definition extension:

Attribute of <connection>	Definition in accordance to Norwegian usage
@course	Defines the course of this diverging track in relation to the entering track, as seen in track direction. If this diverging track has the same radius as the entering track, the value is "straight". If this diverging track curves more to the left (or less to the right) than the entering track, the value is "left", and vice versa for "right".
@orientation	This railML2.4 core attribute is interpreted in track direction. The value "incoming" refers to a merging and the value "outgoing" refers to a splitting of the tracks in the switch.
@maxSpeed	This refers to the maximum speed allowed while passing over the switch/crossing from this diverging track to the track the switch/crossing is placed on and vice versa. The speed over a crossing between the two diverging tracks is not defined. The speed replaces the <speedChange>@vMax value for the @length of the switch. If no @length is set, assume a length of 50 meters. Please also refer to chapter 3.9.3 Flow of "<*Change>" elements along the track(s), for correct usage of the attribute.
@passable	Denotes if it is possible to pass over the switch/crossing between this diverging track and the track the switch/crossing is placed on. This does not restrict the movement over a crossing between the two diverging tracks A special case occurs when setting a switch/crossing in a reduced state in railML2.4nor. This usually means (pad)locking a switch/crossing, i.e. disabling the movement of the blades. The following needs to be indicated: • <state>@disabled="true" • <connection>@passable=true/false for the applicable connection Note that if @disabled="true" on a switch and @passable="true" on the diverging track’s connection, implies that the connection between the entering track and the principle track is not passable.
@radius	Radius of the curve with which the switch rails converge into or diverge from the principal track, measured from an imaginary centre of a circle. Value will usually be non-zero. Signed value, wherein negative values indicate a centre of an arch left of the principal track, and positive values indicate centre of an arch right of the principal track, relative to track heading ("up" direction).

For a further, more generic description see also the Norwegian technical design rules: Teknisk regelverk – TRV.

4.3.2 Attribute extension

The elements <switch> and <crossing> were extended by the following attributes in railML2.4nor:

Attribute of <switch>/<crossing>	Description	Type
@nor:remoteOperated	Indicates if a switch can be operated from a remote location. This value is almost always set in combination with @nor:remoteIndicated.	xs:boolean

Attribute of <switch>/<crossing>	Description	Type
@nor:remoteIndicated	Indicates that the switch can relay information about the locked position of its tongues (left/right/straight) to a remote location. If only this value is set, a switches tongues position can be observed remotely, but the position cannot be changed.	xs:boolean
@nor:lockRef	Reference to a lock, locking the switch.	rail:tGenericRef
@nor:clearance	Distance in meters between the position of a switch/crossing (see @pos) and to the position of its clearance point behind the switch/crossing. The value is always positive. PS. For a switch with @orientation="incoming" you need to calculate backwards for the position of the clearance.	xs:double

4.3.3 Code example

Switch:

```
<switch absPos="200.0" code="KO-SPV-803680" controllerRef="id5"
description="Naming rule: [Arabic number]" id="sw41" name="1"
nor:change="new" nor:clearance="50.0" nor:lockRef="id10"
nor:remoteIndicated="true" nor:remoteOperated="true"
normalPosition="straight" pos="0.0" trackContinueCourse="straight"/>
   <connection course="right" id="c41-43" maxSpeed="40.0"
orientation="outgoing” ref="c43"/>
</switch>
```

Crossing:

```
<crossing absPos="800.0" code="KO-SPV-803681" controllerRef="id5"
description="Naming rule: [number/number]" id="sw27" name="50/52"
nor:change="new" nor:clearance="50.0" nor:remoteIndicated="true"
nor:remoteOperated="true" normalPosition="straight" pos="0.0"
trackContinueCourse="straight" type="doubleSwitchCrossing">
   <connection course="left" id="c27-32" maxSpeed="40.0"
orientation="outgoing" ref="c32"/>
   <connection course="right" id="c27-36" maxSpeed="40.0"
orientation="incoming" ref="c36"/>
</crossing>
```

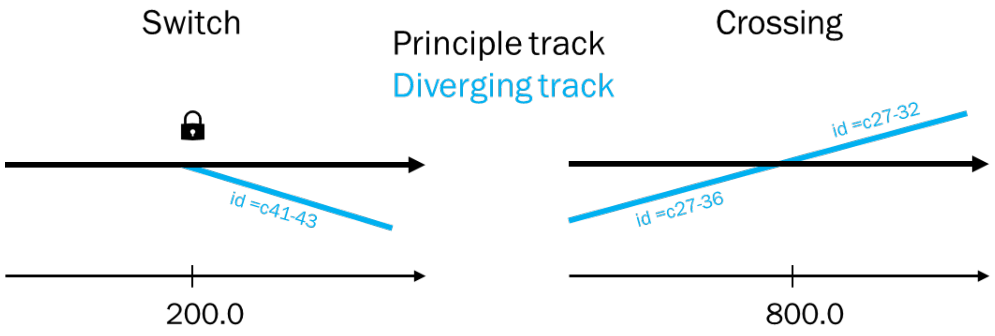


Figure 14: Switch and crossing elements as described in code

4.4 The element <signal> - modelling of boards and signals

4.4.1 Definition extension in accordance to Norwegian usage

railML2 sees both light signals and boards as signals.

Use of @switchable

railML2.4 does not offer the possibility to define boards as an individual element. Hence, in railML2.4nor the, in core railML2.4 deprecated attribute, attribute @switchable becomes mandatory for <signal> and differentiates boards and signals:

- Signals: <signal>@switchable = “true”
- Boards: <signal>@switchable = “false”

@switchable is seen as in the switching ability of the physical element (signal) on the track.

The attribute @switchable in the sub elements is not used.

Use of @ruleCode

All Norwegian signals have a unique @ruleCode value. This follows the naming rule:

NOR:[legislation/rulebook]:[signal ID]

The signal ID for boards is the signal (aspect) number. Use signal number without "S" for signal. For example: "NOR:TJN:67B" (level crossing announcement signal).

The signal ID for light signals is the paragraph of the signal (as it can have multiple signal aspect numbers) Use § with "dash". For example: "NOR:TJN:§8-10" (main entry signal).

The combined signal type uses a combined signal ID in @ruleCode. For example: "NOR:TJN:§8-13+14" (combined main block and distant signal).

A complete list of all implemented boards and signals with mapping from the Norwegian values can be found in the railML2.4nor Infrastructure Model Excel sheet. A description of the different Norwegian signals is available in the Norwegian regulation.

Use of @type and @function

The values for the attribute @type and @function do not map all Norwegian signals (and boards) in a generic manner. We have chosen not to extend the generic model in railML2 with "other:" values, with one exception (this is covered in railML3). Therefore, only the most basic light signals, as part of core railML2, make use of @type.

For signals of <signal>@type="main" and "combined" the use the attribute @function is mandatory. The attribute is also only used here. For end of route signals (both conventional and in ETCS) we have added the enumeration value @function="other:end".

Use of @sigSystem

railML2.4nor uses the, in core railML2.4 deprecated, attribute @sigSystem with the value “simplified” for two Norwegian signal types and "temporary" for four Norwegian signal types. See Excel sheet for the specific signals. Other signals do not use the @sigSystem attribute. Signals with use of the sub element <etcs> are considered part of the ETCS signaling system. All other signals are considered conventional optical signals.

Multiple signals on the same pole

In the case of multiple signals located on the same pole, the following attributes of the concerned elements will have the same values: <signal>@absPos, <signal>@nor:mounted and <signal>@nor:side.

4.4.2 Attribute extension

railML2.4nor uses the sub-elements defined in railML2.4 to further define the properties of a signal and additionally introduces further attributes of the <signal> element.

Attribute of <signal>	Description	Type
@nor:lamps	Defines the number of lamps a signal has. The Norwegian sector refers to aspects with this attribute. This element is only relevant for signals, i.e. when the attribute @switchable="true".	xs:integer
@nor:mounted	Defines how a signal/panel is constructed.	xs:enumeration
@nor:side	Defines on which side of a track the signal or board is placed. The side is interpreted in track direction.	xs:enumeration

4.4.3 Code example

Board:

```
<signals>
   <signal absPos="900.0" code="KO-SKT-800934" description="Naming rule:
[just value where applicable, no name]" dir="up" id="id14" name=""
nor:mounted="pole" nor:side="right" pos="100.0" switchable="false"
ruleCode="NOR:TJN:60F" virtual="false"/>
</signals>
```

Signal:

```
<signals>
   <signal absPos="100.0" code="SA-SIG-804897" controllerRef="id5"
description="naming rule: [signal letter]+[signal number]" dir="up"
function="home" id="id11" name="A101" nor:change="new" nor:lamps="3"
nor:mounted="pole" nor:side="right" ocpStationRef="id4" pos="0.0"
sigSystem="conventional" ruleCode="NOR:TJN:§8-10" sight="50.0"
switchable="true" type="main" virtual="false"/>
</signals>
```

4.5 The element <controller>

A controller sets the track side objects in a certain safe position or aspect defined by the routes. For instance, it sets the switch either in the position "left" or "straight". Usually this is referred to as an interlocking unit. However, interlocking is not modelled in railML2, which is why the term "controller" is used. A controller is located between an OCP and the objects the controller controls. It controls the objects on behalf of an OCP.

The controller in accordance to Norwegian usage is defined in TRV and TJN.

4.5.1 Definition extension in accordance to Norwegian usage

railML2.4nor allows the controller to be implicitly part of the OCP in very simple models. However, in most models a separate <controller> element is modelled. In most cases one OCP has one controller. Larger stations can have multiple controllers for designated areas. Modern electronic controllers can control objects belonging to multiple OCPs.

The controller has no defined area. The area can implicitly follow the defined area of an OCP it references, or individual objects can reference the controller they belong to, e.g. <switch>, <crossing>, <levelCrossing>, <signal>, <trackCircuitBorder>, <trainDetector> and <lock>. Some elements reference the controller indirectly, e.g. the <derailer> via the reference to <lock> or <nor:route> via the reference to <entrySignal>.

4.5.2 Attribute extension

The following new attributes of <controller> are introduced in railML2.4nor:

Attribute of <controller>	Description	Type
@nor:routeSetTime	Represents the average time for a route to be set and locked (including averaged switch set time for the whole controller), from the moment the control command is issued until a point when the route has been established and confirmed to the operator. As this is a controller specific value (all routes are considered to have the same set-time value) it is not an attribute of <nor:route> and only specified once as attribute of controller.	xs:duration
@nor:typicalThrowTime	Represents the time from the moment a command is issued until a new position is confirmed and indicated within the controller's main machine interface. This time includes interlocking processing, switch unlocking, blade switching, switch locking.	xs:duration

4.5.3 Code example

```
<controllers>
   <controller code="SA-SIK-800400" description="desc" id="id5" name="OSL"
nor:routeSetTime="5" nor:controllerSystem="NSB-77"
technologyType="electrical" nor:typicalThrowTime="10">
       <ocpRef ref="id4"/>
   </controller>
</controllers>
```

4.6 The element <baliseGroup>

4.6.1 Definition extension in accordance to Norwegian usage

railML2.4 defines a <balise> and a <baliseGroup> element. railML2.4nor defines clearer how these are to be modelled in relation to each other. This is done in three different ways:

1. **The balise group**
   This refers exclusively to the mapping of a balise group's location and its function. It does not contain any single balises.
   The location is modelled by placing one dummy balise to indicate its position. The dummy balise uses only the attributes @dir, @id and @pos.

```
<balises>
   <balise dir="up" id="id24-dummy" pos="250.0"/>
   <baliseGroup code="SA-ATC-845612" description="Naming Rule: [signal
name]" id="id24" name="A101" nor:switchable="true" type="infill">
       <baliseRef ref="id24-dummy" sequence="1"/>
   </baliseGroup>
</balises>
```

2. **The balise group with individual defined functional balises**
   In this case all balises that are part of a balise group are placed. The location is identical (same pos value) for all balises as only the location of the balise group is known. The name indicates the functional type (conventional in Norway: "P","A","B" or "C") The balises only use the attributes @dir, @id, @name and @pos.

```
<balises>
   <balise dir="up" id="id13-P" name="P" pos="250.0"/>
   <balise dir="up" id="id13-A" name="A" pos="250.0"/>
   <balise dir="up" id="id13-B" name="B" pos="250.0"/>
   <balise dir="up" id="id13-C" name="C" pos="250.0"/>
   <baliseGroup code="BD" description="Naming rule: [signal name]"
id="id13" name="A101" nor:switchable="true" type="other:norFF">
       <baliseRef ref="id13-P" sequence="1"/>
       <baliseRef ref="id13-A" sequence="2"/>
       <baliseRef ref="id13-B" sequence="3"/>
       <baliseRef ref="id13-C" sequence="4"/>
   </baliseGroup>
</balises>
```

3. **Balise groups with individual balises with known locations**
   In this case all balises that are part of the balise group are placed with their real location (different pos values). The "A" or the first ETCS balise indicates the position (pos) of the balise group. Distances between balises vary between 2,3 and 3,5 meters according to Signal/Bygging/ATC – Teknisk regelverk (banenor.no).

```
<balises>
   <balise absPos="247.0" code="BD" description="Naming rule: [signal
name]" dir="up" id="id13-P" name="P" pos="247.0"/>
   <balise absPos="250.0" code="BD" description="Naming rule: [signal
name]" dir="up" id="id13-A" name="A" pos="250.0"/>
   <balise absPos="253.0" code="BD" description="Naming rule: [signal
name]" dir="up" id="id13-B" name="B" pos="253.0"/>
   <balise absPos="256.0" code="BD" description="Naming rule: [signal
name]" dir="up" id="id13-C" name="C" pos="256.0"/>
   <baliseGroup code="BD" description="Naming rule: [signal name]"
id="id13" name="A101" nor:switchable="true" type="other:norFF">
         <baliseRef ref="id13-P" sequence="1"/>
         <baliseRef ref="id13-A" sequence="2"/>
         <baliseRef ref="id13-B" sequence="3"/>
         <baliseRef ref="id13-C" sequence="4"/>
   </baliseGroup>
</balises>
```

