

RCA



Reference CCS Architecture

*An initiative facilitated by the ERTMS Users Group
and the EULYNX consortium*

A. Interface Specification SCI_1

This is a snapshot version. Please use it with caution.

There is ongoing work. The content of this document may be unfinished, will likely contain errors and can be changed without prior notice.

	Document Number and Issue	RCA.Doc.25, Public Snapshot (v0.0.7)
	Date of Publish	06-12-2019
	Page No	2

REVISION HISTORY

Version	Date	Superseded documents/description/details	Change Request No
0.0.7	06-12-2019	Initial version of component specification	n/a

	Document Number and Issue	RCA.Doc.25, Public Snapshot (v0.0.7)
	Date of Publish	06-12-2019
	Page No	3

TABLE OF CONTENTS

1. Concept (Phase 1).....	6
2. System Definition (Phase 2)	7
3. Risk Analysis and Evaluation (Phase 3).....	8
4. System Requirements (Phase 4)	9
4.1. Class Diagram: SCI_1 ExecutionState.....	9
4.2. Class Diagram: SCI_1 IO Items	11
4.3. Class Diagram: SCI_1 OperationalPlan	13
4.4. Class Diagram: SCI_1 OperationalPlanEventLink.....	18

	Document Number and Issue	RCA.Doc.25, Public Snapshot (v0.0.7)
	Date of Publish	06-12-2019
	Page No	4

TABLE OF FIGURES

Figure 1 Class Diagram: SCI_1 ExecutionState	9
Figure 2 Class Diagram: SCI_1 IO Items.....	11
Figure 3 Class Diagram: SCI_1 OperationalPlan.....	13
Figure 4 Class Diagram: SCI_1 OperationalPlanEventLink	18

	Document Number and Issue	RCA.Doc.25, Public Snapshot (v0.0.7)
	Date of Publish	06-12-2019
	Page No	5

LIST OF TABLES

Table 1 Description of Classes.....	11
Table 2 Description of Classes.....	12
Table 3 Description of Classes.....	17
Table 4 Description of Classes.....	20

	Document Number and Issue	RCA.Doc.25, Public Snapshot (v0.0.7)
	Date of Publish	06-12-2019
	Page No	6

1. CONCEPT (PHASE 1)

Cenelec Phase 1 is not covered in this document

	Document Number and Issue	RCA.Doc.25, Public Snapshot (v0.0.7)
	Date of Publish	06-12-2019
	Page No	7

2. SYSTEM DEFINITION (PHASE 2)

Cenelec Phase 2 is not covered in this document

	Document Number and Issue	RCA.Doc.25, Public Snapshot (v0.0.7)
	Date of Publish	06-12-2019
	Page No	8

3. RISK ANALYSIS AND EVALUATION (PHASE 3)

Cenelec Phase 3 is not covered in this document

	Document Number and Issue	RCA.Doc.25, Public Snapshot (v0.0.7)
	Date of Publish	06-12-2019
	Page No	9

