railML® Use Case Definition

Interlocking Module Engineering Data

Abbreviation	RSIM
Implementation	railML® 3.1 (release: 19.02.2019)
Date	17.02.2019
Version	1.0
Author	Jörg von Lingen

Revision History

Version	Date	Description	Author
0.1	10.11.2018	Initial Version	Jörg von Lingen
0.2	10.12.2018	Revision according schema update and comments	Jörg von Lingen
0.3	11.12.2018	Renaming "point" to "switch"	Jörg von Lingen
0.4	27.01.2019	Revision after schema refactoring	Jörg von Lingen
1.0	17.02.2019	Prepared for release	Jörg von Lingen

Executive Summary

The railML 3.1 provides an XML schema for detailed description of infrastructure and interlocking features. The interlocking part is based on the use cases “Routes for timetable simulation” (RSIM) and “Interlocking module engineering data” (IMED). Whereas the infrastructure part focuses on the physical objects and features the interlocking part is considering the functional or logical features of the objects. Thus the interlocking is using the infrastructure with references to their objects. The interlocking part allows the description of typical objects like:

- TVD sections
- movable elements like switches, derailers and crossings
- signals with their aspects and aspect relations
- level crossings
- train routes with start area, running path and overlap
- route release
- special operational areas and element groups
- interfaces between different interlocking
- control relations from HMI to interlocking module and the element

Although some infrastructure manager dependent characteristics can be defined the interlocking sub-schema does not directly describe operational rules of that infrastructure manager.

1 Informal Use Case Description

Infrastructure Managers simulate traffic on increasingly realistic models of their networks to design and test timetables. Alternative route-setting is essential to model behaviour in case of disruptions. Simulation includes realistic interlocking routesetting behaviour. Simulation models include time-to-set and time-to-release parameters, restrictions due to flank protection, number of blocks reserved by trains due to the nature of the signalling system, alternative routes, mutually exclusive routes, etc.

Consider for instance routesetting. An interlocking is required to set and lock routes. A description of routes in terms of railML is of interest to both engineering and simulation. However, a simulation program will focus on the time it takes for a route to be set and released. This information is generally not of interest when engineering an interlocking for a particular site. Furthermore, timing, e.g. the time before a switch is thrown and locked depends on the underlying hardware and software. In other words, this information depends on specific hardware which suggest that it has no place in a RailML model which by vocation is portable and system-independent. One would suggest that simulation users add specific extensions to suit their needs, e.g. add information about throwing delay to specific types of switches. Simulation of an interlocking by nature replicates the behaviour of the interlocking. In other words, the interlocking model would require the functions. The Euro-Interlocking project (UIC, 2013) makes a stab at functional modelling. Several IM's need interlocking data for modelling capacity (e.g. (Jernbaneverket, 2015)). Questions about the network and the signalling system need answering. How many extra trains can be squeezed in, given a legacy interlocking? What happens if we remove or add a signal? How many trains are affected by the loss of a neuralgic point?

Most capacity operational simulation tools use their own internal generic interlocking (table) model. Thus this use case does not require a interlocking table. This distinguishes this IL UC from other IL UC. A route description with attributes (primarily timings and speed) or route attributes on signal infrastructure objects is sufficient.

railML originated from the need to test timetables against realistic simulations. The railway network had to enter the model in terms of distances, gradients, curves, etc. The interlocking and signalling system adds time constraints to the model. Routes are slow to clear or set, switches are slow to throw and so on. The timetable must resist perturbations by providing alternative routes. Capturing routes and timing parameters in railML will improve precision of the simulation and so produce more robust timetables.

Alternatively, Infrastructure Managers may be interested in improving the performance of signalling and interlocking. What happens if a signal is removed ? What if we release routes more quickly ? What happens if we introduce ETCS L3 ? Specialised designers of simulation software designed to answer such questions would find it highly helpful when uniform railML data are available to answer such what-if questions.

1.1 Data Flows and Interfaces

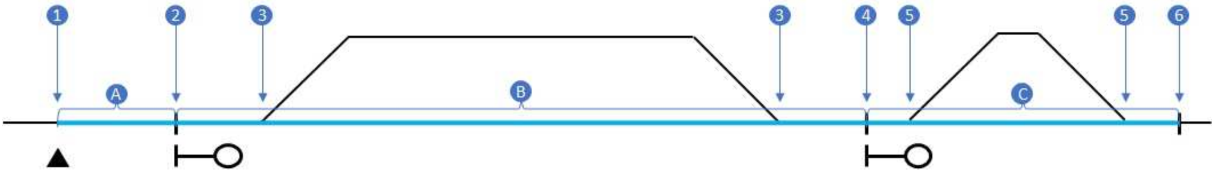
A simulation program typically requires interlocking data to control train operation.