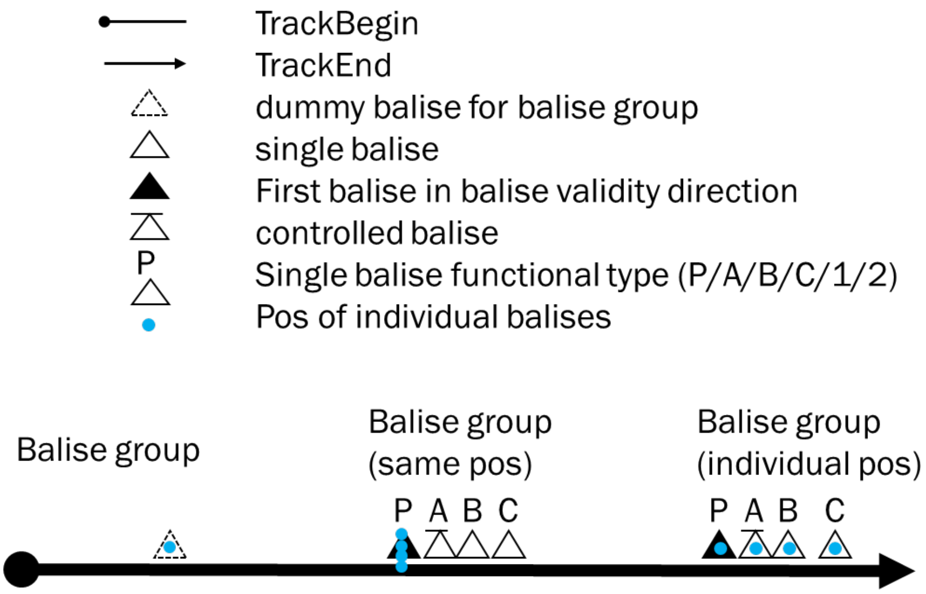

TrackBegin
TrackEnd
dummy balise for balise group
single balise
First balise in balise validity direction
controlled balise
P
Single balise functional type (P/A/B/C/1/2)
Pos of individual balises
Balise group
Balise group
(same pos)
P A B C
Balise group
(individual pos)
P A B C

Figure 15: Illustration of balise modelling

4.6.2 Attribute extension

railML2.4nor extends the <baliseGroup> element by one attribute:

Attribute of <baliseGroup>	Description	Type
@nor:switchable	Indicates that information in the balise group is switchable. For the balise group model 1, the indication is valid only for the balise group. For the balise group with functional balises with or without their known location (II or III) it indicates that the "A" or the first ETCS balise is switchable.	xs:boolean

Furthermore, railML2.4nor extends the enumeration value list of the @type attribute of the <baliseGroup> element with Norwegian values: "other:norX", with X being the code for the Norwegian functional type (e.g. “other:norFF”, “other:norA”, “other:norBU”, etc.) . See Excel sheet "railML2.4nor Infrastructure Model" for complete value list.

4.7 The elements <speedProfile> and <speedChange> - modelling of speed

4.7.1 Definition extension in accordance to Norwegian usage

In railML2.4nor, two in railML2.4 existing elements are used to model speed restrictions. The values of the attributes of these elements are partially specific to Norwegian railways and thus explained in the following:

The <speedProfile> element

In railML2.4nor, the speed for trains is modelled via speed profiles. The speed profiles are specified for different types of trains. railML2.4nor uses the railML2.4 <speedProfile>@name attribute to indicate the speeds accordingly. The @name attribute is defined as a string in railML2.4. The allowed string values are restricted to the following in railML2.4nor:

Value of <speedProfile> @name	Definition in accordance to Norwegian usage
“basis”	Baseline value, valid for all trains, if no other specific speed restriction value is given. The speed profile always has an additional attribute @influence="increasing".
“nor:pluss”	This speed profile applies to rolling stock with enhanced capabilities. For which exact trains this applies is defined in railML2.4 Rolling Stock schema. The speed profile always has an additional attribute @influence="increasing". If this speed profile is not given, the speed limit of the basis profile is valid.
“nor:krenge”	This speed profile is applicable to rolling stock enhanced with the capability of tilting. For which exact trains this applies is defined in railML2.4 Rolling Stock schema. The speed profile always has the attribute @influence="increasing". If this speed profile is not given, the speed limit of the nor:pluss or basis profile is valid. *Note that compliance to these values does not guarantee a train to be allowed to run under “nor:krenge”. A speed profile certification is required.*
“nor:local”	This speed profile is applicable to all rolling stock. It is defined in SJN for local areas (SJN 2.2.2) and is not signalised. It is mandatory to also indicate the speedProfile with the following: • <speedProfile>@influence="decreasing" • <speedProfile>@verbalConstraint=” SJN defined condition” • <speedChange>@signalised="false"
“nor:temporary”	This speed profile is applicable to all rolling stock. It is temporary and signalised. It is mandatory to also indicate the speedProfile with the following: • <speedProfile>@influence="decreasing" In order to define the time validity please use the <state> element of the corresponding <soeedChange>,
“nor:conditional”	This speed profile is applicable to certain rolling stock conditions. It is defined in SJN for local areas (SJN 2.2.1) and is not signalised. It is mandatory to also indicate the speedProfile with the following: • <speedProfile>@influence="decreasing" • <speedProfile>@verbalConstraint= “SJN defined condition” • <speedChange>@signalised="false"
“nor:avalanche”	When approaching certain defined areas with a high likelihood of avalanches, the speed is restricted until the driver has made sure that the track is clear. This speed profile is applicable to all rolling stock for areas defined in SJN section 2.3. Even though these avalanche restrictions are not signalised, 8 out of the 9 sections defined in SJN per 14.03.2019 specify the same speed restriction as the signalised “basis” profile. It is mandatory to also indicate the speed profile with the following: • <speedProfile>@influence="decreasing" • <speedProfile>@verbalConstraint= SJN defined condition • <speedChange>@signalised="false"

Please note that “nor:x” profiles are not omni-present, they are defined only for parts of the lines or even segments of the tracks. For example, currently, only the lines Sørlandsbanen and Dovrebanen have “nor:krenge” speed profiles. A continuous combination of different speed changes for a certain speed profile is called a speed section. The speed section is also used in railML3. A speed section is started with a <speedChange>-element with a @vMax value in km/h and ends with an @vMax="end" value. In between the start and the end of a speed section, the speed section can alter its speed value by introducing new <speedChange>@vMax values in km/h. Figure 15 illustrates this issue.

railML2.4nor allows the speed sections of a speed profile to be modelled in two ways. The preferred alternative is that only the “basis” profile is continuous throughout the model, while the other applicable speed profiles are only defined where they really exist along the track. Where such a speed profile ends, there will be a <speedChange> for this profile with @vMax="end", and the same profile will start again later with a new @vMax value. The second alternative is to define all speed profiles as continuous. Where the speed profile no longer exists along the track, the writing system must determine the correct speed profile to fall back to and use the same @vMax value as for the fall-back profile. For “nor:krenge” the fall-back is “nor:pluss”, and for “nor:pluss” the fall-back is “basis”.

Except for the “end” value the @vMax value is always a speed in km/h. The corresponding <speedProfile> is referred to from all <speedChange> elements.

Please also read chapter 3.11.1 for correct placement of <speedChange> along the tracks.

In case a speed restriction applies while passing through the diverging track of a switch, the <connection>@maxSpeed attribute must be used. Please read chapter 4.3 for correct usage of <switch> and <crossing>.

How to choose the right speed profile if multiple speed profiles are present

If more than one speed profile is present (see Figure 15), the permitted speed for a specific train is determined in the following steps:

1. Consider all available speed profile(s) for the train.
2. Consider all applicable conditional speed profiles, if present.
3. Of all the available @vMax values select the minimum of: all “decreasing” speed profiles and the highest “increasing” speed profile.
4. Consider also <switch> and <crossing>@maxSpeed (see section 4.3.1) and <route>@proceedSpeed (see section 4.8.3).

Available speed profile(s) for the train
Only the speed profiles “basis”, “nor:local” and “nor:avalanche” are valid for all trains. Which trains that can use “nor:pluss” and “nor:krenge” is defined in railML2.4 Rolling Stock scheme.

Applicable conditional speed profile
The “nor:conditional” depends upon the set string condition in the value for @verbalConstraint. The reading system needs to interpret for which trains “nor:conditional” should apply. Also, the reading system needs to decide how to apply the “nor:avalanche” speed profile. This profile applies to all trains but is lifted as soon as the driver can see that the track is clear. Per 14.03.2019 there is only one short “nor:avalanche” section in the Norwegian railway network that deviates from the signalised “basis” speed profile.

Increasing and decreasing speed profiles and value of @vMax
The permitted speed is determined from all the applicable speed profiles by selecting the minimum of: all @vMax values of speed profiles with @influence="decreasing" and the highest @vMax value of any speed profile with @influence="increasing". In other words, first find the lowest value of @vMax for all “decreasing” speed profiles. Then find the highest value of @vMax for all “increasing” speed profiles. Finally, take the lowest of these two values.

Value of <speedProfile> @influence	Definition in accordance to Norwegian usage
“increasing”	This speed profile increases the permitted speed. If multiple “increasing” speed profiles are applicable, select the one with the highest @vMax value.
“decreasing”	This speed profile decreases the permitted speed. If multiple “decreasing” speed profiles are applicable, select the one with the lowest @vMax value. If this value is lower than the speed of an “increasing” speed profile it overrides that speed.

The <speedChange> element
In addition to the <speedProfile>@name attribute, the speed profile is also indicated in the attribute @etcsTrainCategory of the <speedChange> element. The values of the <speedChange>@etcsTrainCategory are railML2.4nor specific and a mapping of the values of the international train category numbers, adapted accordingly to the definition of the attribute in the railML wiki. The values are defined as:

<speedProfile> @name	Value of <speedChange>@etcsTrainCategory
"basis"	5
"nor:pluss"	6
"nor:krenge"	13
"nor:temporary"	The attribute is not to be used for this speed profile.
"nor:conditional"	The attribute is not to be used for this speed profile.
"nor:avalanche"	The attribute is not to be used for this speed profile.
"nor:local"	The attribute is not to be used for this speed profile.

In railML2.4nor the speed information modelled via <speedChange> elements must be defined along the full extent of all tracks. Thus, it is mandatory to place a <speedChange> in the beginning of a model with a @dir="up" and on the end of a model with a @dir="down" value or at a mileage direction change once for each direction (See also 3.11).

Following the logics of the model, the <speedChange> that is placed on a <trackBegin> or <trackEnd> that is not connected to another track will have to have the following values:

Position of <speedChange>	Value of <speedChange>@pos	Value of <speedChange>@dir
On <trackBegin>	"0"	"up"
On <trackEnd>	Value equal to the <trackEnd>@pos value	"down"

Further information about how speed is modelled is available in the railML Forum post "Hierarchy of overlaying speed profiles and national vs. generic speed profiles" and the <speedChange> example in the use of @dir in railML. Please note here the requirement to end speed bands of a given <speedProfile>, if the value of the <speedProfile> in question falls down to/equals the value of the basis <speedProfile>, with speedChange@vMax="end" and **not** to copy the basis <speedProfile> value, see illustration in figure 15.

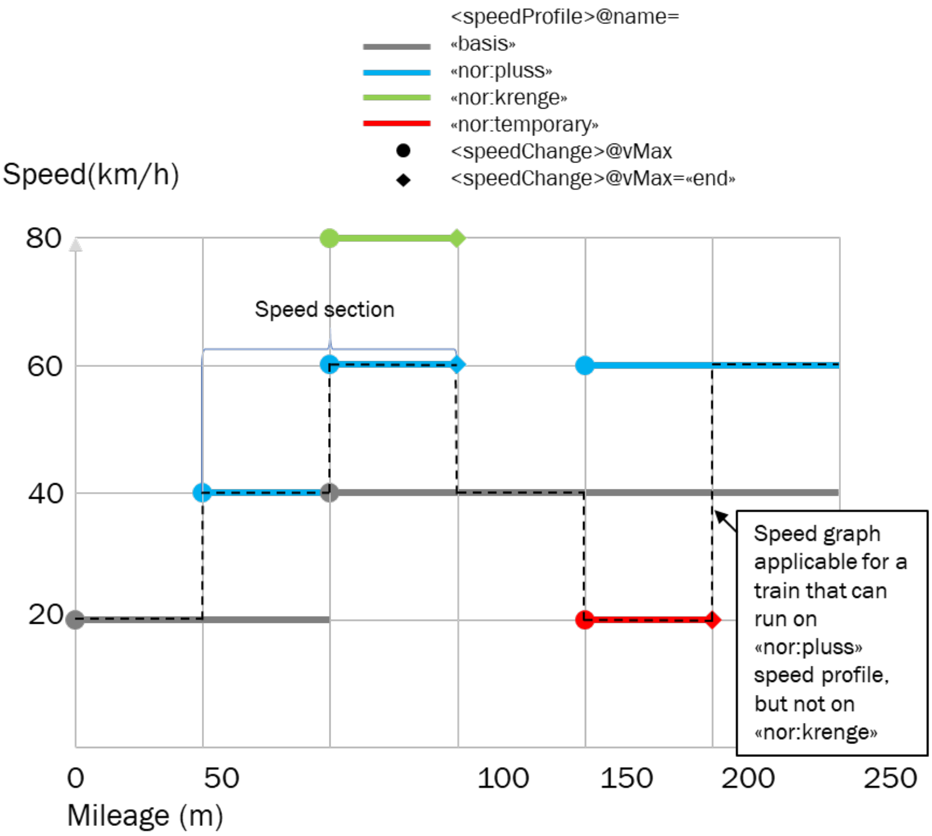


Figure 16: Illustration of speed sections and profiles in the preferred alternative