4. SYSTEM REQUIREMENTS (PHASE 4)

4.1. Class Diagram: SCI_1 ExecutionState

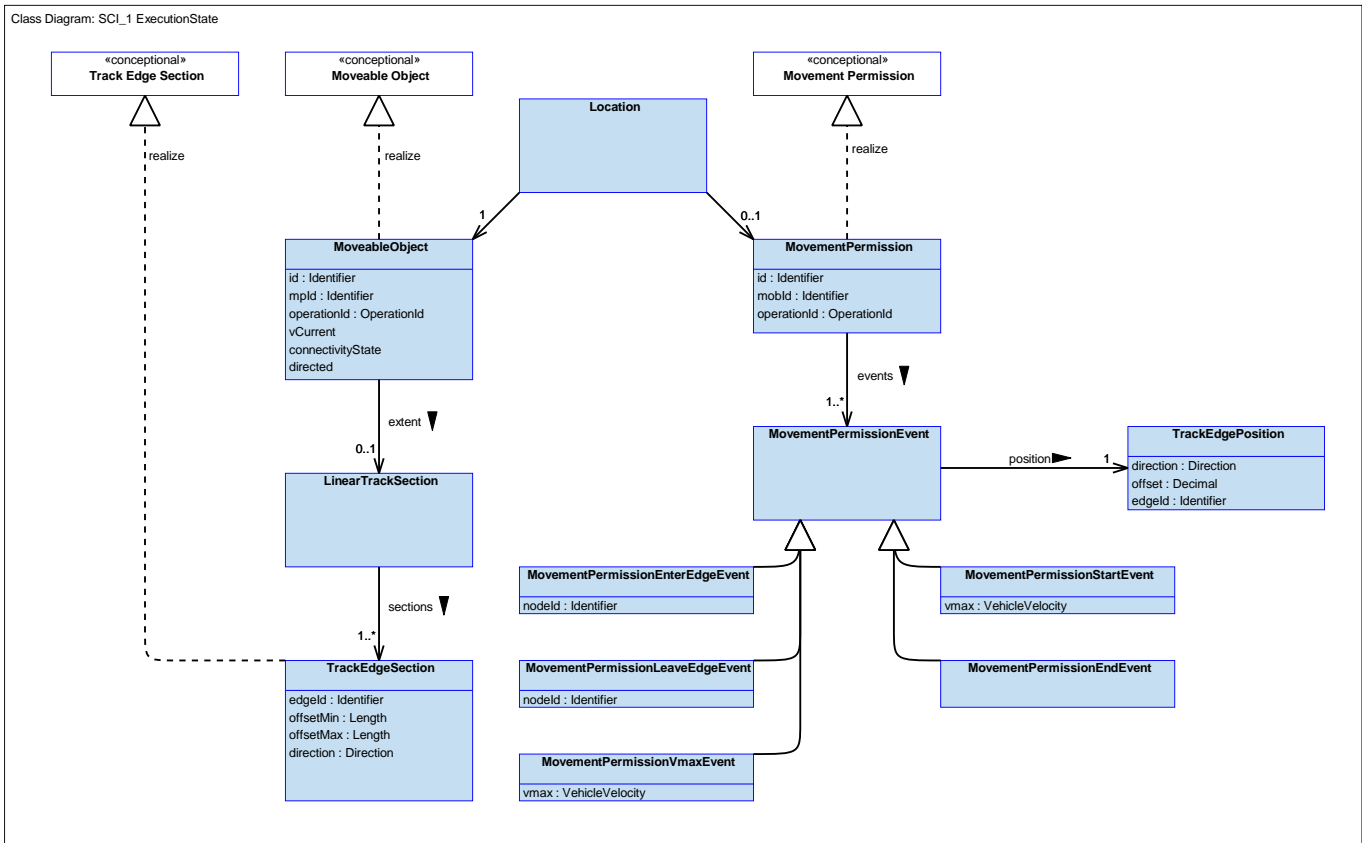


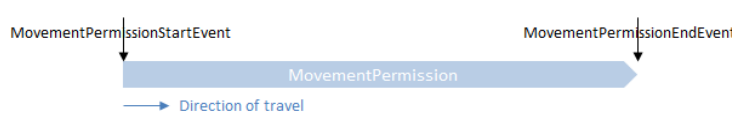
Figure 1 Class Diagram: SCI_1 ExecutionState

Description: [Class Diagram: SCI_1 ExecutionState](#) shows the structure of ExecutionState message exchanged on [SCI_1](#).

DynamicUmlClassName: Class Diagram

Name	Description
LinearTrackSection	A LinearTrackSection represents a linear path or extent within the topology. Its contained ordered list of Track Edge Section describes the track occupation. Each Track Edge Section of a LinearTrackSection specifies a partially or fully occupied section. To express that an Track Edge is fully covered by an extent, the Track Edge Section contains offsetMin=0 and offsetMax=sectionLength. To express that an Track Edge is partially covered by an extent, the Track Edge Section contains offsetMin>0 and/or offsetMax<sectionLength.
Location	Location describes which Track Edge Section are occupied by Moveable Object .