Block Sections, Routes (OpenTrack uses the term Route for the Block Sections):

- Start and end location of the block section (start signal, end signal)
- Unique definition (sorted sequence of elements, switches, …)
- Reservation point of the block section (optional)
- Release point
- Partial release groups
- Reservation time
- Release time
- Switching time (of switches) (optional)
- Signal aspect at the start signal (which signal aspect is provided when the block section is reserved)
- Overlaps (optional)
- Slow speed zone for restrictive signal aspects (e.g 40 km/h, from where to where is the speed restriction valid [e.g. from the first signal, from the first switch] and where does it end)
- Additional signal aspects (e.g. to allow an entry into an occupied block section)

Itineraries:

- Sequence of block sections, e.g. to define the complete journey of a train (optional)



Route elements for use in the use case:

1. approachPoint
2. routeEntry
3. switchPosition, TVDs grouped in release groups
4. routeExit, overlap, overlapRelease (timer)
5. overlap switchPosition
6. overlap limitedBy

- A: approachSpeed
- B: proceedSpeed, releaseSpeed
- C: overlapValidityTime
- movableElement - max/average throwTime
- route - releaseTime, reserveTime

1.2 Interference with other railML schemas

Interlocking data rely heavily on infrastructure data. In fact, the interlocking data mostly are pointers to objects defined in the infrastructure section. This implies that any modification of the IS model is likely to affect the interlocking model.

Additionally to infrastructure the rolling stock schema is required for running trains on the defined infrastructure with interlocking. This for amongst others technical minimum headway calculations in the simulation and timetable free capacity assessments.

Additionally to the infrastructure and rolling stock schemas the timetable schema is required for running multiple trains interactively in a defined timetable concept for timetable feasibility and robustness studies, amongst others through UIC406 method.

1.3 Data Characterization

This section serves to specify the required data regarding certain aspects.

How often do the data change (update)?

This use case is likely to occur in an iterative design process. For example, an IM requests external simulation specialists to optimise throughput of a busy yard. The IM exchanges the yard's railML data with the simulation specialists. The latter shift signals and switches to find the optimal signalling and interlocking design. Then they return the railML data to the IM who may adapt according to their rules and regulations. The frequency of data exchange in such a design-Simulation cyclus can be high. Alternatively, railML data may be fixed. An IM would provide read-only railML interlocking data such that interested third parties may provide suggestions for improvement. One can imagine a open-data scenario where university students access railML data for modelling assignments (and come up with wonderful suggestions for improvement).

How big are the data fragments to be exchanged (complexity)?

Data fragments are likely to be range from yards to corridors, depending on the subnet under investigation.

Which views are represented by the data (focus)?

The data provide a more functional view with respect to their use in interlocking context, especially for train routes.

- Signalling
- Schematic view of topology (Indispensable for interlocking engineers used to graphic presentation of signalling plans)

Which specific data do you expect to receive/send (elements)?

- Infrastructure:
    - Topology: Lines + operational points
    - Signalling: signals, train detection
    - Rails: track sections, switches, crossings, derailer, level crossings
    - Operational Points: name, type
- Interlocking:
    - IM specific Types: signal aspects, route types
    - Track Assets: TVD sections, movable elements, signals
    - Logical Assets: routes, overlaps, partial routes, route relations, conflicting routes

  - Signal Plan: aspect relations, signalled speeds
  - Timers: specific timers for route setting and release and operation of track assets

2 Schema Implementation

2.1 Sub-schema References

The implementation of the RSIM use case is based on elements of the following sub-schemas:

- Infrastructure
- Interlocking

2.2 Element Classification

This section serves to list all the railML® elements that are required for the implementation of the RSIM use case. For all the related sub-schemas the listing is done using tables in the following structure:

{Topic}		
{mandatory element/@attribute}	m	{Description}
{optional element/@attribute}	o	{Description}

Mandatory elements have to be provided by an export interface and have to be understood by an import interface. However, this does not mean mandatory elements or attributes have to be occur at each possible position. Such elements or attributes shall occur at least once in the import or export, if not set to mandatory in the schema. Optional elements are recommended for export and import interfaces.

2.2.1 Common

There are no railML® common elements required for the RSIM use case.

2.2.2 Infrastructure

The infrastructure used for the RSIM use case is comparable in its extent with the SCTP use case [2]. It shall contain all the related infrastructure elements as listed below with their physical characteristics. This means the topology of the network and element positions are crucial. However, the details of the infrastructure are not included here.

The following tables contain the railML® infrastructure elements required for the RSIM use case:

Infrastructure		
topology	m	The topology of the described railway network has to be given.
positioning	m	
geometry	x	
functionalInfrastructure	m	The functional infrastructure objects with their features have to be given.

2.2.2.1 Topology

Topology		
netElement	m	
netRelation	m	
network	m	The <network> shall contain at least one child element <level> for describing the microscopic railway network on line section level and line level.

2.2.2.2 Positioning

Positioning		
geometricPositioningSystem	o	
linearPositioningSystem	m	There shall be at least one <linearPositioningSystem> for defining the mileage along the railway line.
screenPositioningSystem	x	

GeometricPositioningSystem		
@crsDefinition	o	The geometric positioning system shall be identified by its EPSG code to be found in this attribute.

LinearPositioningSystem		
anchor	m	All mileage changes and other mileage anomalies shall be modelled using <anchor> elements.
@units	m	Mileage shall be given in meters.
@startMeasure, @endMeasure	m	Mileage shall be given in meters with one digit after the comma.