4.7.2 Code example

```
<speedChanges>
       <speedChange absPos="0.0" dir="up" etcsTrainCategory="5" id="id1a"
pos="0.0" profileRef="spprf1" vMax="20.0"/>
       <speedChange absPos="50.0" dir="up" etcsTrainCategory="6" id="id2a"
pos="50.0" profileRef="spprf2" vMax="40.0"/>
       <speedChange absPos="100.0" dir="up" etcsTrainCategory="5" id="id3a"
pos="100.0" profileRef="spprf1" vMax="40.0"/>
       <speedChange absPos="100.0" dir="up" etcsTrainCategory="6" id="id3b"
pos="100.0" profileRef="spprf2" vMax="60.0"/>
       <speedChange absPos="100.0" dir="up" etcsTrainCategory="13"
id="id3c" pos="100.0" profileRef="spprf3" vMax="80.0"/>
       <speedChange absPos="150.0" dir="up" etcsTrainCategory="6" id="id4a"
pos="150.0" profileRef="spprf2" vMax="end"/>
       <speedChange absPos="150.0" dir="up" etcsTrainCategory="13"
id="id4b" pos="150.0" profileRef="spprf3" vMax="end"/>
       <speedChange absPos="200.0" dir="up" etcsTrainCategory="6" id="id5a"
pos="200.0" profileRef="spprf2" vMax="60.0"/>
       <speedChange absPos="200.0" dir="up" id="id5b" pos="200.0"
profileRef="spprf4" vMax="20.0"/>
       <speedChange absPos="250.0" dir="up" id="id6a" pos="250.0"
profileRef="spprf4" vMax="end"/>
       </speedChanges>
[…]
<speedProfiles>
    <speedProfile id="spprf1" influence="increasing" name="basis"/>
    <speedProfile id="spprf2" influence="increasing" name="nor:pluss"/>
    <speedProfile id="spprf3" influence="increasing" name="nor:krenge"/>
    <speedProfile id="spprf4" influence="decreasing" name="nor:temporary"/>
</speedProfiles>
```

4.8 The element <nor:route>

A route is defined as a configured piece of infrastructure, which allows a train to pass a set railway path safely and legally. The use cases "capacity planning" and "routes for timetable simulation" have a need to model routes without interlocking. This means that there is a list of routes. The routes are described with their properties independently from one another. Routes are currently not represented in core railML2.4, which is why a new high-level grouping element <nor:routes> with its specific element <nor:route> is introduced in railML2.4nor.

4.8.1 Definition in accordance to Norwegian usage

As illustrated in the figure below a route can be grouped in four parts:

1. The path of the route and the route properties (marked blue in the illustration).
2. If applicable, the approach speed in front of the route (marked orange in the illustration).
3. If applicable, the overlap behind the route (marked red in the illustration).
4. Its release groups: Grouping of track sections to allow a partial release (marked green in the illustration).

Nr	Attribute
1	@approachPointRef
-	@id, @name, @code, @description, @nor:change, <nor:state@status>
2	@entryRef
3	<switchAndPosition>@switchRef, @switchPosition
4	@exitRef
5	<overlapSwitchAndPosition>@switchRef, @switchPosition
6	@overlapEndRef
7	@releaseTriggerRef @releaseTriggerHead

Nr	Attribute
A	@approachSpeed
B	@proceedSpeed, @releaseSpeed
C	@overlapValidityTime
D	<nor:releaseGroup>/<nor:areaRef>@ref (TVDs)

At other	Attribute
controller	@nor:routeSetTime, @nor:typicalThrowTime

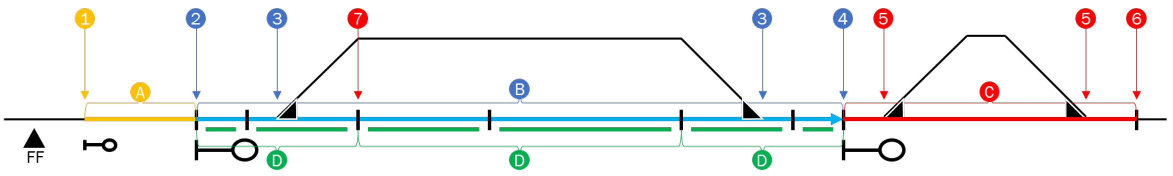


Figure 17: Illustration of route modelling

In order to unambiguously describe the path of the route and the overlap, the new <nor:route> element has a <nor:switchAndPosition> and a <nor:overlapSwitchAndPostion> sub-element.

4.8.2 Sub-elements of the element

<nor:switchAndPosition>
The path of the route is modelled through the sub-element <nor:switchAndPosition> by defining all switches (@switchRef) with their set course (@switchPosition) along the route.

As some switches are not part of the routes path but have to be controlled to perform flank protection for the route, they have to be included and declared as such.

Attribute of <nor:switchAndPosition>	Description	Type
@switchRef	References all switches and crossings that are part of the route, or that must have a specific position before the route can be set.	rail:tGenericRef
@switchPosition	Provides the required position of the referenced switch or crossing. The position is determined independently of the route direction, using the same coordinate system as for the physical switch and crossing model described in 4.3.1. *For example, if a <switch> or <crossing> is required to be in a position that connects the entering and the principal track (as defined in 4.3.1), use the @trackContinueCourse value of the <switch> or <crossing>. Note that this allows traversing both axes of a <crossing>. For a <switch>that is required to be in a deflecting position, use the @course value of the relevant <connection>.* For a <crossing> in the deflecting position use the @course value of the **outgoing** <connection>. *Note that @switchPosition=“left” or “right” does not indicate a direction but only that the position of the crossing is “deflecting”. The direction is given by the path of the route.* See Figure 17 and table below for example values.	xs:enumeration

@flankProtection	Set to “true” if the referenced switch/crossing is not part of the path of the route but gives the route flank protection.	xs:boolean (optional)

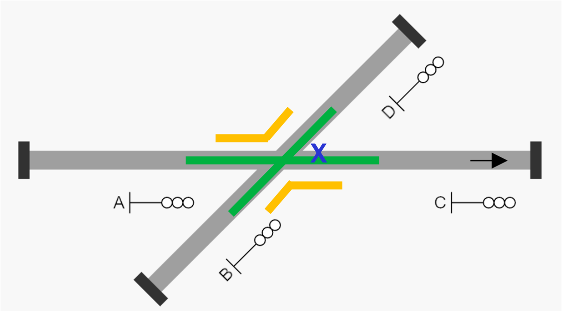


Figure 18- Illustration of routes over a crossing (track edge direction left to right, orientation=”outgoing” course=”left”)

```
<crossing id="sw27" name="50/52" normalPosition="straight" pos="0.0"
trackContinueCourse="straight" type="doubleSwitchCrossing">
 <connection course="left" id="c27-32" maxSpeed="30.0"
orientation="outgoing" passable="true" ref="c32"/>
              <connection course="right" id="c27-36" maxSpeed="30.0"
orientation="incoming" passable="true" ref="c36"/>
            </crossing>
```

Route (from signal - to signal)	@switchPosition
A-C	straight
A-D	left
B-C	left
B-D	straight
C-A	straight
D-A	left
C-B	left
D-B	straight

<nor:overlapSwitchAndPosition>
The path of the overlap is modelled by defining all switches, @switchRef, with their set course, @switchPosition, along the path of the overlap.

Attribute of <nor:overlapSwitchAndPosition>	Description	Type
@switchRef	References all switches/crossings that are part of the overlap of the route.	rail:tGenericRef
@switchPosition	Defines the position of the referenced switches/ crossings.	xs:enumeration

<nor:releaseGroup>
This element refers to an extended feature of a route, which gives the optional possibility to group track sections into a release group. This is used to model partial route releases. The release group is released when all track sections of a group are cleared.

All the track sections listed in <nor:releaseGroup> also form the track sections that have to be controlled for the route to be set. As some track sections are included to form flank protection (and are thus not part of the routes path), they have to be declared as such.

The <nor:releaseGroup> element references track sections by a sub-element <nor:trackSectionRef> with the following attribute:

Attribute of <nor:trackSectionRef>	Description	Type
@ref	Reference to the track section. This is an area of @type="trackSection", which is part of the release group.	rail:tGenericRef
@flankProtection	Set to “true” if the referenced track section is not part of the path of the route but gives the route flank protection.	xs:boolean (optional)

<nor:states>
This sub-element was added to railML2.4nor as the core railML2.4 element <state> does not apply to <nor:route>. The <nor:states> element is structured in the same way as the core <states> and its sub-element <state> with the nor-extensions.

4.8.3 Attributes of the element

The new <nor:route> element consists of the common (see chapter 3.7) and the following attributes:

Attribute of <nor:route>	Description	Type
@proceedSpeed	The maximum speed to be kept over the entire length of the route. The speed can be higher or lower than the track (<speedChange>@vMax) or deflecting switch (<connection>@maxSpeed) speed. The lowest value is applicable.	rail:tSpeedKmPerHour
@releaseSpeed	Speed under which a train is unsupervised from its braking curve (where applicable).	rail:tSpeedKmPerHour
@entryRef	Defines the start of the route. No constraints, but usually a main signal.	rail:tGenericRef
@exitRef	Defines the end of a route. Usually a main signal.	rail:tGenericRef
@conditional	Condition under which the route applies. If no condition is set, the route can be applied freely as needed by the simulation/TMS.	xs:enumeration

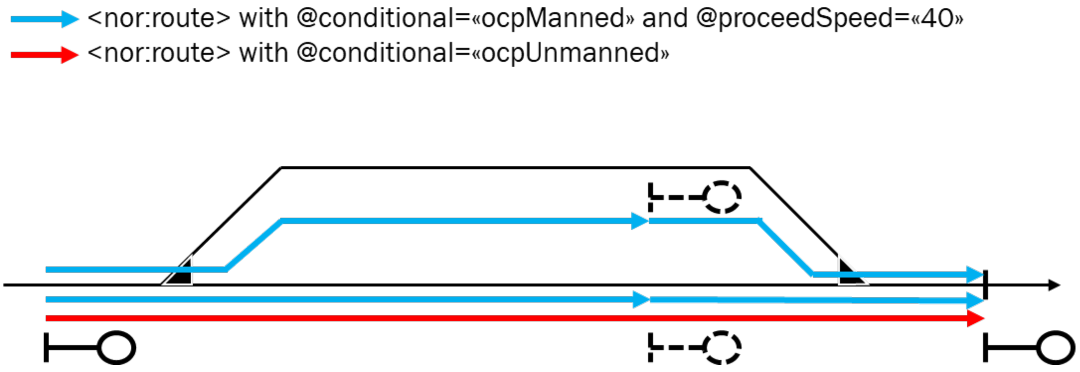

<nor:route> with @conditional=«ocpManned» and @proceedSpeed=«40»
<nor:route> with @conditional=«ocpUnmanned»

Figure 19: Example for <ocp>@type=”station” with <controller>@model=”enkelt Innkjør”

The following attributes are applicable for a route if there is a speed restriction in front of the entry signal (@entryRef) and in the case the route has a restrictive aspect. I.e. the distant signal of the entryRef signal shows “expect stop” and the signal referenced from @entryRef shows a stop aspect:

Attribute of <nor:route>	Description	Type
@approachPointRef	Position, at which the train complies to the approach speed restriction (usually the distant signal).	rail:tGenericRef
@approachSpeed	The value for the restrictive approach speed in km/h. The approach speed is defined in TRV 550-10-3.6. The approach speed is applied in front of a route with a restrictive aspect (usually showing stop).	rail:tSpeedKmPerHour

The following attributes are applicable if the route has an overlap for protection purposes:

Attribute of <nor:route>	Description	Type
@overlapEndRef	References the end of the overlap. This is usually a track circuit border or a train detector. To be referenced to the OCP id for modelling of a safety element (like a cross locking element).	rail:tGenericRef
@overlapValidityTime	The validity time of the overlap after the route has been released. If no value is set, the overlap releases together with the route (for slips). In railML2.4nor the overlap validity time is used for: • Cross locking time for remote controlled stations without simultaneous entry • Reaction time for the local dispatcher for stations without simultaneous entry • Time to confirm stop for ETCS on tracks with stop posts	xs:duration

The release of the route is defined completely in TRV 550-5-4.1.3. To sum up, a route is released completely when:

- The train’s tail (its last axle) has passed the:
  - Exit reference (if no track sections have been defined in the route) or
  - Last track circuit border or train detector which is part of the route (baseline 2; if track sections have been defined in the routes release groups)
- The release trigger has triggered if it has been set

The release trigger is defined by the following attributes:

Attribute of <nor:route>	Description	Type

@releaseTriggerRef	The trigger position used for releasing the complete route (and thus its overlap) while the train still occupies a track section(s) on the route. The same applies to the alternative modelling philosophy where the occupied partial route is kept (from the track section occupied by the train to its exitRef signal), but the overlap is released by the trigger. The placement of the attributes reference depends on the technology: • Through a passage (on remote controlled stations): This is a track circuit border or a train detector. • Through order (on locally controlled stations): This is the local dispatcher observing that the train has come to a stand-still. As the local dispatcher is not modelled, the @releaseTriggerRef is placed on the <crossSection> element. • Through order (on stations with ETCS): This is the train itself reporting to the RBC that the train has come to a stand-still. As the train is not modelled in infrastructure, the @releaseTriggerRef is placed on the <crossSection> element.	rail:tGenericRef
@releaseTriggerHead	This is an additional information for the @releaseTriggerRef attribute placed on a track circuit border or a train detector. It Indicates whether the head of the train (its first axle) triggers the release trigger with the value "true". Or if the tail of the train (its last axle) triggers the release trigger with the value "false".	xs:boolean

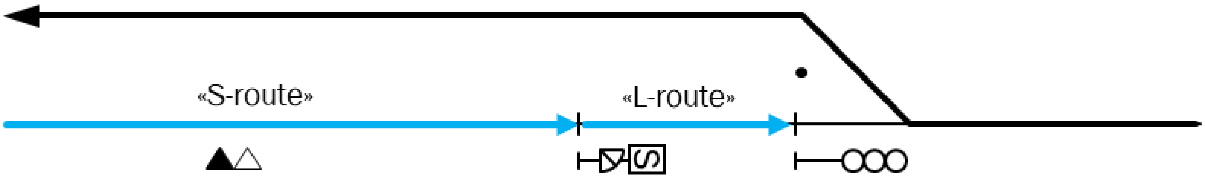
Additionally further generic route properties are defined within in the <controller> element: @nor:routeSetTime and @nor:typicalThrowTime (see chapter 4.5 for further details).