	Document Number and Issue	RCA.Doc.25, Public Snapshot (v0.0.7)
	Date of Publish	06-12-2019
	Page No	10

Name	Description
MoveableObject	An implementation of the Moveable Object concept.
MovementPermission	An implementation of the Movement Permission concept.
MovementPermissionEndEvent	MovementPermissionEndEvent is always the last MovementPermissionEvent of the sequence of events of a Movement Permission . It determines the position at which the Movement Permission ends. By convention, the sequence of events in a Movement Permission is listed in the direction of travel.
MovementPermissionEnterEdgeEvent	A MovementPermissionEnterEdgeEvent appears for every Track Edge that is part of the extent of a Movement Permission , except for the first Track Edge . This MovementPermissionEvent always comes in pairs with MovementPermissionLeaveEdgeEvent : whenever an Track Edge is declared to be entered, another Track Edge must have been left immediately before.
MovementPermissionEvent	MovementPermissionEvent is the common superclass of all events occurring Movement Permission . This class itself is abstract.
MovementPermissionLeaveEdgeEvent	A MovementPermissionLeaveEdgeEvent appears for every Track Edge that is part of the extent of a Movement Permission , except for the last Track Edge . This MovementPermissionEvent always comes in pairs with MovementPermissionEnterEdgeEvent : whenever an Track Edge is declared to be left, another Track Edge must be entered.
MovementPermissionStartEvent	<p>MovementPermissionStartEvent is always the first event of the sequence of events of a Movement Permission. It determines the position at which the Movement Permission begins. By convention, the sequence of events in a Movement Permission is listed in the direction of travel.</p> 
MovementPermissionVmaxEvent	A MovementPermissionVmaxEvent specifies a change in the allowed maximum velocity of a Movement Permission . It can occur at any position within the extent of a Movement Permission . Commonly, it appears at positions where the topology dictates a change in the allowed maximum velocity.

	Document Number and Issue	RCA.Doc.25, Public Snapshot (v0.0.7)
	Date of Publish	06-12-2019
	Page No	11

Name	Description
TrackEdgePosition	An implementation of the Track Edge Point concept.
TrackEdgeSection	An implementation of the Track Edge Section concept.