2.2.2.3 Functional Infrastructure

Functional Infrastructure		
balise	o	
border	m	Borders between infrastructure manager responsibilities shall be modelled as well as national borders
bufferStop	m	
crossing	m	
derailer	m	
electrification	x	
embankment	x	
levelCrossing	m	It is recommended to provide elementary information about existence of level crossings along the line
line	o	
loadingGauge	x	
lock	m	
operationalPoint	m	
operationFacility	x	
overCrossing	x	
platform	x	
platformEdge	x	
restrictionArea	x	

serviceSection	x	
signal	m	
speed	m	
stoppingPlace	o	
switch	m	
track	m	
trackBed	x	
trackGauge	x	
trainDetectionElement	m	
trainProtectionElement	o	
trainRadio	x	
undercrossing	x	
weightLimit	x	

2.2.3 Interlocking

The following tables contain the railML® interlocking elements required for the RSIM use case:

Interlocking		
specificIM	m	The specific types of the infrastructure manager of the described railway network have to be given. There may be more than one entry of specificIM.
assetsForIL	m	The list of assets used for interlocking purpose is essential for this use case. There is only one list of assets.
signalBox	m	The main characteristics of the particular interlocking module with all related assets have to be given. There may be more than one entry of signalBox.
controller	m	The controller definition is in most cases not essential for route data. However, a particular controller is needed for providing the list of train itineraries. There may be more than one entry of controller.

2.2.3.1 specificIM

specificIM		
ownsSetsOfAssets @ref	m	This reference to the list of assets is required for consistency check.
usesTypes	m	This parent element for the list of IM specific types is required.
designator @entry, @register	m	The name of the selected Infrastructure Manager is required in @entry for readability of the data set. The @register shall contain the name of the related register from where the name was taken. In case of project specific names only the register shall name the project with a leading underscore.
@id	m	The ID of the Infrastructure Manager will be used for references to it.

specificIM | usesTypes

hasAspect | designator @entry, @register	m	Each defined signal aspect shall have a name (from the IM).
has Aspect @id	m	Each defined signal aspect shall have an ID to refer to it.
hasAspect @GenericAspect	m	The attribute provides a basic classification of the defined signal aspect. It uses an enumeration without the possibility to add “other”-entries.
hasTVDresetStrategy | designator @entry, @register	x	
hasTVDresetStrategy @id	x	
hasTVDresetStrategy @ResetStrategy	x	
hasRouteType | designator @entry, @register	m	Each defined route type shall have a name (from the IM).
hasRouteType @id	m	Each defined route type shall have an ID to refer to it.
hasRouteType @GenericRouteType	m	The attribute provides a basic classification of the defined route type. It uses an enumeration with the possibility to add “other”-entries.
hasLevelCrossingType | designator @entry, @register	o	Each defined level crossing type shall have a name (from the IM).
hasLevelCrossingType @id	o	Each defined level crossing type shall have an ID to refer to it.
hasLevelCrossingType @isControlType	o	The attribute provides a basic classification of the defined level crossing type. It uses an enumeration without the possibility to add “other”-entries.
hasLevelCrossingType @allowsLocalOperation	x	
hasLevelCrossingType @hasBarrier	x	
hasLevelCrossingType @hasTrafficWarning	x	
hasElementGroupType | designator @entry, @register	x	
hasElementGroupType @id	x	
hasElementGroupType @elementGroupType	x	
hasDetectorType | designator @entry, @register	x	
hasDetectorType @id	x	
hasDetectorType @detectorType	x	

2.2.3.2 assetsForIL

assetsForIL		
tvdSection | designator @entry, @register	m	The list of TVD section is required as they shall be known to the interlocking. Each defined TVD section shall have a name (from the IM).
tvdSection @id	m	Each defined TVD section shall have an ID to refer to it.
switchIL | designator @entry, @register	m	The list of switches is required as they shall be known to the interlocking. Each defined switch shall have a name (from the IM).

switchIL @id	m	Each defined switch shall have an ID to refer to it.
derailerIL | designator @entry, @register	m	The list of derailers is required as they shall be known to the interlocking. Each defined derailer shall have a name (from the IM).
derailerIL @id	m	Each defined derailer shall have an ID to refer to it.
movableCrossing | designator @entry, @register	m	Each defined movable crossing shall have a name (from the IM).
movableCrossing @id	m	Each defined movable crossing shall have an ID to refer to it.
levelCrossingIL | designator @entry, @register	o	The list of level crossings is required as they shall be known to the interlocking. Each defined level crossing shall have a name (from the IM).
levelCrossingIL @id	o	Each defined level crossing shall have an ID to refer to it.
key | designator @entry, @register	x	
key @id	x	
keyLockIL | designator @entry, @register	x	
keyLockIL @id	x	
genericDetector | designator @entry, @register	x	
genericDetector @id	x	
signalIL | designator @entry, @register	m	The list of signals is required as they shall be known to the interlocking. This applies only to switchable or virtual signals. Each defined signal shall have a name (from the IM) and an ID to refer to it.
signalIL @id		
atpDevice | designator @entry, @register	x	not used in railML v3.1
atpDevice @id	x	not used in railML v3.1
interface | designator @entry, @register	x	
interface @id	x	
workZone | designator @entry, @register	x	
workZone @id	x	
localOperationArea | designator @entry, @register	x	
localOperationArea @id	x	
shuntingZone | designator @entry, @register	x	
shuntingZone @id	x	
permissionZone | designator @entry, @register	x	
permissionZone @id	x	