Please note that the reference to the controller takes place indirectly by the @entryRef signal, which has a @controllerRef attribute.

Ending a route at a buffer stop is modelled in railML2.4nor by placing a <signal> element with the attributes @type=”main” and @function="other:end" and referencing it in <nor:route>@exitRef. If no end of route board is present in reality the attribute @virtual is set as "true"

4.8.4 Shortened routes

In Norway shunting signals and end of route boards may shorten a route. Shunting signals are commonly used for stations with common exit signals. End of route boards are also used for stations with alternative simultanious entry design. This is modelled with two adjoining (none overlapping) routes. The first route („S-route“ in illustration) ends at a type="main" function="other:end" signal and the second route („L-route“ in illustration) starts at the same signal. If the end of route board is not present in reality, use virtual="true".



Figur 17: shortened route

4.8.5 Code example

```
<nor:routes>
   <nor:route approachPointRef="id13" approachSpeed="100" code="[BD code]"
description="naming rule:[entry signal name]-[exit signal name]"
entryRef="id11" exitRef="id46" id="id76" name="A101-L201" nor:change="new"
overlapEndRef="id21" overlapValidityTime="PT40S" proceedSpeed="60"
releaseSpeed="20" releaseTriggerHead="false" releaseTriggerRef=”id77”>
       <nor:switchAndPosition switchPosition="straight" switchRef="sw41"/>
   </nor:route>
</nor:routes>
```

4.9 The element <nor:area>

4.9.1 Definition

The <nor:area> Norwegian extension element was introduced in railML2.4nor to describe several different types of areas, e.g. specification of specially controlled areas within the network. It is a direct child of the <infrastructure> element. Special modelling definitions are explained in accordance to the specific type of area.

Track section

The track section is an area, which was created to serve the schematic track plan for signalling and a more detailed description of the route settings (<nor:releaseGroup>). In railML2.4nor the track vehicle detection sections (TVDs) are defined as <nor:area>@type=”trackSection”. The borders of a track section can consist of the following elements:

- <bufferStop>
- <openEnd>
- <trainDetector>
- <trackCircuitBorder>

Work area

A work zone is an area that can be separated from the network for special purposes. When this happens, the assets of the area are not available for normal operation or train traffic. The activation and deactivation are controlled by special routines to ensure the safety of any workers in that zone. A work zone cannot be reopened for traffic without proper action and consent from the outside staff.

A work zone is mainly activated for the protection of staff from train traffic. There is normally no intention for any train movements inside it. The limits of the zone are defined by the end of TVD sections, i.e. axle detection points or insulated joints.

Project area

The project area defines the spatial extent of a <nor:project>, see chapter 4.12. The project area is defined with <nor:area>@type=”project”and references to borders of type “area” . A project area references to its <nor:project> metadata (see chapter 4.13) through the @projectRef attribute under the <state> element (see chapter 4.14.2) that can be placed under <nor:area>.
[note in railML2.5 the <project> references the <genericArea> with <projectArea>@ref].

Local area

The local operation area brings the assets in a special mode where they can be operated freely from on-site devices, e.g. button panel near a point. These assets are not available for any normal operation by the interlocking operator. The activation is done from the interlocking operator giving authorisation for the mode. The return of operational control might be done by commands from the interlocking operator or a device on-site (special deactivation button). Local operation areas are mainly used for shunting purpose without the use of any route. The traffic safety within this area is solely dependent on the on-site staff. The active status of a local operation area is indicated to the railway staff by special signals or special signal aspects. The limits are defined by the end of TVD sections, i.e. axle detection points or insulated joints.

railML2.4nor does not use the railML2.4 core element <locallyControlledArea> as this element requires to segment the tracks beyond the railML2.4nor micro node model definition. Also, at the time of modelling, the semantic of the element was not clear to the Norwegian sector.

Service area
Area where train services are performed. See chapter 4.30.

Overlapping or gapping track section borders
Overlapping or gapping track sections work the same way as regular sequential track sections. It is the reading software that must interpret the data to determine/illustrate if the track sections are sequential, gapping or overlapping. The following figure illustrates the issue.

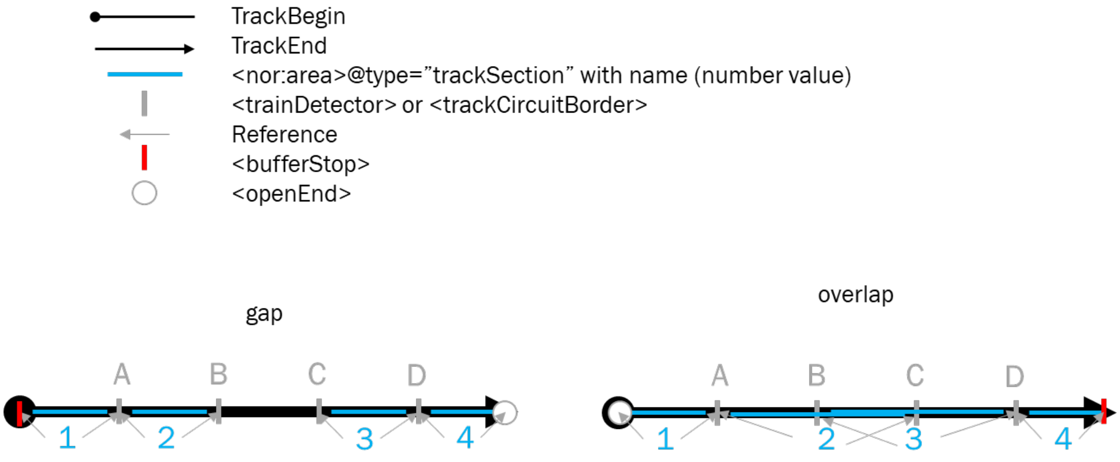

TrackBegin
TrackEnd
<nor:area>@type="trackSection" with name (number value)
<trainDetector> or <trackCircuitBorder>
Reference
<bufferStop>
<openEnd>
gap
A
B
C
D
1
2
3
4
overlap
A
B
C
D
1
2
3
4

Figure 20: References of overlapping and gapping track section borders

4.9.2 Attributes of the element

The <nor:area> element has a selection of the common set of attributes - explained in chapter 3.66 - in addition to the following attributes:

Attribute of <nor:area>	Description	Type
@type	Defines the type of area.	xs:enumeration
@controllerRef	Reference to the controller a track section, work or local area belongs to. This attribute is not relevant for project areas.	rail:tGenericRef

4.9.3 Sub-elements of the element

<nor:isLimitedBy>
In order to reference the border elements <nor:area> has the sub-element <nor:isLimitedBy> with the following attributes:

Attribute of <nor:isLimitedBy>	Description	Type
@ref	References the borders of the area. For track section, local and work areas preferably, interlocking elements shall be referenced.	rail:tGenericRef

<nor:states>
This sub-element was added to railML2.4nor as the core railML2.4 element <state> does not apply to <nor:area>. The <nor:states> element is structured in the same way as the core <states> and its sub-element <state> with the nor-extensions.

4.9.4 Code example

```
<nor:area description="desc" id="id21" name="Example" type="trackSection">
   <nor:isLimitedBy ref="id15" >
   <nor:isLimitedBy ref="id16" >
   <nor:isLimitedBy ref="id71" >
</nor:area>
<nor:area description="desc" id="id21" name="Project A" type="project”>
   <nor:isLimitedBy ref="id1" >
   <nor:isLimitedBy ref="id2" >
      <nor:states>
          <state disabled="false" nor:projectRef="pr1" status="planned"/>
        </nor:states>
</nor:area>
```

4.10 The element <trainDetector>

4.10.1 Definition extension in accordance to Norwegian usage

A <trainDetector> element represents an axle counter and can be referenced as a border by a track section. In this case it is normally referenced by two <nor:area>@type=”trackSection” elements, i.e. two neighboring track sections are using the same axle counter to control the lock and release of the track sections.
Additionally, in the Norwegian railway sector, stations of type “simple entry” (no: “enkelt innkjør”) need to be mapped. These are stations where the local dispatcher is the controller.

4.10.2 Definition extension in accordance to Norwegian usage

If the train detection is handled by the dispatcher at the clearance point, <trainDetector>@medium="other:manual" is used.

4.10.3 Code example

```
<trainDetectionElements>
   <trainDetector absPos="100.0" code="SA-TEL-823091" controllerRef="id5"
description="desc" id="id15" name="" nor:change="modified" pos="100.0"/>
</trainDetectionElements>
```

4.11 The element <trackCircuitBorder>

4.11.1 Definition extension in accordance to Norwegian usage

A <trackCircuitBorder> element represents an insulated rail joint and can be referenced as a border by a track section. In this case it is normally referenced by two <nor:area>@type=”trackSection” elements, i.e. two neighboring track sections are using the same track circuit border to control the lock and release of the two track sections.

4.11.2 Code example

```
<trainDetectionElements>
   <trackCircuitBorder absPos="250.0" code="SA-ATB-856702"
description="desc" id="id71" name="" pos="50.0"/>
</trainDetectionElements>
```

4.12 The element <border>

4.12.1 Definition in accordance of Norwegian usage

The core railML2.4 <border> element is used in railML2.4nor to define the borders of projects without an area.

4.12.2 Attribute extension

The core railML2.4 <border> element is not extended by attributes. However, a new value for the enumeration attribute @type is introduced: “other:norProject”.

4.13 The element <nor:project>

4.13.1 Definition

The <nor:project> element was introduced in railML2.4nor to describe the purpose of the infrastructure objects which are part of a model. One <infrastructure> can have multiple projects (<nor:project>).

The data in <nor:project> must not be confused with the <metadata> data. The latter describes the data source.

Note the link between the project metadata and the project area is made in the <nor:area> of type “project” with the use of the sub element <state> with the railML2.4nor attribute extension @nor:projectRef (see chapter 4.14.2 and 4.9).

4.13.2 Attributes of the element

The element has no further than a selection of the common attributes defined in chapter 3.7. Which exact attributes it has is defined in the Excel sheet.

4.13.3 Sub-elements of the element

In order to fully describe a project, <nor:project> has the following mandatory sub-elements. Please note that railML2.4nor is only a snapshot and therefore only contains one set of these sub-elements. <nor:projectDesignator>, <nor:alternative>, <nor:phase> and <nor:revision>.

<nor:projectDesignator>

This element univocally designates a project. It has a unique project number. The element uses the same attributes and definitions as the core railML element IS:designator - railML 2 Wiki: @register, @entry, @startDate, @endDate.

<nor:alternative>

This sub-element is used to describe alternatives in the planning phase of a project, e.g. tunnel vs. non-tunnel construction. It therefore only applies to infrastructure of status “planned” and “conceptual”. The element has a selection of the common attributes (see chapter 3.7). Which exact attributes it has is defined in the Excel sheet.

Attribute of <nor:alternative>	Value example
@name	"1A"
@description"	"Alternative with elevated main track"

<nor:phase>

This element refers to phases a project can be in. In addition to a selection of the common attributes listed in chapter 3.7, this element has the following attributes:

Attribute of <nor:phase>	Description	Type
@regulationPhase	Refers to phases according to the public planning process. The following predefined values exist in Norway: • "regionalplan" • "kommunedelplan" • "reguleringsplan" • "built"	xs:string

@planningPhase	Refers to phases according to the railway sector or the company/department that is in charge of the internal planning process. The following predefined values exist in Norway: • "KVU" • "utredning" • "hovedplan" • "detaljplan" • "byggeplan"	xs:string
@startDate	Refers to when the phase starts	xs:date
@endDate	Refers to when the phase ends	xs:date

<nor:revision>

This element further defines the revision of a project. In addition to a selection of the common attributes listed in chapter 3.7, this element has the following attributes:

Attribute of <nor:revision>	Description	Type
@nor:initialProjectRef	Refers to the initial project model the revision is based on. This requires the initial project to be written as a separate <nor:project>. No objects under <infrastructure> should reference the initial project (with <state>@nor:projectRef). The project is only mapped to describe/reference the initial project.	rail:tGenericRef

Furthermore, the element <nor:revision> has the following optional sub elements to further define the revision:

- **<nor:objectsRevised>**
  This is a container for all <nor:objectRevised> elements.
- **<nor:objectRevised>**
  This is a subelement of <nor:objectsRevised> and a reference to objects, which are part of the revision. In addition to a selection of the common attributes listed in chapter 3.7, this element has the following attributes:

Attribute of <nor:objecsRevised>	Description	Type
@ref	Reference to the specific objects that have been revised or their more general location (e.g. <nor:lineSection>, <ocp> or <track>)	rail:tGenericRef

- **<nor:revisedBy>**
  This sub-element provides the possibility to provide the revision with information about when it was revised. In addition to a selection of the common attributes listed in chapter 3.6, this element has the following attributes:

Attribute of <nor:revisedBy>	Description	Type
@date	Date of the revision	xs:date

The following two common attributes have a specific definition for the sub-element:

Attribute of <nor:revisedBy>	Description	Type
@name	Name of the person/institution who revised the revision.	xs:string
@description	Optional description provided by reviewer.	xs:string

- <nor:checkedBy>
  This sub-element provides the possibility to provide the revision with information about when it was checked in addition to a selection of the common attributes listed in chapter 3.7, this element has the following attributes:

Attribute of <nor:checkedBy>	Description	Type
@date	Date of the control of the revision	xs:date

The following two common attributes have a specific definition for the sub-element:

Attribute of <nor:checkedBy>	Description	Type
@name	Name of the person/institution who controlled the revision.	xs:string
@description	Optional description provided by controller.	xs:string

- <nor:approvedBy>
  This sub-element provides the possibility to provide the revision with information about when it was approved. In addition to a selection of the common attributes listed in chapter 3.7, this element has the following attributes:

Attribute of <nor:approvedBy>	Description	Type
@date	Date of the approval of the revision	xs:date

The following two common attributes have a specific definition for the sub-element:

Attribute of <nor:approvedBy>	Description	Type
@name	Name of the person/institution who approved the revision.	xs:string
@description	Optional description provided by approver.	xs:string

4.13.4 Code example

```
<nor:projects>
   <nor:project code="myCode" description="myDescr" id="pr1"
name="Testprojekt">
     <nor:projectDesignator entry="PROJ-11" register="PRO" startDate="1999-
12-31" endDate="2001-12-31"/>
     <nor:alternative code="Alt1" description="myAlternativeDescription"
id="alt1" name="Better Alternative"/>
     <nor:phase code="Phase22.1" description="myPhaseDescription" id="pha1"
name="Better Phase" regulationPhase="built" planningPhase="KVU"/>
     <nor:revision code="Rev3.5" description="myRevisionDescription"
id="rev1" name="Better Revision" initialProjectRef="pr2">
          <nor:objectsRevised>
             <nor:objectRevised ref="id4"/>
          </nor:objectsRevised>
          <nor:revisedBy date="2018-12-04" description="ok?" name="John
Doe"/>
          <nor:checkedBy date="2018-12-03" description="ok?" name="Frank"/>
          <nor:checkedBy date="2018-12-01" description="ok?" name="Mary"/>
          <nor:approvedBy date="2018-12-05" description="ok!" name="John
Doe"/>
     </nor:revision>
   </nor:project>
   <nor:project code="myCode2" description="myDescr2" id="pr2"
name="MainProject">
   </nor:project>
</nor:projects>
```

4.14 The element <state>

The element <state> is part of core railML2.4 and is used to define the status of infrastructure pieces with a given time frame for a certain duration. It is a sub-element that can be placed under nearly all infrastructure elements. An overview of all elements that have a <state> sub-element can be found in the Excel sheet.