Table 1 Description of Classes

4.2. Class Diagram: SCI_1 IO Items

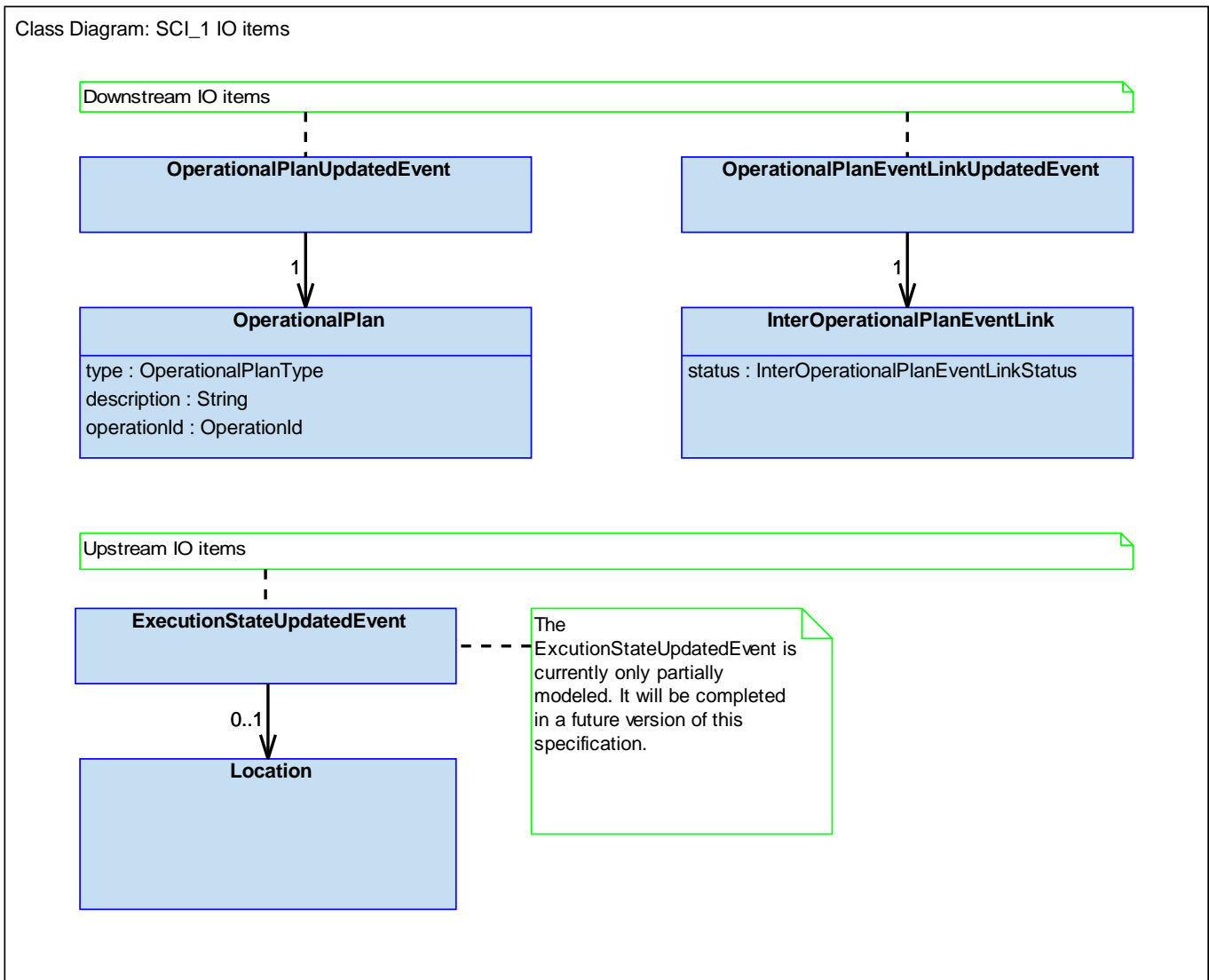


Figure 2 Class Diagram: SCI_1 IO Items

Description: [Class Diagram: SCI_1 IO Items](#) shows the IO items exchanged over [SCI_1](#).

DynamicUmlClassName: Class Diagram

Name	Description
------	-------------

	Document Number and Issue	RCA.Doc.25, Public Snapshot (v0.0.7)
	Date of Publish	06-12-2019
	Page No	12

Name	Description
ExecutionStateUpdatedEvent	ExecutionStateUpdatedEvent is an upstream IO item on SCI 1 . It is published by APS to inform about changes, such as updated positions of Moveable Object , changes of Movement Permissions , progress or differences to the Operational Plan .
InterOperationalPlanEventLink	Defines the relationship between two Operational Segments from different Operational Plans . Those different Operational Plans that represent different train runs.
Location	Location describes which Track Edge Section are occupied by Moveable Object .
OperationalPlan	An implementation of the Operational Plan concept.
OperationalPlanEventLinkUpdatedEvent	OperationalPlanEventLinkUpdatedEvent is published by TMS-PAS to inform consumers about a newly created, planned or updated Operational Plan Event Link . The Operational Plan Event Link describes relations between events of OperationalSegment and is intended to model relations between different Operational Plans (Inter Operational Plan relations, e.g. connection, train sequence). However the data model allows to model relations inside an Operational Plan .
OperationalPlanUpdatedEvent	OperationalPlanUpdatedEvent is published by TMS-PAS on SCI 1 to inform about a newly created or updated Operational Plan . OperationalPlanUpdatedEvent is a downstream IO item.