routeReleaseGroupAhead | designator @entry, @register	o	The list of route release groups ahead of the train is optional as they shall be known to the interlocking, if it supports the release of several elements in a group from set routes. Each defined route release group ahead of the train shall have a name (from the IM).
routeReleaseGroupAhead @id	o	Each defined route release group ahead of the train shall have an ID to refer to it.
routeReleaseGroupRear | designator @entry, @register	o	The list of route release groups in rear of the train is optional as they shall be known to the interlocking, if it supports the release of several elements in a group from set routes. Each defined route release group in rear of the train shall have a name (from the IM).
routeReleaseGroupRear @id	o	Each defined route release group in rear of the train shall have an ID to refer to it.
route | designator @entry, @register	m	The list of routes is required as they shall be known to the interlocking. Each defined route shall have a name (from the IM).
route @id	m	Each defined route shall have an ID to refer to it.
conflictingRoute | designator @entry, @register	o	The list of conflicting routes is optional as they shall be known to the interlocking, if some routes need to be set incompatible explicitly. Each defined conflicting route shall have a name (from the IM).
conflictingRoute @id	o	Each defined conflicting route shall have an ID to refer to it.
routeRelation | designator @entry, @register	o	The list of routes relation is optional as they shall be known to the interlocking, if there additional conditions to be configured for a defined route. Each defined route relation shall have a name (from the IM).
routeRelation @id	o	Each defined route relation shall have an ID to refer to it.
combinedRoute | designator @entry, @register	x	
combinedRoute @id	x	
overlap | designator @entry, @register	m	The list of overlaps is required as they shall be known to the interlocking. Each defined overlap shall have a name (from the IM).
overlap @id	m	Each defined overlap shall have an ID to refer to it.
dangerPoint | designator @entry, @register	x	
dangerPoint @id	x	
destinationPoint | designator @entry, @register	x	
destinationPoint @id	x	
powerSupplyIL | designator @entry, @register	o	The list of power supplies for interlocking is required as they shall be known to the interlocking. Each defined power supply shall have a name (from the IM).
powerSupplyIL @id	o	Each defined power supply shall have an ID to refer to it.

designator @entry, @register	o	The name of the asset list supports readability of data set.
@id	m	The ID is required as it is referred to it from the IM specific data definition.

assetsForIL | tvdSection		
hasDemarcatingBufferstop @ref	o	The reference to a buffer stop in the infrastructure part is optional as the railway network might not have any.
hasExitSignal @ref	o	The reference to a signal is optional as it might not affect the interlocking data.
hasDemarcatingTraindetector @ref	m	The reference to train detectors in the infrastructure part is required as there are no TVD sections without them.
hasResetStrategy @ref	x	
@isBerthingTrack	o	The Boolean value may be used to indicate any special station tracks where trains usually start, end, split, join or reverse.
@residualRouteCancellationDelay	m	The Boolean value is mandatory in the schema.
@partialRouteReleaseDelay	m	The Boolean value is mandatory in the schema.
@technology	x	
@frequency	x	
designator @entry, @register	m	The TVD section shall have a name (from the IM).
@id	m	The ID is required as it is used to refer to it.

assetsForIL | switchIL		
refersTo @ref	m	The reference to an element in the infrastructure part is mandatory in the schema.
hasGaugeClearanceMarker @ref	x	
hasTvdSection @ref	x	
connectedToPowerSupply @ref	o	The reference to the related power supply may be used to control the number of switching switch actuators.
relatedMovableElement @ref	o	The reference is only used if there is a related movable element. Its use is dependent on value of @switchPointType.
hasFoulingTrainDetectors @ref	m	The references to train detection elements in infrastructure part mark the branches of the switch, which may receive lack of clearance.
branchLeft @ref	m	The reference to the underlying track in infrastructure part allows to determine the neighbour relation. It is mandatory in schema.
branchRight @ref	m	The reference to the underlying track in infrastructure part allows to determine the neighbour relation. It is mandatory in schema.
hasPositionRestriction | designator @entry, @register	x	
hasPositionRestriction @id		
hasPositionRestriction | relatedSwitchInPosition	o	Dependent on the relation to another switch there might be a restriction in positions. It shall be only used either relatedSwitchInPosition or relatedDerailerInPosition.

hasPositionRestriction | relatedDerailerInPosition	o	Dependent on the relation to another derailer there might be a restriction in positions. It shall be only used either relatedSwitchInPosition or relatedDerailerInPosition.
hasPositionRestriction | relatedElementInPosition @restrictedPosition	o	Dependent on the relation to another movable element there might be a restriction in positions.
@maxThrowTime	m	The duration value is mandatory in the schema. It is needed to supervise the switching process.
@typicalThrowTime	o	The duration value is optional and normally not needed for the interlocking.
@returnsToPreferredPosition	m	The Boolean value is needed for handling after use in a route.
@isKeyLocked	m	The Boolean value is needed to determine whether the switch is normally clamped by a locking device. If true there shall be no switch actuators connected.
@preferredPosition	o	In case the movable element has a preferred position (when not in use) this position shall be given. It uses an enumeration without the possibility to add “other”-entries. If there is no preferred position the attribute shall be omitted.
@numberOfBladeSwitchActuators	o	The non-negative integer value shall be given for switches. It shall be “0” in case of no switch actuator.
@numberOfFrogSwitchActuators	o	The non-negative integer value shall be given for switches. It shall be “0” in case of no switch actuator.
designator @entry, @register	m	The switch shall have a name (from the IM).
@id	m	The ID is required as it is used to refer to it.