Note, do not mix up the parent <states> of the core element <state> with the Norwegian extension <nor:states> used under the elements <nor:route> (see chapter 4.8.2) and <nor:area> (see chapter 4.9.3).

4.14.1 Definition in accordance of Norwegian usage

The Norwegian sector decided to extend the element by the attributes listed below to define a start and an end date. The existing @operatingPeriodRef attribute shall only be used in cases of recurring time periods, e.g. a certain element is closed each Tuesday. This is seldom in Norway.

In core railML2.4 the usage of @startDate and @endDate in <operatingPeriod> is restricted by the following:

- *The timeframe must be closed. I.e. both @startDate and @endDate must be used.*
- *A timeframe of the <timetablePeriod> must be set.*
- *The timeframe of <operatingPeriod> must be within the timeframe of the <timetablePeriod>.*
- *The <timetablePeriod> can be as short or as long as required.*

Due to these requirements railML2.4nor requires the usage of the following for:

- Single time period: <state>@nor:startDate and/or <state>@nor:endDate (open or closed time periods)
- Recurring time periods: <state>@operatingPeriodRef with <operatingPeriod>@startDate and <operatingPeriod>@endDate

In addition, the element <state> is used for the description of certain characteristics of track and catenary works in railML2.4nor. For track works use <track>/<states>/<state> or <nor:area>/<nor:states>/<state>. For catenary works use <electrificationChange>/<states>/<state>. For the description of a directional additional running time, the sub-element <nor:additionalRunningTime> is introduced, see section4.13.3.

4.14.2 Attribute usage and extension

The core railML2.4 <state> element does not entirely meet the Norwegian needs. Therefore, the following extensions have been made to the core element:

Attribute of <state>	Description	Type
@nor:startDate	First day the stated status is valid for. If no @startTime is set the status is valid from the beginning of the day.	xs:date
@nor:endDate	Last day the stated status is valid for. If no @endTime is set the status is valid until the end of the day.	xs:date
@nor:projectRef	Reference to the project the object with a <state> belongs to.	rail:tGenericRef

The core railML2.4 the <state>@status attribute has been extended by the value “other:norUnknown”.

To attend to the use case of track and catenary work the attributes of the element <state> under <track>, <nor:area>, and <electrificationChange> are used as described in the following table:

Attribute of <state>	Description	Type
@disabled	true if trains cannot be run, false otherwise	xs:boolean
@status	Values according to core railML. See table below for mapping.	xs:enumeration
@operatingPeriodRef	Reference to an <operatingPeriod> describing the starting days of each restriction period.	rail:tGenericRef
@startTime	The time of day when the restriction starts	xs:time
@endTime	The time of day when the restriction ends	xs:time
@endDayOffset	Number of midnights passed between start time and end time.	xs:integer
@remarks	Free text comment, can be used to describe data source of element	xs:string
@nor:restrictionID	Official reference code for the restriction. Use @status=”operational” or “disabled” in combination with @nor:restrictionID to indicate a restricted or temporarily disabled section [railML2.5: “impairment section”].	xs:string

Mapping of <state>@status values with Banedata values

To ensure a correct mapping between Norwegian Banedata values and defined core railML values we list the mapping here:

Value <state>@status	Banedata value	Banedata description
conceptual	-	-
planned	«planlagt»	infrastructure planned or under construction
operational	«I DRIFT» «Ikke i regulær drift»	operational not in regular operation
disabled	«Midlertidig ute av drift»	temporarily not operational
closed	«nedlagt»	No longer available, removed, dismantled or no longer existing.
other:norUnknown	-	-

4.14.3 Subelement <nor:additionalRunningTime>

The subelement <nor:additionalRunningTime> is introduced to describe the additional running time that results from track and catenary works in a restriction section modelled with the element <state>. The subelement extends the elements <state> under <track>, <nor:area>, or <electrificationChange>.

Attribute of <nor:additionalRunningTime>	Description	Type
@time	Additional running time for trains passing through the restriction area	xs:duration
@absDir	Direction of traffic that the additional time applies to. Possible values are “raising” (misspelling of “rising” inherited from railML2.2) and “falling”.	xs:enumeration

4.14.4 Code example

```
<states>
   <state disabled="false" nor:startDate="2022-03-02" nor:endDate="2022-03-
14" nor:projectRef="pr1" status="operational" remarks="kilde:BaneData"
nor:restrictionID="BN-K107150_209408">
<nor:additionalRunningTime time="P2M30S" absDir="raising"/>
</state>
   <state disabled="false" nor:startDate="2022-03-15" status="operational"
remarks="kilde:IDAM"/>
</states>
```

4.15 The element <gradientChange>

The <gradientChange> element describes a new slope value on a track, whenever this changes in comparison to the previous slope.

4.15.1 Definition extension in accordance to Norwegian usage

In railML2.4nor, the gradient must be defined along the full extent of all tracks. Thus, it is mandatory to place a <gradientChange> at the beginning of a model or at a mileage direction change once for each direction. See also chapter 3.11.

The gradient itself is given in the @slope of <gradientChange>. It signifies the gradient of the slope in ‰ in track direction. Inclines are defined as positive slope values, declines as negative values. Vertical curves are not modelled. Thus, in the model, the slope is extended to a theoretical breakpoint. If it is necessary to distinguish between a high and low breakpoint, the reading software will have to deduct this from the data. The gradient can be microscopic or averaged macroscopic.

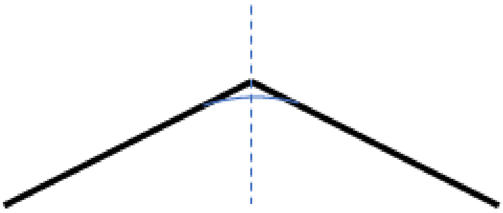


Figure 21: Illustration of the theoretical break point

As stated in chapter 3.9.2, the <gradientChange> - in railML2.4nor - does not use the attribute @dir. The gradients are always valid in both directions, and the gradient change is - as all objects - defined in orientation of the track.

4.15.2 Code example

```
<gradientChanges>
   <gradientChange absPos="0.0" id="id56" pos="0.0" slope="0.0"/>
</gradientChanges>
```

4.16 The element <tunnel>

Tunnels in railML are defined as point elements positioned at a @pos location within a track. Their length is stated by the @length attribute. The correct placement of the element is described in chapter 3.8.

4.16.1 Definition extension in accordance to Norwegian usage

The tunnel air resistance factor plays a crucial role for speeds exceeding 160 km/h. Core railML2.4 currently does not address tunnel resistance factors directly. Instead, it provides physical properties of the tunnel, which - in the conceptual planning stage (<state>@status =”conceptual”) - are not known. railML2.4nor introduces an attribute for a calculated tunnel resistance factor: @nor:resistanceFactorA.

4.16.2 Attribute extension

Attribute of <tunnel>	Description	Type
@nor:resistanceFactorA	Resistance factor for additional aerodynamic resistance in tunnels. Formula for calculation for macroscopic runtime calculation: $F_{addTun} = A * v^2$ F_{addTun} : Additional aerodynamic resistance in tunnels given in Newton. A : Tunnel resistance factor A given in kg/m v^2 : Square of speed given in m/s	xs:double

4.16.3 Code example

```
<tunnels>
   <tunnel absPos="850.0" code="KU-TUN-873405" crossSection="45.0"
description="desc" id="id17" length="100.0" nor:change="modified"
nor:resistanceFactorA="15.0" pos="750.0"/>
</tunnels>
```

4.17 The element <stopPost>

4.17.1 Definition extension in accordance to Norwegian usage

As in railML2.4, the <stopPost> element represents a panel indicating where a train of a certain length should halt to be in an optimal position, e.g. for the exchange of passengers. railML2.4nor does not specify the semantics of the element further compared to core railML2.4. The following attributes are defined more precisely in accordance to Norwegian usage:

Attribute of <stopPost>	Definition in accordance to Norwegian usage
@dir	Direction validity of the stop post board. See chapter 3.9.2.
@trainLength	The <stopPost> is valid for trains with a length equal to or less than the given length. This attribute is mandatory for all passenger trains.

4.17.2 Code example

```
<stopPosts>
  <stopPost absPos="630.0" code="KO-SKT-896301" description="Naming rule:
[text on board]" dir="up" id="id12" name="220 m" nor:change="modified"
pos="530.0" trainLength="220.0" trainRelation="midOfTrain" virtual="false">
<stopPost absPos="630.0" code="KO-SKT-896301" description="Naming rule:
[text on board]" dir="up" id="id12" name="220 m" nor:change="modified"
pos="530.0" trainLength="220.0" trainRelation="midOfTrain" virtual="false">
    <validForMovements kind="passengerTrains"/>
  </stopPost>
</stopPosts>
```

4.18 The element <nor:lineSection>

4.18.1 Definition

In Norway, railway lines - <line> element - are segmented into line sections. They consist of either stations (defined from home to home signal) or open sections (section between stations). Tracks are to be segmented at these borders.

In core railML2.4, stations (OCP with attribute @propOperational="station") reference their tracks with <propEquipment><trackRef>@ref. Open sections are not defined in railML2.4 and do not have a track reference. However, this is important in Norway and thus shall be modelled with the <nor:lineSection> element. It is a new child element of <trackGroups>.

<nor:lineSection> has a sub-element <nor:trackRef> to reference the tracks which belong to the section.

4.18.2 Attributes of the element

The <nor:lineSection> element has the common set of attributes, as described in chapter 3.6. The element does not have additional attributes to these.

The <nor:trackRef> element has the following attributes:

Attribute of <nor:trackRef>	Description	Type
@ref	Reference of tracks	rail:tGenericRef

4.18.3 Code example

```
<trackGroups>
   <nor:lineSection id="ls0" code="8750-85435" description="naming rule:
[designator entry neighbour station down"-"designator entry neighbour
station up"]" name="OSL-NTH">
        <nor:trackRef ref="tr1"/>
        <nor:trackRef ref="tr68"/>
   </nor:lineSection>
</trackGroups>
```

4.19 The element <derailer>

A derailer is a physical safety element, which is part of core railML2.4. In Norway, these are usually secured by a lock (see chapter 4.21). In order to reference the specific lock of a derailer, a new attribute was implemented in railML2.4nor: @nor:lockRef.

Another new attribute is introduced indicating the preferred position of the derailer which it is switched to when not in use: @nor:preferredPosition, in accordance with railML 3.x where this attribute exists.

The type of derailer for the Norwegian railway sector can be indicated indirectly, distinguishing between a manual derailer with no lock, a manual derailer with a lock, and a remotely controlled derailer as followed: A manual derailer with no lock will have no lock reference, thus the attribute @nor:lockRef is not used. On the other side, will both a manual derailer with a lock and a remotely controlled derailer have a lock reference. To distinguish a remotely controlled derailer, the referenced <lock> will have a reference to a controller using @nor:controllerRef. In addition to that, a double derailer can be modelled by two derailers referring to the same lock.

4.19.1 Attribute usage and extension

Attribute of <derailer>	Description	Type
@nor:lockRef	References the lock of a derailer.	rail:tGenericRef
@nor:preferredPosition	Preferred position of the derailer which it is switched to when not in use	xs:enumeration

4.19.2 Code example

```
<derailers>
  <derailer id="dr0" pos=100.0 nor:lockRef="l1"
         nor:preferredPosition=”derailingPosition"/>
</derailers>
```

4.20 The element <trainProtectionChange>

The <trainProtectionChange> element denotes the operational border for the type of train protection which is to be adhered to. This is not the individual train protection element.

In Norway, only balises are available as train protection elements. As these are already mapped as such, the Norwegian sector does not see the need to map them again as train protection elements.

4.20.1 Attribute extension

railML2.4nor extends this element by a new attribute: @nor:trainProtectionType.

Attribute of <trainProtectionChange>	Description	Type
@nor:trainProtectionType	Type of the new train protection system in place after the change point.	Code list: TrainProtectionSystems

4.21 The element <lock>

The <lock> element is a railML2.4core element. However, it is not defined precisely in core railML. Therefore, it is defined in the following.