Table 2 Description of Classes

	Document Number and Issue	RCA.Doc.25, Public Snapshot (v0.0.7)
	Date of Publish	06-12-2019
	Page No	13

4.3. Class Diagram: SCI_1 OperationalPlan

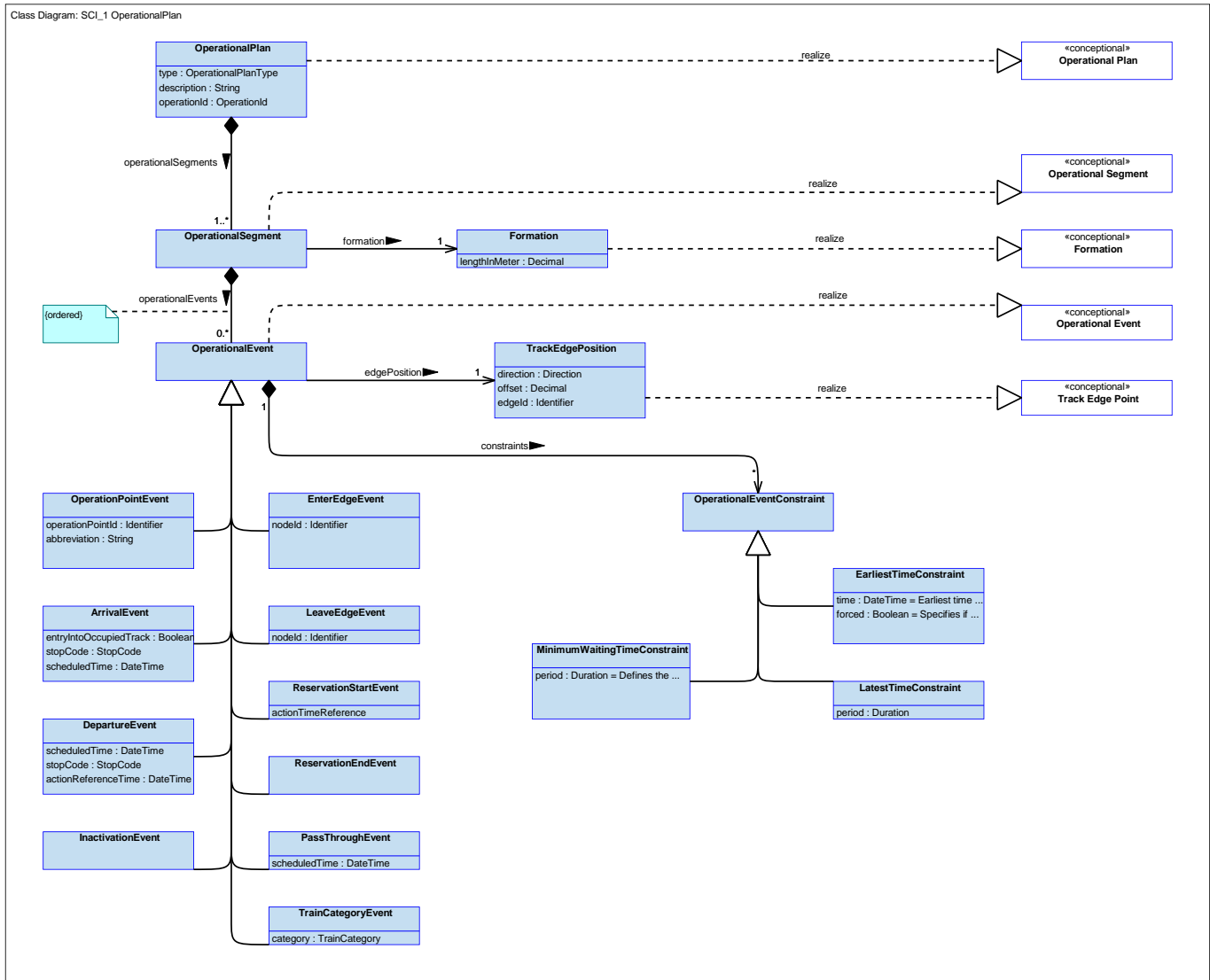


Figure 3 Class Diagram: SCI_1 OperationalPlan

Description: [Class Diagram: SCI_1 OperationalPlan](#) shows the structure of [OperationalPlan](#) message exchanged on [SCI_1](#).