assetsForIL | switchIL | relatedSwitchInPosition		
refersToSwitch @ref	m	The reference to a switch in the asset list is mandatory in the schema.
@inPosition	m	The position of the related switch
designator @entry, @register	x	
@id	o	The marking of the element with @id is optional. The ID is never referred to.

assetsForIL | switchIL | relatedDerailerInPosition		
refersToDerailer @ref	m	The reference to an element in the infrastructure part is mandatory in the schema.
@inPosition	m	The position of the related movable element
designator @entry, @register	x	
@id	o	The marking of the element with @id is optional. The ID is never referred to.

assetsForIL | derailerIL		
refersTo @ref	m	The reference to an element in the infrastructure part is mandatory in the schema.
hasGaugeClearanceMarker @ref	x	
hasTvdSection @ref	o	The reference to a TVD section allows the relation to the occupational status of the derailer.

connectedToPowerSupply @ref	o	The reference to the related power supply may be used to control the number of switching switch actuators.
relatedMovableElement @ref	x	
@maxThrowTime	m	The duration value is mandatory in the schema. It is needed to supervise the switching process.
@typicalThrowTime	o	The duration value is optional and normally not needed for the interlocking.
@returnsToPreferredPosition	o	The Boolean value is optional for handling after use in a route.
@isKeyLocked	o	The Boolean value is optional to determine whether the derailer is normally clamped by a locking device. If true there shall be no switch actuators connected.
@preferredPosition	o	In case the movable element has a preferred position (when not in use) this position shall be given. It uses an enumeration without the possibility to add "other"-entries. If there is no preferred position the attribute shall be omitted.
@numberOfBladeSwitchActuators	o	The non-negative integer value shall be given for derailers. It shall be "0" in case of no switch actuator.
@numberOfFrogSwitchActuators	x	
designator @entry, @register	m	The derailer shall have a name (from the IM).
@id	m	The ID is required as it is used to refer to it.

assetsForIL | movableCrossing		
refersTo @ref	m	The reference to an element in the infrastructure part is mandatory in the schema.
hasGaugeClearanceMarker @ref	x	
hasTvdSection @ref	x	
connectedToPowerSupply @ref	o	The reference to the related power supply may be used to control the number of switching switch actuators.
relatedMovableElement @ref	x	
branchUpLeft @ref	m	The reference to the underlying track in infrastructure part allows to determine the neighbour relation. It is mandatory in schema.
branchUpRight @ref	m	The reference to the underlying track in infrastructure part allows to determine the neighbour relation. It is mandatory in schema.
branchDownLeft @ref	m	The reference to the underlying track in infrastructure part allows to determine the neighbour relation. It is mandatory in schema.
branchDownRight @ref	m	The reference to the underlying track in infrastructure part allows to determine the neighbour relation. It is mandatory in schema.
hasFoulingTrainDetectors @ref	o	The references to train detection elements in infrastructure part mark the branches of the switch, which may receive lack of clearance.
@maxThrowTime	m	The duration value is mandatory in the schema. It is needed to supervise the switching process.

@typicalThrowTime	o	The duration value is optional and normally not needed for the interlocking.
@returnsToPreferredPosition	x	
@isKeyLocked	x	
@preferredPosition	o	In case the movable element has a preferred position (when not in use) this position shall be given. It uses an enumeration without the possibility to add “other”-entries. If there is no preferred position the attribute shall be omitted.
@numberOfBladeSwitchActuators	x	
@numberOfFrogSwitchActuators	o	The non-negative integer value may be given for crossings, if there is any switch actuator connected.
designator @entry, @register	m	The crossing shall have a name (from the IM).
@id	m	The ID is required as it is used to refer to it.

assetsForIL | levelCrossingIL		
isLevelCrossingType @ref	m	The reference to the basic level crossing type from IM specific data definition is required.
refersTo @ref	m	The reference to an element in the infrastructure part is mandatory in the schema.
hasInterface @ref	x	
deactivatedBy	m	The definition of deactivation is required as it is mandatory in the schema.
activationCondition	m	The definition of activation is required as it is mandatory in the schema.
hasTvdSection @ref	x	
@preferredPosition	o	In case the level crossing has a preferred position (when not in use) this position shall be given. It uses an enumeration without the possibility to add “other”-entries. If there is no preferred position the attribute shall be omitted.
@unprotectedSpeed	o	The speed value in km/h in case of unprotected level crossing.
@typicalTimeToClose	m	The duration value is required to control the switching process. It is mandatory in the schema.
@constantWarningTime	m	The duration value is required to set up the delay between activation and closing when road traffic is warned. It is mandatory in the schema.
@minimumOpenTime	x	The duration value is required to define the time span a reclosing command shall be rejected after opening. If unknown the attribute can be omitted.
@maximumClosedTime	x	The duration value is required to define the time after a closed level crossing is considered problematic. If unknown the attribute can be omitted.
@requiresStopBeforeUnprotected LevelCrossing	o	The Boolean value may be needed to consider train behavior in case of unprotected level crossing.
designator @entry, @register	m	The level crossing shall have a name (from the IM).
@id	m	The ID is required as it is used to refer to it.