4.21.1 Definition in accordance to Norwegian usage

Locks are physical safety elements, which use a physical key to lock or unlock objects to change their position. These are usually one or more switches and/or derailers.

The primary use case in railML2.4nor for the modelling of locks is capacity planning. Therefore, only the operational aspects of the lock are modelled and not what kind of object it locks and in which relation it locks them (interlocking).

4.21.2 Attributes of the element

Additionally, to the set of attributes available in core railML2.4 the lock element is extended by the following in railML2.4nor:

Attribute of <lock>	Description	Type
@nor:lockSide	Indicates which side of the track the lock is placed on.	xs:enumeration
@nor:lockType	Type specification of locks deployable in Norway. See Excel table "railML2.4nor Infrastructure Model" for list of values. See operational legislation (TJN) or technical design rules (TRV) for definition of the types.	xs:enumeration
@nor:controllerRef	Reference to the controller that can release the key access to the lock.	rail:tGenericRef
@nor:keyAtRef	Reference to the OCP where the key of the lock is always or predominantly located, when the key is not inserted in the lock.	rail:tGenericRef
@nor:trackDist	Distance from the track center to the lockbox for locking of switches with manual or partially manual locking.	rail:tLengthM

4.21.3 Code example

```
<locks>
   <lock absPos="220.0" code="KO-SPV-803680" nor:controllerRef="id5"
description="Naming rule: [roman numbers]" id="id10" name="II "
nor:controllerRef="id5" nor:keyAtRef="id4" nor:lockSide="left"
nor:lockType="trvA-las" pos="120.0"/>
</locks>
```

4.22 The element <electrificationChange>

4.22.1 Definition extension in accordance to Norwegian usage

In railML2.4nor the @dir attribute of <electrificationChange> is not used.

4.23 The element <levelCrossing>

The <levelCrossing> element is a core railML2.4 element which is extended by two attributes to meet Norwegian needs.

4.23.1 Definition extension in accordance to Norwegian usage

The <levelCrossing> element in railML2.4nor uses the railML2.4 core elements @pos and the @absPos. However, the following specific Norwegian information is stored in them:

Attribute of <levelCrossing>	Description in accordance to Norwegian usage
@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.

In railML2.4nor the <levelCrossing> element has no @dir attribute.

4.23.2 Attributes of the element

The <levelCrossing> element is extended by the following two additional attributes in railML2.4nor:

Attribute of <levelCrossing>	Description	Type

@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.	rail:tLengthM
@nor:roadEndPos	This attribute is used to store the measured end position of the road the railway track is crossing. It is measured along the track, as relative value from <trackBegin>, similar to the attribute @pos.	rail:tLengthM

These two attributes may be used to calculate the length of the level crossing if required by the user system. The core railML2.4 attribute @length is not used for <levelCrossing> in railML2.4nor.

4.23.3 Code example

```
<levelCrossings>
   <levelCrossing absPos="2750.0" code="KO-PLO-809435" controllerRef="id5"
description="naming rule: location name" id="id18" name="Rakkestad"
nor:roadEndPos="1495.0" nor:roadStartPos="1490.0" pos="550.0"/>
</levelCrossings>
```

4.24 The element <infrastructureVisualizations>

railML.org has only defined the structure of this element to be the following in core railML2.4:

```
<infrastructureVisualizations>
   <visualization id="id" infrastructureRef="id of infrastructure">
     <lineVis ref="id of line element">
       <trackVis ref="id of track element">
         <trackElementVis ref="id of track objects">
           <position x="370.0" y="-620.0"/>
          </trackElementVis>
       </trackVis>
      </lineVis>
     <ocpVis ref="id of ocp element>
       <position x="470.0" y="-610.0"/>
     </ocpVis>
   </visualization>
</infrastructureVisualizations>
```

4.24.1 Definition in accordance to Norwegian usage

The usage of the <infrastructureVisualizations> element is optional. If used, the visualization system shall be described in the <dc:source> starting with "infrastructureVisualization:" e.g. producing system, coordinate system, grid.

The <infrastructureVisualizations> element contains the following sub-elements:

Subelement of <infrastructureVisualizations><visualization>	Description
<lineVis>	This element references a <line> and groups the visualized tracks belonging to that line.
<ocpVis>	This element is used to position any object that is not connected to the tracks (today primarily OCP or controller) in the visualization.

Subelement of <lineVis>	Description
<trackVis>	This element references a <track> and groups the visualized track elements belonging to that track.

When using <infrastructureVisualizations> the placement of the <trackBegin>, <trackEnd> and <geoMapping> (for kinks, see chapter 4.25) is mandatory. As track objects are placed on a then visually defined track, it's optional to define the coordinates of the track objects.

The source of the infrastructure visualizations must be clearly defined in the metadata of the railML file it belongs to, via the <dc:source> element. In case there are two infrastructure visualizations it has to be made clear which source belongs to which visualizations via a reference to the id of the visualization. Furthermore, the used coordinate system has to be defined. See code example.

4.24.2 Code example

```
<metadata xmlns:dc="http://purl.org/dc/elements/1.1/">
   <dc:date>2018-11-19T08:10:32.703Z</dc:date>
   <dc:source>railOscope https://railoscope.com</dc:source>
   <dc:source>railML2.3 objects rev71, last changed Tue Nov 13 11:10:03 UTC
2018 by </dc:source>
<dc:source>https://railoscope.com/files/5ac4807594c299059659bba4/views/topo
Editor/11?revision=71</dc:source>
   <dc:source>infrastructureVisualization: For id="vis0" used coordinate
system ortho: 0,0 top-left corner; y-axis vertical from bottom to top, x-
axis horizontal from left to right</dc:source>
   <dc:creator> […] </dc:creator>
</metadata>
```

4.25 The element <geoMapping>

This is a railML2.4 core element that is used to model kinks – logical positions along a track - in a track's schematic visualisation in railML2.4nor.

4.25.1 Definition extension in accordance to Norwegian usage

The <geoMapping> element only contains an @id and a @pos attribute.

4.25.2 Code example

```
<trackElements>
   <geoMappings>
      <geoMapping id="id61" pos="50.0"/>
      <geoMapping id="id62" pos="550.0"/>
   </geoMappings>
</trackElements>
```

4.26 The element <brigde>

This element defines a bridge as an infrastructure element and its attributes. In the Norwegian railway sector, the attribute @kind of <brigde> is used to specify a crossing over or under the railway tracks, as shown in the following table:

String value of <brigde>@kind	Description	Norwegian term
“over”	crossing, something crosses over the railway line	overgangsbro
“under”	railway bridge, something crosses under the railway line	jernbanebro

To further specify the type of crossing, the subelement <crossedElement> can be used, see chapter 4.27.

4.27 The element <crossedElement>

The element <crossedElement> specifies the elements that cross the belonging <brigde> element. To define the type of crossing, the attribute @type is used as shown in the following table. The “other:any” values are based on the corresponding values in railML3.2.

Type (en)	Type (no)	crossedElement@type
national road linking large towns; no motorway	riksveg	“other:primaryRoad”
county road linking towns	fylkesveg	“other:secondaryRoad”
municipal road within towns and in the countryside	kommunal veg	“other:tertiaryRoad”
private road with public access	privat veg med allmenn ferdsel	«other:permissiveRoad»
private road with no public access	privat veg uten allmenn ferdsel	«other:privateRoad»
path, track or sidewalk restricted to pedestrian traffic	gangvei	«highway:footway»

4.28 The element <nor:operatingRules>

Many objects mapped by railML2.4nor have special operating rules. The new trunk element <nor:operatingRules> will group and map those special rules. Only special rules that differ from the generic rule book and apply for specific physical objects are mapped. The generic rule book shall not be mapped here.

As the same rule can apply for multiple objects, we form a list of rules that can be referred to from individual elements (objects).

The usage of the container element <nor:operatingRules> element is optional.

4.28.1 Attributes of the element

The sub element <nor:operatingRule> to the container element <nor:operatingRules>, placed on the <infrastructure> root, contains the standard common attributes except position (see chapter 3.7).

To be able to create groups of rules the element <nor:operatingRule> has an optional sub element <nor:childRule> with the attribute @ruleRef that refer to all operating Rules that shall be contained in this group.

4.28.2 Code example

```
<nor:operatingRules>
   <nor:operatingRule id="id62" code="rule1" name="shunting signal with
required report to local dispatcher" description=" The driver of a vehicle
shall always report to the local dispatcher. Train radio shall be used for
the communication. If the way is free, then permission will be given by
shunting signal using aspect 44 "Cautious shunting allowed" or aspect 45
"Shunting allowed".”/>
   <nor:operatingRule id=”id99” code=”rule2” name="another human readable
rule name” description=" Another rule...”>
   <nor:operatingRule id=”id100” code=”ruleGroup1”>
      <nor:childRule ruleRef=”id62”/>
      <nor:childRule ruleRef=”id99”/>
   </nor:operatingRule>
</nor:operatingRules>
…
<signal id="si52" nor:ruleRef="id62"/>
<signal id="si53" nor:ruleRef="id100"/>
```

4.29 The element <uptime>

Implementation note: To allow the new attributes, the element <uptime> is defined with xsi:type=”nor:uptime” as shown in the example.

To allow the possibility to model an <ocp> attendance requirement of a train in railML2.4nor, the element <uptime> can be used when expanded by two attributes, see chapter 4.29.1.

4.29.1 Attribute extension

railML2.4nor extends this element by two attributes: @nor:operatingPeriodRef, for amongst other to specify cyclical attendance over the week in <operatingPeriod><operatingDay>, and @nor:endDayOffset.

Attribute of <uptime>	Description	Type
@nor:operatingPeriodRef	Reference of the <operatingPeriod> valid for the attendance of the <ocp>.	rail:tGenericRef
@nor:endDayOffset	Duration of the attendance, counted in midnight overruns.	xs:integer

4.29.2 Code example

```
<propOperational operationalType="station" trafficType="passenger">
  <uptime xsi:type="nor:uptime" from="02:00:00" until="04:00:00"
mode="unmanned" operatingPeriodRef="op3"/>
  <uptime xsi:type="nor:uptime" from="04:00:00" until="02:00:00"
endDayOffset="1" mode="manned" operatingPeriodRef="op3"/>
</propOperational>
```

4.30 The element <serviceSection>

The Norwegian railway sector defines services for trains as maintenance of the trains vehicles or loading/unloading facilities for train cargo. These services need to be defined in three different aggregation levels:

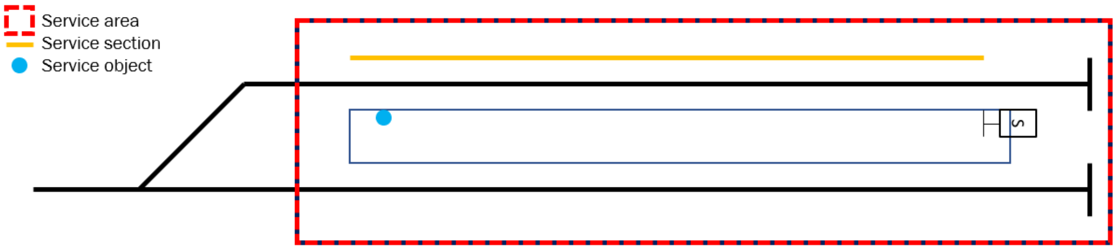
- Service areas
- Service sections
- Service object

Service areas are defined with use of the element <nor:area> and use of <nor:area>@type="service", see also chapter 4.9. Service areas have no requirement for describing attributes, other than name and description. There is no need for a reference between a service area and a service section except the one that is given implicit through the overlapping topology.

For service sections and service objects the existing element <serviceSection> is used as follows. The service section is defined as the full extent of a track segment where the service can be performed on a train. Describing attributes are used according to existing core railML attributes.

The service object is the maintenance object that provides the service. For service objects the @length attribute is not used. The attribute @parentServiceSectionRef must be used for reference towards the service section it is part of. Note that there is a redundancy between the type of service provided by the service section and the service object. To describe the service objects, the use human readable free text of the attribute @description can be used.

Example: See Figure 20 - ServicesFigure 20. Here, the workshop walls are a service area. One of the tracks inside the workshop is a service section, with the extent the longest train might have inside the workshop. A service object could be a preheater post.



4.30.1 Code example

```
<nor:area description="2 tracks for maintenance" id="id21" name="Lodalen
workshop II" type="service">
   <nor:isLimitedBy ref="id15" >
   <nor:isLimitedBy ref="id16" >
   <nor:isLimitedBy ref="id71" >
</nor:area>
…
<serviceSection absPos="200.0" code="803680" description="maintenance track
2" id="id10" name="II " length="220" maintenance="true" preheating="true"
parking="false"/>
<serviceSection absPos="210.0" code="EL-TVP-003680" description="preheating
1000V, 20A, UIC plugg, 10m cable length" id="id11" name="heater I "
parentServiceSection="id10" preheating="true" side="right"/>
```

5 railML2.4nor Examples

The Norwegian sector decided to attach three examples of railML2.4nor to this documentation:

- "Simplest Example"
- "Simple Example Nor"
- "operational RHB example"

The examples are also available for viewing and exporting in NorRailView – the railML2.4nor editor and viewer. In the future further examples will be provided publicly accessible by the Norwegian sector in NorRailView.