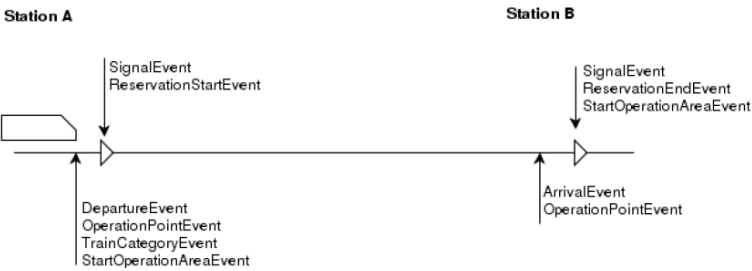
DynamicUmlClassName: Class Diagram

Name	Description
ArrivalEvent	ArrivalEvent represents the arrival (position on track) of the train in the operation point with a planned stop
DepartureEvent	DepartureEvent represents the departure (position on track) of a train. It may follow after a planned ArrivalEvent or it may be the first departure in an OperationalPlan .

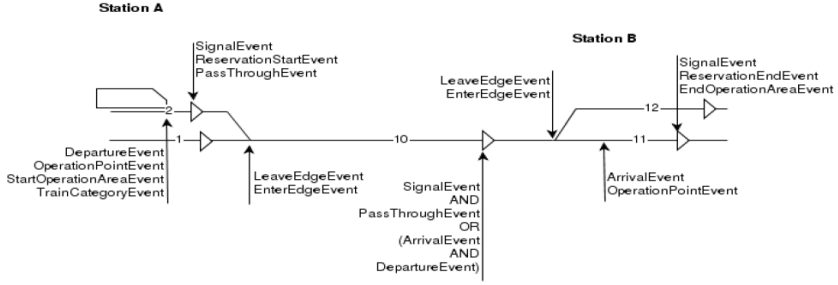
	Document Number and Issue	RCA.Doc.25, Public Snapshot (v0.0.7)
	Date of Publish	06-12-2019
	Page No	14

Name	Description
EarliestTimeConstraint	An EarliestTimeConstraint requires that its associated OperationalEvent is supposed to happen no earlier than the time specified in this constraint. The EarliestTimeConstraint can be only be applied to DepartureEvent to define an earliest departure time.
EnterEdgeEvent	EnterEdgeEvent represents the head of the train reaching the beginning of a Track Edge . EnterEdgeEvent usually follows a LeaveEdgeEvent . There is an exception to this rule: If EnterEdgeEvent comes after an ArrivalEvent and the train will not pass the end of the Track Edge . This sequence of Operational Events describes the route path defined by the Operational Plan .
Formation	An implementation of the Formation concept. As an simplification it only contains the length of the Formation .
InactivationEvent	InactivationEvent represents a point in the OperationalSegment , after that a train won't continue its journey on the planned route. It is used for trains with undetermined delay of the journey. The inactivation of the train ends, if after an update of the OperationalPlan this InactivationEvent is not present anymore in the OperationalSegment .
LatestTimeConstraint	LatestTimeConstraint requires that its associated OperationalEvent is supposed to happen no later than the time specified in this constraint. It is used, for example, to specify the latest arrival time in an ArrivalEvent . This constraint can only be defined on ArrivalEvents .
LeaveEdgeEvent	LeaveEdgeEvent represents the head of train leaving (position on track) the Track Edge at its end. LeaveEdgeEvent is always immediately followed by an EnterEdgeEvent . This sequence of OperationalEvents describes the route path defined by the OperationalPlan .
MinimumWaitingTimeConstraint	A MinimumWaitingTimeConstraint requires that its associated OperationPointEvent may not be passed before the specified time (period) after the most recent stop is elapsed. This constraint must be defined on the DepartureEvent .
OperationalEvent	An implementation of the Operational Event concept.
OperationalEventConstraint	OperationalEventConstraint is the superclass of all constraints that can be attached to an OperationalEvent . Such a constraint describes conditions that must be met before an OperationalEvent may be considered ready for production.
OperationalPlan	An implementation of the Operational Plan concept.