assetsForIL | levelCrossingIL | deactivatedBy		
tvdDetectorRef @ref	m	The reference to the train detection device or TVD section is required for the location.
@delay	m	The duration value is required to set up the delay between the trigger and the action. It is mandatory in the schema.
designator @entry, @register	x	
@id	o	The marking of the element with @id is optional. The ID is never referred to in this use case.

assetsForIL | levelCrossingIL | activationCondition		
delayBySwitch | designator @entry, @register	x	
delayBySwitch @id	x	
delayBySwitch @delay	o	If the element is used the value shall be given.
delayBySwitch @inPosition	o	If the element is used the value shall be given. It uses an enumeration without the possibility to add "other"-entries.
delayBySwitch | refersToSwitch @ref	o	If the element is used the reference shall be given.
aspectRelatedDelay | designator @entry, @register	x	
aspectRelatedDelay @id	x	
aspectRelatedDelay @delay	o	If the element is used the value shall be given.
aspectRelatedDelay | refersToSignal @ref	o	If the element is used the reference shall be given.
aspectRelatedDelay | showsAspect @ref	o	If the element is used the reference shall be given.
hasSignalDelayTime | designator @entry, @register	x	
hasSignalDelayTime qid	x	
hasSignalDelayTime @delay	o	If the element is used the value shall be given.
hasSignalDelayTime | delayedBySignal @ref	o	If the element is used the reference shall be given.
activatedBy | designator @entry, @register	x	
activatedBy @id	x	
activatedBy @delay	m	The duration value is required to set up the delay between the trigger and the action. It is mandatory in the schema.
activatedBy | refersTo @ref	m	The reference to the train detection device or TVD section is required for the location.
@andOr	o	The attribute shall be used if more than one activation condition is given here. It uses an enumeration without the possibility to add "other"-entries.
designator @entry, @register	x	
@id	o	The marking of the element with @id is optional. The ID is never referred to in this use case.

assetsForIL | signalIL		
refersTo @ref	m	The reference to an element in the infrastructure part is mandatory in the schema.
protectsBlockExit @ref	x	

@releaseSpeed	o	The speed value may be given but refers to the IM specific operational rules and is not directly used in an interlocking.
@malfunctionSpeed	o	The speed value may be given but refers to the IM specific operational rules and is not directly used in an interlocking.
@approachSpeed	m	The speed value shall be given but refers to the IM specific operational rules and is not directly used in an interlocking.
@passingSpeed	m	The speed value shall be given but refers to the IM specific operational rules and is not directly used in an interlocking.
@releaseDelay	m	The duration value is required to set up the delay for signal closure after the trigger event.
@function	x	
@isVirtual	m	The Boolean value is required to mark whether the signal is connected to an interlocking for switching.
designator @entry, @register	m	The signal shall have a name (from the IM).
@id	m	The ID is required as it is used to refer to it.

assetsForIL | routeReleaseGroupAhead		
@isAutomatic	o	The Boolean value may be given to indicate the automatic release.
@delay	o	The duration value used for delay release after use may be given.
@typicalDelay	m	The duration value used to indicate the typical time needed for release after use shall be given.
hasTvdSection @ref	m	The list of references to the TVD sections contained in this partial route shall be given.
designator @entry, @register	o	The partial route may have a name (from the IM) to enhance readability.
@id	m	The ID is required as it is used to refer to it.

assetsForIL | routeReleaseGroupRear		
@delay	o	The duration value used for delay release after use may be given.
@typicalDelay	m	The duration value used to indicate the typical time needed for release after use shall be given.
hasTvdSection @ref	m	The list of references to the TVD sections contained in this partial route shall be given.
designator @entry, @register	o	The partial route may have a name (from the IM) to enhance readability.
@id	m	The ID is required as it is used to refer to it.

assetsForIL | route		
handlesRouteType @ref	m	The reference to the IM specific route type is required.
routeActivationSection	o	The route activation condition may be given.
facingSwitchInPosition …	m	The distinct definition of route path by relevant switch positions is required. It may be used for any other movable element type if really needed to define the path.
routeEntry	m	The definition of the start of the route is mandatory in the schema.

hasTvdSection @ref	m	The reference to any TVD section within the running path of the route shall be given. This shall be preferable an ordered list starting from route entry and it shall conform to the path defined by the switch positions.
hasReleaseGroup @ref	o	The list of references to partial routes may be given if needed for route release.
switchPositionInDepartureTrack …	o	The defined position of any movable element in the departure track (in rear of route entry) may be needed.
routeExit	m	The definition of the destination of the route is mandatory in the schema.
additionalRelation @ref	o	The reference to route relations which shall be considered in addition may be listed.
@locksAutomatically	x	
@processingDelay	m	The duration value shall be given if the route locking is delayed after the trigger event for setting it.
@proceedAspectDelay	o	The duration value may be given if there is a delay for opening the start signal after the route is locked.
@signalClosureDelay	x	
@approachReleaseDelay	x	
designator @entry, @register	m	The route shall have a name (from the IM).
@id	m	The ID is required as it is used to refer to it.

assetsForIL | route | routeActivationSection		
@delayForLock	m	The duration value shall be given to allow control of route activation.
@automaticReleaseDelay	o	The duration value may be given to allow control of release of activated route.
activationSection @ref	m	The reference to the TVD section used as trigger for route setting/locking is required.
designator @entry, @register	x	
@id	o	The marking of the element with @id is optional. The ID is never referred to in this use case.