To extract the railML2.4nor from the examples in NorRailView, go to the link of the examples and press the export railML button:

5.1 "Simplest Example"

This is a very simple example. It contains only one element of each of the basic types (type of elements roughly corresponding to baseline 1 list) with an exception if multiple elements of the same type were needed to create a consistent example.

https://railoscope.com/tickets/bRbcASDgBifKJ9rM?modelId=5c815c32137e0f14761f0ba8

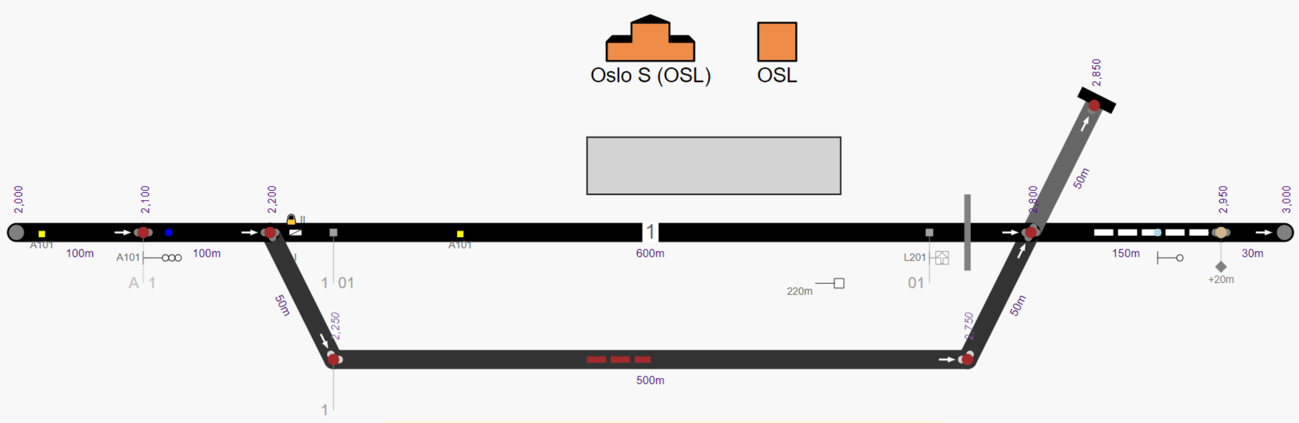


Figure 23 - Track layout of "simplest Example"

5.2 “Norwegian Simple Example”

This example is based on the railML simple example by railML.org. It is extended with all elements of railML2.4nor, except those stated under development above.
https://railoscope.com/tickets/bRbcASDgBifKJ9rM?modelId=5c815c32137e0f14761f0bae

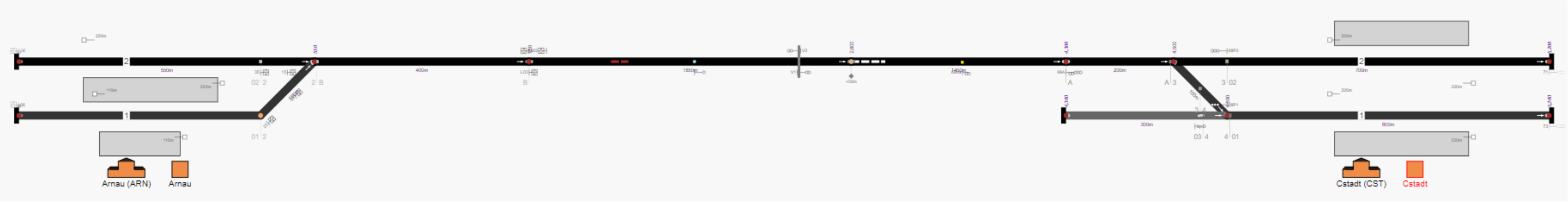


Figure 24 - Track layout of "Simple Example Nor"

5.3 "BB excerpt simple mapping"

This example shows an actual operational line of the Norwegian network mapped on a less granular level with some missing elements.
https://railoscope.com/tickets/bRbcASDgBifKJ9rM?modelId=5d1b75ce72b32b08683bc7c7

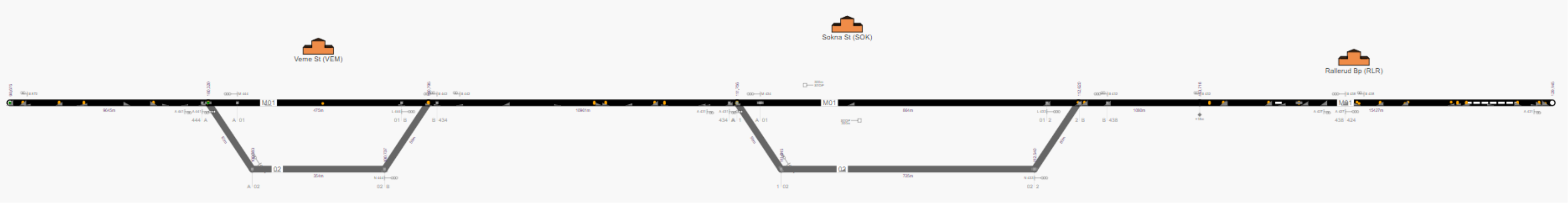


Figure 25 - Track layout of "BB excerpt simple mapping"