assetsForIL | route | facingSwitchInPosition		
refersToSwitch @ref	m	The reference to the switch in the asset list is mandatory in the schema.
@inPosition	m	The required position of the switch is mandatory in the schema. It uses an enumeration without the possibility to add "other"-entries.
designator @entry, @register	x	
@id	o	The marking of the element with @id is optional. The ID is never referred to in this use case.

assetsForIL | route | routeEntry		
refersTo @ref	m	The reference to the track asset marking the start of the route is mandatory in the schema. This is typical the reference to a signal of the interlocking assets.
nonReplacement @ref	x	
designator @entry, @register	x	
@id	o	The marking of the element with @id is optional. The ID is

		never referred to in this use case.

assetsForIL | route | routeExit		
refersTo @ref	m	The reference to the track asset marking the destination of the route is mandatory in the schema. This is typical the reference to a signal of the interlocking assets.
hasDangerpoint @ref	x	
hasOverlap @ref	m	The reference to one or more related overlaps shall be given.
designator @entry, @register	x	
@id	o	The marking of the element with @id is optional. The ID is never referred to in this use case.

assetsForIL | route | switchPositionInDepartureTrack		
refersToSwitch @ref	m	The reference to the switch in the asset list is mandatory in the schema.
@inPosition	m	The required position of the switch is mandatory in the schema:
designator @entry, @register	x	
@id	o	The marking of the element with @id is optional. The ID is never referred to in this use case.

assetsForIL | conflictingRoutes		
refersTo @ref	m	The reference to the route having a conflict shall be given.
conflictsWithRoute @ref	m	The reference to a route causing the conflict shall be given.
reasonForConflict @origin	m	The reasons for the conflict shall be given. It uses an enumeration with the possibility to add "other"-entries.
reasonForConflict @refersTo	m	The reference to the element causing the conflict shall be given.
designator @entry, @register	o	The conflicting route pairing may have a name (from the IM) to enhance readability.
@id	m	The ID is required as it is used to refer to it.

assetsForIL | routeRelations		
requiredSwitchPosition …	o	instance of type AssetAndGivenState
requiredDerailerPosition …	o	instance of type AssetAndGivenState
requiredCrossingPosition …	o	instance of type AssetAndGivenState
requiredDetectorState …	o	instance of type AssetAndGivenState
requiredSignalAspect …	o	instance of type AssetAndGivenState
requiredSectionState …	o	instance of type AssetAndGivenState
requiredKeyLockState …	o	instance of type AssetAndGivenState
requiredLevelCrossingState …	o	instance of type AssetAndGivenState
… {AssetAndGivenState} | designator @entry, @register	x	
… {AssetAndGivenState} @id	o	The marking of the element with @id is optional. The ID is never referred to in this use case.
… {AssetAndGivenState} | refersTo… @ref	m	The reference to a related track asset shall be given.
… {AssetAndGivenState} | …InPosition/State	m	The position or state of the related track asset shall be given.

@mustOrShould	m	The level of state enforcement is required. It uses an enumeration without the possibility to add "other"-entries.
@proving	m	The way of state proving is required. It uses an enumeration without the possibility to add "other"-entries.
@isNegated	o	The Boolean value may be set in case the position or state shall be any other but the listed one. It is only useful if there are more than two positions or states available for the element.
designator @entry, @register	o	The route relation may have a name (from the IM) to enhance readability.
@id	m	The ID is required as it is used to refer to it.

assetsForIL | combinedRoute		
comboEntry @ref	m	The reference to the entry point of the route combination is required.
comboExit @ref	m	The reference to the exit point of the route combination is required.
containsRoute @ref	m	The list of references to the single routes in the combination is required. The list shall be preferable in the order of the routes.
designator @entry, @register	m	The combined route shall have a name (from the IM).
@id	m	The ID is required as it is used to refer to it.

assetsForIL | overlap		
activeForApproachRoute @ref	m	The list of references to the related routes of the overlap shall be given. This is the indirect reverse link to the route exit elements.
relatedToTrackAsset @ref	o	The reference to the related track asset of the overlap may be given. This is the link to a signal or so used as route exit element.
requiresSwitchInPosition …	m	The movable elements in overlap required to be in a particular position shall be given for definition of the overlap path.
hasTvdSection @ref	m	The reference to any TVD section within the path of the overlap shall be given. This shall be preferable an ordered list starting from route exit and it shall conform to the path defined by the switch positions.
isLimitedBy @ref	m	The reference to the limiting train detection element shall be given.
overlapRelease	m	The condition for overlap release is required.
overlapRelease | releaseTriggerSection @ref	m	The reference to the TVD section triggering the overlap release is required.
overlapRelease | overlapReleaseTimer	m	The timer for overlap release is required.
overlapRelease | overlapReleaseTimer @timerValue	m	The duration value is required to set up the release timer.
overlapRelease | overlapReleaseTimer @overlapReleaseCondition	m	The detail of the trigger condition is required. It uses an enumeration without the possibility to add "other"-entries.
@overlapSpeed	o	The speed value may be given.
@releaseSpeed	o	The speed value in km/h may be given.

@overlapValidityTime	o	The duration value may be given.
@length	o	The length value of the overlap in metres may be given. It shall be coordinated with the limiting train detection element.
designator @entry, @register	o	The overlap may have a name (from the IM) to enhance readability.
@id	m	The ID is required as it is used to refer to it.

assetsForIL | powerSupplyIL		
@numberOfSimultaneousSwitching Actuators	m	The non-negative integer value is required to control the switch actuator activation.
@signalVoltageMode	x	
designator @entry, @register	o	The power supply may have a name (from the IM) to enhance readability.
@id	m	The ID is required as it is used to refer to it.

2.2.3.3 signalBox

signalBox		
controlsSystemAsset	x	
controlsTrackAsset	m	The list of track assets known to the interlocking is required.
controlsRoute @ref	m	The list of references to the routes controlled by the interlocking is required.
controlsCombinedRoute @ref	x	
controlsInterface	x	
controlledBy @ref	x	
implementsSignalplan	m	The signal plan for the interlocking is required.
implementsElementGroup	x	
hasPermissionZone @ref	x	
hasConflictingRoutes @ref	o	The list of references to the route conflicts known to the interlocking may be given.
hasConfiguration	x	
designator @entry, @register	o	The signal box (interlocking) shall have a name (from the IM).
@id	o	The marking of the element with @id is optional. The ID is never referred to in this use case.

signalBox | controlsTrackAsset		
connectedTrackAsset @ref	m	The reference to the track asset from the list is required.
@extentOfControl	m	The control level is required. It uses an enumeration without the possibility to add "other"-entries.
designator @entry, @register	o	The track asset relation may have a name (from the IM) to enhance readability.
@id	o	The marking of the element with @id is optional. The ID is never referred to in this use case.

signalBox | implementsSignalplan		

aspectRelation | designator @entry, @register	m	The signal plan shall have a name (from the IM) to enhance readability.
aspectRelation @id	m	The ID is required as it is used to refer to it.

signalBox | implementsSignalplan | aspectRelation		
masterAspect …	o	The reference to the signal and the aspect shown at the route exit may be given. It is required if this aspect is to be announced at the route entry by the slave aspect. Instance of type SignalAndAspect
slaveAspect …	m	The reference to the signal and the aspect shown at the route entry is required. Instance of type SignalAndAspect
distantAspect …	o	The reference to the signal and the aspect shown at the distant signal related to the signal of the masterAspect. Instance of type SignalAndAspect
… {SignalAndAspect} | designator @entry, @register	x	
… {SignalAndAspect} @id	o	The ID of the aspect relation may be given.
… {SignalAndAspect} | refersToSignal @ref	m	The reference to the particular signal in the asset list shall be given.
… {SignalAndAspect} | showsAspect @ref	m	The reference to the aspect shown at the signal shall be given.
signalsSpeedProfile @ref	o	The reference to an applicable speed profile in infrastructure may be given.
appliesToRoute @ref	m	The reference to the applicable route is required.
@passingSpeed	m	The speed value related to the slave aspect is required.
@expectingSpeed	o	The speed value related to the master aspect may be given.
@endSectionTime	o	The duration value may be given for controlling route release.
designator @entry, @register	o	The aspect relation may have a name (from the IM) to enhance readability.
@id	m	The ID is required as it is used to refer to it.

2.2.3.4 Controller

controller		
controlledAssets	o	The container for controlled signalboxes and system assets may be given.
itineraries	m	The container for itineraries is required.
designator @entry, @register	m	The controller shall have a name (from the IM).
@id	m	The ID is required as it is used to refer to it.

controller | controlledAssets		
controlledInterlocking | connectedSignalBox @ref	o	The reference to the signal box controlled from this place may be given.
controlledInterlocking @extentOfControl	o	The extent of control over this signal box may be given.
controlledInterlocking | designator @entry, @register	o	The control relation may have a name (from the IM) to enhance readability.

controlledInterlocking @id	o	The marking of the element with @id is optional. The ID is never referred to in this use case.
controlledSystemAsset | connectedSystemAsset @ref	x	
controlledSystemAsset @extentOfControl	x	
controlledSystemAsset | designator @entry, @register	x	
controlledSystemAsset @id	x	

controller | itineraries		
itinerary | comboEntry @ref	m	The reference to the entry point of the route combination is required.
itinerary | comboExit @ref	m	The reference to the exit point of the route combination is required.
itinerary | containsRoute @ref	m	The list of references to the single routes in the combination is required. The list shall be preferable in the order of the routes.
itinerary | designator @entry, @register	o	The combined route may have a name (from the IM) to enhance readability.
itinerary @id	o	The marking of the element with @id is optional. The ID is never referred to in this use case.

2.2.4 Timetable

No elements are required. Timetable part is not yet available in railML3.

2.2.5 Rollingstock

No elements are required. Rollingstock part is not yet available in railML3.

3 Additional Conditions

None.

4 References

[1] railML.org Wiki: *Use case Interlocking Module Engineering Data*. In: https://wiki.railml.org/index.php?title=IL:UC:InterlockingEngineering; last access: 14.08.2018

[2] railML.org Wiki: *Use case Schematic Track Plan.* In: https://wiki.railml.org/index.php?title=UC:IS:Schematic_Track_Plan; last access: 18.06.2018