5.4 Source code of “Simplest Example”

```
<?xml version="1.0" encoding="UTF-8"?><railml xmlns="https://www.railml.org/schemas/2018"
xmlns:nor="http://www.jernbanedirektoratet.no/railml" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
version="2.4" xsi:schemaLocation="https://www.railml.org/schemas/2018
https://schemas.railml.org/2018/railML-2.4/schema/railML.xsd http://www.jernbanedirektoratet.no/railml
https://www.jernbanedirektoratet.no/globalassets/documenter/railml/norExtension.xsd">
  <metadata xmlns:dc="http://purl.org/dc/elements/1.1/">
    <dc:date>2023-03-29T12:00:52.098Z</dc:date>
    <dc:source>railOscope https://railoscope.com</dc:source>
    <dc:source>Simplest Example rev107, last changed 2023-03-27T11:22:14.228Z[Etc/UTC] by Torben
Brand</dc:source>
    <dc:source>infrastructureVisualization: For id="vis0" used coordinate system ortho: y-axis vertical from
bottom to top, x-axis horizontal from left to right</dc:source>
    <dc:language>no-NO</dc:language>
    <dc:relation>norExtension.xsd_5.4</dc:relation>

<dc:source>https://railoscope.com/workspace/5c0e3000dc7adc0696c960e2?capability=bRbcASDgBifKJ9rM&amp;modelId
=5c815c32137e0f14761f0ba8&amp;revision=107</dc:source>
    <dc:creator>Vivian Augele, Torben Brand, Thomas Nygreen, Janne Moller</dc:creator>
    <organizationalUnits>
      <infrastructureManager code="BN" id="id102" name="Bane NOR"/>
    </organizationalUnits>
  </metadata>
  <infrastructure description="/*This is railML2.4nor Simplest example[431.0,-663.0]*/" id="inf0"
name="Simplest Example">
    <nor:projects>
      <nor:project id="pr641037018230705e93ea77bc" name="&quot;Bruse&quot; brigde">
        <nor:alternative description="Axcle load increase according to concept A2" id="pra8b26d91f-791b-
16f0-b980-da5ff7ee80e8" name="Brigde improvement"/>
        <nor:phase id="prp6473936e-7fc0-996f-e56e-0db99c3020ca" name="As built" regulationPhase="built"
startDate="2020-10-01"/>
      </nor:project>
    </nor:projects>
    <nor:routes>
      <nor:route approachPointRef="id13" approachSpeed="100" code="[BD code?]" conditional="ocpManned"
description="naming rule:[entry signal name]&quot;-&quot;[exit signal name]" entryRef="id11" exitRef="id46"
```

```
id="id76" name="A101-L201" nor:change="new" overlapEndRef="sw27" overlapValidityTime="PT40S"
proceedSpeed="60" releaseSpeed="20" releaseTriggerHead="false" releaseTriggerRef="id16">
        <nor:switchAndPosition switchPosition="straight" switchRef="sw41"/>
        <nor:releaseGroup>
          <nor:trackSectionRef ref="ar87"/>
        </nor:releaseGroup>
        <nor:releaseGroup>
          <nor:trackSectionRef ref="ar50-3"/>
        </nor:releaseGroup>
        <nor:states>
          <state disabled="false" status="conceptual"/>
        </nor:states>
      </nor:route>
    </nor:routes>
    <nor:areas>
      <nor:area id="ar87" name="1" type="trackSection">
        <nor:isLimitedBy ref="id15"/>
        <nor:isLimitedBy ref="id16"/>
        <nor:isLimitedBy ref="id71"/>
      </nor:area>
      <nor:area id="ar50-3" name="01" nor:controllerRef="id5" type="trackSection">
        <nor:isLimitedBy ref="id16"/>
        <nor:isLimitedBy ref="id91"/>
      </nor:area>
      <nor:area id="ar81-3" name="Project new bridge &quot;Bruse&quot;" type="project">
        <nor:isLimitedBy ref="id99"/>
        <nor:isLimitedBy ref="id100"/>
        <nor:states>
          <state disabled="false" nor:projectRef="pr641037018230705e93ea77bc" status="planned"/>
        </nor:states>
      </nor:area>
    </nor:areas>
    <tracks>
      <track code="KO-SPO-800681" description="naming rule: [arabic numbers]" id="tr1" mainDir="none"
name="01" type="mainTrack">
        <states>
          <state disabled="false" status="operational"/>
        </states>
        <trackTopology>
          <trackBegin absPos="2000.0" id="id1" pos="0.0">
            <openEnd id="oe1"/>
          </trackBegin>
          <trackEnd absPos="2100.0" id="id67" pos="100.0">
            <connection id="c67" ref="c68"/>
          </trackEnd>
        </trackTopology>
        <trackElements>
          <speedChanges>
            <speedChange absPos="2000.0" dir="up" etcsTrainCategory="5" id="id54a" pos="0.0"
profileRef="spprf1" vMax="60.0">
              <states>
                <state disabled="false" status="conceptual"/>
              </states>
            </speedChange>
          </speedChanges>
          <gradientChanges>
            <gradientChange absPos="2000.0" id="id56" pos="0.0" slope="0.0">
              <states>
                <state disabled="false" status="conceptual"/>
              </states>
            </gradientChange>
          </gradientChanges>
          <trainProtectionChanges>
            <trainProtectionChange absPos="2000.0" dir="up" id="id86" nor:trainProtectionType="none"
pos="0.0"/>
          </trainProtectionChanges>
          <electrificationChanges>
            <electrificationChange absPos="2000.0" id="id85" pos="0.0" type="none"/>
          </electrificationChanges>
        </trackElements>
        <ocsElements>
          <signals>
            <signal absPos="2100.0" code="SA-SIG-804897" controllerRef="id5" description="naming rule:
[signal letter]+[signal number]" dir="up" function="home" id="id11" name="A101" nor:change="new"
```

```
nor:lamps="3" nor:mounted="pole" nor:side="right" ocpStationRef="id4" pos="100.0" ruleCode="NOR:TJN:§8-10"
sight="50.0" switchable="true" type="main" virtual="false">
              <states>
                <state disabled="false" status="planned"/>
              </states>
            </signal>
          </signals>
          <trainDetectionElements>
            <trainDetector absPos="2100.0" code="SA-TEL-823091" controllerRef="id5" id="id15"
name="Tp(376/378)" nor:change="modified" pos="100.0">
              <states>
                <state disabled="true" status="closed"/>
              </states>
            </trainDetector>
          </trainDetectionElements>
          <balises>
            <balise absPos="2020.0" dir="up" id="id13-P" name="P" pos="20.0">
              <states>
                <state disabled="true" status="operational"/>
              </states>
            </balise>
            <balise absPos="2020.0" dir="up" id="id13-A" name="A" pos="20.0">
              <states>
                <state disabled="true" status="operational"/>
              </states>
            </balise>
            <balise absPos="2020.0" dir="up" id="id13-B" name="B" pos="20.0">
              <states>
                <state disabled="true" status="operational"/>
              </states>
            </balise>
            <balise absPos="2020.0" dir="up" id="id13-C" name="C" pos="20.0">
              <states>
                <state disabled="true" status="operational"/>
              </states>
            </balise>
            <baliseGroup code="SA-ATC-845613" description="Naming rule: [signal name]" id="id13" name="A101"
nor:change="removed" nor:switchable="true" type="other:norFF">
              <baliseRef ref="id13-P" sequence="1"/>
              <baliseRef ref="id13-A" sequence="2"/>
              <baliseRef ref="id13-B" sequence="3"/>
              <baliseRef ref="id13-C" sequence="4"/>
            </baliseGroup>
          </balises>
        </ocsElements>
      </track>
      <track code="KO-SPO-800683" description="naming rule: [arabic numbers]" id="tr26" mainDir="none"
name="1" type="mainTrack">
        <states>
          <state disabled="false" status="operational"/>
        </states>
        <trackTopology>
          <trackBegin absPos="2800.0" id="id26" pos="0.0">
            <connection id="c27-25" ref="c25"/>
          </trackBegin>
          <trackEnd absPos="3000.0" id="id2" pos="180.0">
            <openEnd id="oe2"/>
          </trackEnd>
          <mileageChanges>
            <mileageChange absPos="2970.0" absPosIn="2950.0" code="KO-KJE-800025" id="id65" pos="150.0"
type="missing"/>
          </mileageChanges>
          <connections>
            <crossing absPos="2800.0" code="KO-SPV-803681" controllerRef="id5" description="Naming rule:
[number/number]" id="sw27" length="27.0" name="50/52" nor:change="new" nor:clearance="50.0"
nor:remoteIndicated="true" nor:remoteOperated="true" normalPosition="straight" pos="0.0"
trackContinueCourse="straight" type="doubleSwitchCrossing">
              <states>
                <state disabled="false" status="planned"/>
              </states>
              <connection course="left" id="c27-32" maxSpeed="30.0" orientation="outgoing" passable="true"
ref="c32"/>
```

```
              <connection course="right" id="c27-36" maxSpeed="30.0" orientation="incoming" passable="true"
ref="c36"/>
            </crossing>
          </connections>
        </trackTopology>
        <trackElements>
          <speedChanges>
            <speedChange absPos="2830.0" code="KO-HAS-800373" description="desc" dir="up"
etcsTrainCategory="6" id="id23b" pos="30.0" profileRef="spprf2" vMax="end">
              <states>
                <state disabled="false" status="operational"/>
              </states>
            </speedChange>
            <speedChange absPos="3000.0" dir="down" etcsTrainCategory="5" id="id55a" pos="180.0"
profileRef="spprf1" vMax="100.0">
              <states>
                <state disabled="false" status="conceptual"/>
              </states>
            </speedChange>
          </speedChanges>
          <gradientChanges>
            <gradientChange absPos="2850.0" description="desc" id="id20" pos="50.0" slope="15.0">
              <states>
                <state disabled="false" status="operational"/>
              </states>
            </gradientChange>
          </gradientChanges>
          <tunnels>
            <tunnel absPos="2850.0" code="KU-TUN-873405" crossSection="45.0" description="desc" id="id17"
length="100.0" name="Oslotunnelen" nor:change="modified" nor:resistanceFactorA="15.0" pos="50.0">
              <states>
                <state disabled="false" status="operational"/>
              </states>
            </tunnel>
          </tunnels>
          <trainProtectionChanges>
```

```
            <trainProtectionChange absPos="2900.0" description="desc" dir="up" id="id21"
nor:trainProtectionType="F-ATC" pos="100.0">
              <states>
                <state disabled="false" status="operational"/>
              </states>
            </trainProtectionChange>
            <trainProtectionChange absPos="3000.0" dir="down" id="id95" nor:trainProtectionType="ETCS-L2"
pos="180.0"/>
          </trainProtectionChanges>
        </trackElements>
        <ocsElements>
          <signals>
            <signal absPos="2900.0" code="KO-SKT-800934" description="Naming rule: [just value where
applicable, no name]" dir="up" id="id14" nor:mounted="pole" pos="100.0" ruleCode="NOR:TJN:60F"
switchable="false">
              <states>
                <state disabled="true" status="disabled"/>
              </states>
            </signal>
          </signals>
        </ocsElements>
      </track>
      <track code="KO-SPO-800685" description="naming rule: [arabic numbers]" id="tr32" mainDir="none"
name="3" type="sidingTrack">
        <states>
          <state disabled="false" status="planned"/>
        </states>
        <trackTopology>
          <trackBegin absPos="2800.0" id="id32" pos="0.0">
            <connection id="c32" ref="c27-32"/>
          </trackBegin>
          <trackEnd absPos="2850.0" id="id33" pos="50.0">
            <bufferStop code="KO-SST-800256" description="Naming rule: [track name]" id="id73" name="1"/>
          </trackEnd>
        </trackTopology>
        <trackElements>
          <speedChanges>
```

```
            <speedChange absPos="2850.0" dir="down" etcsTrainCategory="5" id="id74a" pos="50.0"
profileRef="spprf1" vMax="40.0"/>
          </speedChanges>
          <radiusChanges>
            <radiusChange absPos="2835.0" code="KO-HOT-874132" description="rad1" id="id38" pos="35.0"
radius="1200.0">
              <states>
                <state disabled="false" status="planned"/>
              </states>
            </radiusChange>
          </radiusChanges>
          <trainProtectionChanges>
            <trainProtectionChange absPos="2850.0" dir="down" id="id96" nor:trainProtectionType="ETCS-L2"
pos="50.0"/>
          </trainProtectionChanges>
        </trackElements>
      </track>
      <track code="KO-SPO-800683" description="naming rule: [arabic numbers]" id="tr40" mainDir="none"
name="1" type="mainTrack">
        <states>
          <state disabled="false" status="operational"/>
        </states>
        <trackTopology>
          <trackBegin absPos="2200.0" id="id40" pos="0.0">
            <connection id="c41-39" ref="c39"/>
          </trackBegin>
          <trackEnd absPos="2800.0" id="id25" pos="600.0">
            <connection id="c25" ref="c27-25"/>
          </trackEnd>
          <connections>
            <switch absPos="2200.0" code="KO-SPV-803680" controllerRef="id5" description="Naming rule:
[Arabic number]" id="sw41" length="43.0" name="1" nor:change="new" nor:clearance="50.0" nor:lockRef="id10"
nor:remoteIndicated="true" nor:remoteOperated="true" normalPosition="straight" pos="0.0"
trackContinueCourse="straight">
              <states>
                <state disabled="false" status="operational"/>
              </states>
              <connection course="right" id="c41-43" maxSpeed="50.0" orientation="outgoing" passable="true"
ref="c43"/>
            </switch>
          </connections>
          <crossSections>
            <crossSection absPos="2500.0" code="8750-85435" description="Naming rule: [track name]" id="id6"
ocpRef="id4" pos="300.0">
              <states>
                <state disabled="false" status="operational"/>
              </states>
            </crossSection>
          </crossSections>
        </trackTopology>
        <trackElements>
          <speedChanges>
            <speedChange absPos="2650.0" code="KO-HAS-800372" description="desc" dir="up"
etcsTrainCategory="5" id="id19a" pos="450.0" profileRef="spprf1" vMax="100.0">
              <states>
                <state disabled="false" status="operational"/>
              </states>
            </speedChange>
            <speedChange absPos="2650.0" code="KO-HAS-800372" description="desc" dir="up"
etcsTrainCategory="6" id="id19b" pos="450.0" profileRef="spprf2" vMax="105.0">
              <states>
                <state disabled="false" status="operational"/>
              </states>
            </speedChange>
          </speedChanges>
          <levelCrossings>
            <levelCrossing absPos="2750.0" code="KO-PLO-809435" controllerRef="id5" description="naming
rule: location name" id="id18" length="20.0" name="Rakkestad" nor:roadEndPos="560.0"
nor:roadStartPos="540.0" pos="550.0" protection="Ba">
              <states>
                <state disabled="true" status="closed"/>
              </states>
            </levelCrossing>
          </levelCrossings>
          <platformEdges>
            <platformEdge absPos="2450.0" code="KO-PLF-807354" description="desc" height="760.0" id="id7"
length="200.0" name="Test" nor:change="modified" ocpRef="id4" pos="250.0" side="left">
              <states>
                <state disabled="false" status="operational"/>
              </states>
            </platformEdge>
          </platformEdges>
        </trackElements>
        <ocsElements>
          <signals>
            <signal absPos="2720.0" code="SA-SIG-804898" controllerRef="id5" description="naming rule:
[signal letter]+[signal number]" dir="up" function="exit" id="id46" name="L201" nor:change="modified"
ocpStationRef="id4" pos="520.0" ruleCode="NOR:TJN:E35b" switchable="false" type="main" virtual="true">
              <states>
                <state disabled="false" status="planned"/>
              </states>
              <etcs level_2="true"/>
            </signal>
          </signals>
          <trainDetectionElements>
            <trackCircuitBorder absPos="2250.0" code="SA-ATB-856701" controllerRef="id5" description="desc"
id="id16" nor:change="new" pos="50.0">
              <states>
                <state disabled="false" status="planned"/>
              </states>
            </trackCircuitBorder>
            <trackCircuitBorder absPos="2720.0" id="id91" pos="520.0"/>
          </trainDetectionElements>
          <balises>
            <balise absPos="2350.0" dir="up" id="id24-dummy" pos="150.0">
              <states>
                <state disabled="false" status="conceptual"/>
              </states>
            </balise>
            <baliseGroup code="SA-ATC-845612" description="Naming Rule: [signal name]" id="id24" name="A101"
type="infill">
```

```
              <baliseRef ref="id24-dummy" sequence="1"/>
            </baliseGroup>
          </balises>
          <stopPosts>
            <stopPost absPos="2630.0" code="KO-SKT-896301" description="Naming rule: [no name]" dir="up"
id="id12" nor:change="removed" ocpRef="id4" pos="430.0" trainLength="220.0" trainRelation="headOfTrain">
              <states>
                <state disabled="true" status="planned"/>
              </states>
              <validForMovements kind="passengerTrains"/>
            </stopPost>
            <stopPost absPos="2720.0" dir="up" id="id98" ocpRef="id4" pos="520.0" trainLength="470.0"
trainRelation="headOfTrain" virtual="true">
              <validForMovements kind="allTrains"/>
            </stopPost>
          </stopPosts>
          <derailers>
            <derailer absPos="2220.0" code="SA-SSP-871032" derailSide="left" description="Naming rule:
[roman numbers]" dir="up" id="id9" name="I" nor:change="new" nor:lockRef="id10" pos="20.0">
              <states>
                <state disabled="false" status="planned"/>
              </states>
            </derailer>
          </derailers>
          <locks>
            <lock absPos="2220.0" code="KO-SPV-803680" description="Naming rule: [roman numbers]" id="id10"
name="II " nor:controllerRef="id5" nor:keyAtRef="id4" nor:lockSide="left" nor:lockType="trvA-las"
pos="20.0">
              <states>
                <state disabled="false" status="other:norUnknown"/>
              </states>
            </lock>
          </locks>
        </ocsElements>
      </track>
      <track code="KO-SPO-800684" description="naming rule: [arabic numbers]" id="tr43" mainDir="none"
name="2" type="secondaryTrack">
```

```
        <trackTopology>
          <trackBegin absPos="2200.0" id="id43" pos="0.0">
            <connection id="c43" ref="c41-43"/>
          </trackBegin>
          <trackEnd absPos="2800.0" id="id36" pos="600.0">
            <connection id="c36" ref="c27-36"/>
          </trackEnd>
          <borders>
            <border absPos="2425.0" id="id99" pos="225.0" type="area"/>
            <border absPos="2525.0" id="id100" pos="325.0" type="area"/>
          </borders>
        </trackTopology>
        <trackElements>
          <bridges>
            <brigde absPos="2450.0" id="id89" kind="under" length="50.0" name="Bruse" pos="250.0">
              <states>
                <state disabled="false" nor:startDate="2023-01-24" startTime="15:06:00"
status="operational"/>
              </states>
            </brigde>
          </bridges>
          <geoMappings>
            <geoMapping id="id93" pos="100.0"/>
            <geoMapping id="id94" pos="500.0"/>
          </geoMappings>
        </trackElements>
        <ocsElements>
          <trainDetectionElements>
            <trackCircuitBorder absPos="2250.0" code="SA-ATB-856702" id="id71" pos="50.0"/>
          </trainDetectionElements>
        </ocsElements>
      </track>
      <track code="KO-SPO-800683" description="naming rule: [arabic numbers]" id="tr68" mainDir="none"
name="1" type="mainTrack">
        <states>
          <state disabled="false" status="conceptual"/>
        </states>
        <trackTopology>
          <trackBegin absPos="2100.0" id="id68" pos="0.0">
            <connection id="c68" ref="c67"/>
          </trackBegin>
          <trackEnd absPos="2200.0" id="id39" pos="100.0">
            <connection id="c39" ref="c41-39"/>
          </trackEnd>
        </trackTopology>
        <trackElements>
          <electrificationChanges>
            <electrificationChange absPos="2120.0" code="[code from BD]" description="[no name]" id="id22"
pos="20.0" type="overhead">
              <states>
                <state disabled="false" status="operational"/>
              </states>
            </electrificationChange>
          </electrificationChanges>
        </trackElements>
      </track>
    </tracks>
    <trackGroups>
      <line id="li0" name="Bergensbanen">
        <trackRef ref="tr1"/>
        <trackRef ref="tr26"/>
        <trackRef ref="tr32"/>
        <trackRef ref="tr40"/>
        <trackRef ref="tr43"/>
        <trackRef ref="tr68"/>
      </line>
      <nor:lineSection id="ls0" name="OSL-">
        <nor:trackRef ref="tr1"/>
      </nor:lineSection>
    </trackGroups>
    <operationControlPoints>
      <ocp code="8750-85435" description="desc" id="id4" name="Oslo S" nor:change="modified">
        <nor:propOperationalAdditional remoteControlled="true" simultaneousEntry="partial"/>
        <propOperational operationalType="station" trafficType="passenger"/>
        <propService bus="true" passenger="true"/>
        <propEquipment>
          <trackRef ref="tr26"/>
          <trackRef ref="tr32"/>
          <trackRef ref="tr68"/>
          <trackRef ref="tr40"/>
          <trackRef ref="tr43"/>
        </propEquipment>
        <propOther>
          <states>
            <state disabled="false" status="other:norUnknown"/>
          </states>
        </propOther>
        <geoCoord coord="52.0 15.0"/>
        <designator entry="OSL" register="SJN"/>
      </ocp>
    </operationControlPoints>
    <controllers>
      <controller code="SA-SIK-800400" description="desc" id="id5" model="NSB-77" name="OSL"
nor:routeSetTime="PT5S" nor:typicalThrowTime="PT10S" technologyType="electrical">
        <states>
          <state disabled="false" status="planned"/>
        </states>
        <ocpRef ref="id4"/>
      </controller>
    </controllers>
    <speedProfiles>
      <speedProfile id="spprf1" influence="increasing" name="basis"/>
      <speedProfile id="spprf2" influence="increasing" name="nor:pluss"/>
      <speedProfile id="spprf3" influence="increasing" name="nor:krenge"/>
      <speedProfile id="spprf4" influence="decreasing" name="nor:local" verbalConstraint="SJN defined
condition"/>
      <speedProfile id="spprf5" influence="decreasing" name="nor:temporary"/>
      <speedProfile id="spprf6" influence="decreasing" name="nor:conditional" verbalConstraint="SJN defined
condition"/>
      <speedProfile id="spprf7" influence="decreasing" name="nor:avalanche" verbalConstraint="SJN defined
condition"/>
```

```
    </speedProfiles>
  </infrastructure>
  <infrastructureVisualizations>
    <visualization id="vis0" infrastructureRef="inf0" version="2.4">
      <lineVis ref="li0">
        <trackVis ref="tr1">
          <trackElementVis ref="id1">
            <position x="370.0" y="-620.0"/>
          </trackElementVis>
          <trackElementVis ref="oe1">
            <position x="370.0" y="-620.0"/>
          </trackElementVis>
…
        </trackVis>
      </lineVis>
      <ocpVis ref="id4">
        <position x="470.0" y="-590.0"/>
      </ocpVis>
      <ocpVis ref="id5">
        <position x="490.0" y="-590.0"/>
      </ocpVis>
    </visualization>
  </infrastructureVisualizations>
</railml>
```

6 References

railML.org. (2018, 09 12). Retrieved from The railML.org initiative: https://www.railml.org/en/introduction/background.html