	Document Number and Issue	RCA.Doc.25, Public Snapshot (v0.0.7)
	Date of Publish	06-12-2019
	Page No	15

Name	Description
OperationalSegment	<p>An implementation of Operational Segment. Depending on the scenario an OperationalSegment must have several OperationalEvent associated.</p> <p>Scenario: Single Track Edge</p> <p>This case describes a basic movement on a single Track Edge. The figure shows the Operational Event required for this movement.</p>  <p>As a minimum an OperationalSegment (except cancelled trains) must contain the following Operational Events:</p> <ul style="list-style-type: none"> • StartOperationAreaEvent (possibly empty list) • DepartureEvent with earliestTimeConstraint • OperationPointEvent (for departure) • TrainCategoryEvent • ReservationStartEvent • ArrivalEvent • OperationPointEvent (for arrival) • ReservationEndEvent • EndOperationAreaEvent (possibly empty list) <p>For all Operational Event an TrackEdgePosition is required.</p> <p>Scenario: Train run from station A to station B</p> <p>This case describes a train run from station A to station B, where 2 switches are passed by the train run.</p>

	Document Number and Issue	RCA.Doc.25, Public Snapshot (v0.0.7)
	Date of Publish	06-12-2019
	Page No	16

Name	Description
	 <p>This scenario requires additional Operational Event in the OperationalSegment than the single Track Edge scenario:</p> <ul style="list-style-type: none"> • and additionally a PassThroughEvent or (ArrivalEvent AND DepartureEvent) • LeaveEdgeEvent • EnterEdgeEvent
OperationPointEvent	OperationPointEvent represents and defines the Operation Points on a position of the track in the OperationalSegment . OperationPointEvent can have a MinimumWaitingTimeConstraint associated with it.
PassThroughEvent	PassThroughEvent represents a drive through of a train at a defined position on the track at a planned time.
ReservationEndEvent	<p>ReservationEndEvent represents the end of an OperationalSegment route. The train should not reach this position, but this event is necessary to indicate the end of an OperationalSegment. It is especially important in places where the end of the path (target signal) is not unique.</p> <p>Example:</p> <p>The planned arrival for the train is on track 4014. Between ArrivalEvent and end of route is a point, so there are multiple ends possible. The ReservationEndEvent defines where the route for this movement ends.</p>

	Document Number and Issue	RCA.Doc.25, Public Snapshot (v0.0.7)
	Date of Publish	06-12-2019
	Page No	17

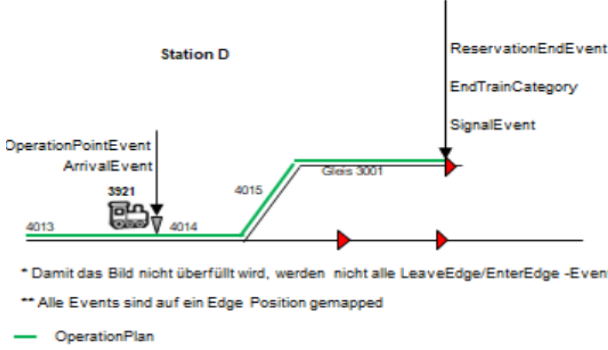
Name	Description
	 <p>* Damit das Bild nicht überfüllt wird, werden nicht alle LeaveEdge/EnterEdge -Events dargestellt ** Alle Events sind auf ein Edge Position gemapped — OperationPlan</p>
ReservationStartEvent	ReservationStartEvent represents the beginning of an OperationalSegment route. An OperationalSegment must contain one ReservationStartEvent and one ReservationEndEvent . These two events define the reserved path for the OperationalPlan (e.g. train run). Note: ReservationStartEvent is located at the same position as the first main signal in driving direction after the DepartureEvent .
TrackEdgePosition	An implementation of the Track Edge Point concept.
TrainCategoryEvent	TrainCategoryEvent defines the TrainCategory and the position on the Track Edge from which point the newly specified train category is valid. Every OperationalSegment must contain a TrainCategoryEvent at the beginning, even if the TrainCategory is the same as on the previous OperationalSegment . The TrainCategory is effective until the next defined TrainCategoryEvent in the OperationalSegment appears.

Table 3 Description of Classes

	Document Number and Issue	RCA.Doc.25, Public Snapshot (v0.0.7)
	Date of Publish	06-12-2019
	Page No	18

4.4. Class Diagram: SCI_1 OperationalPlanEventLink

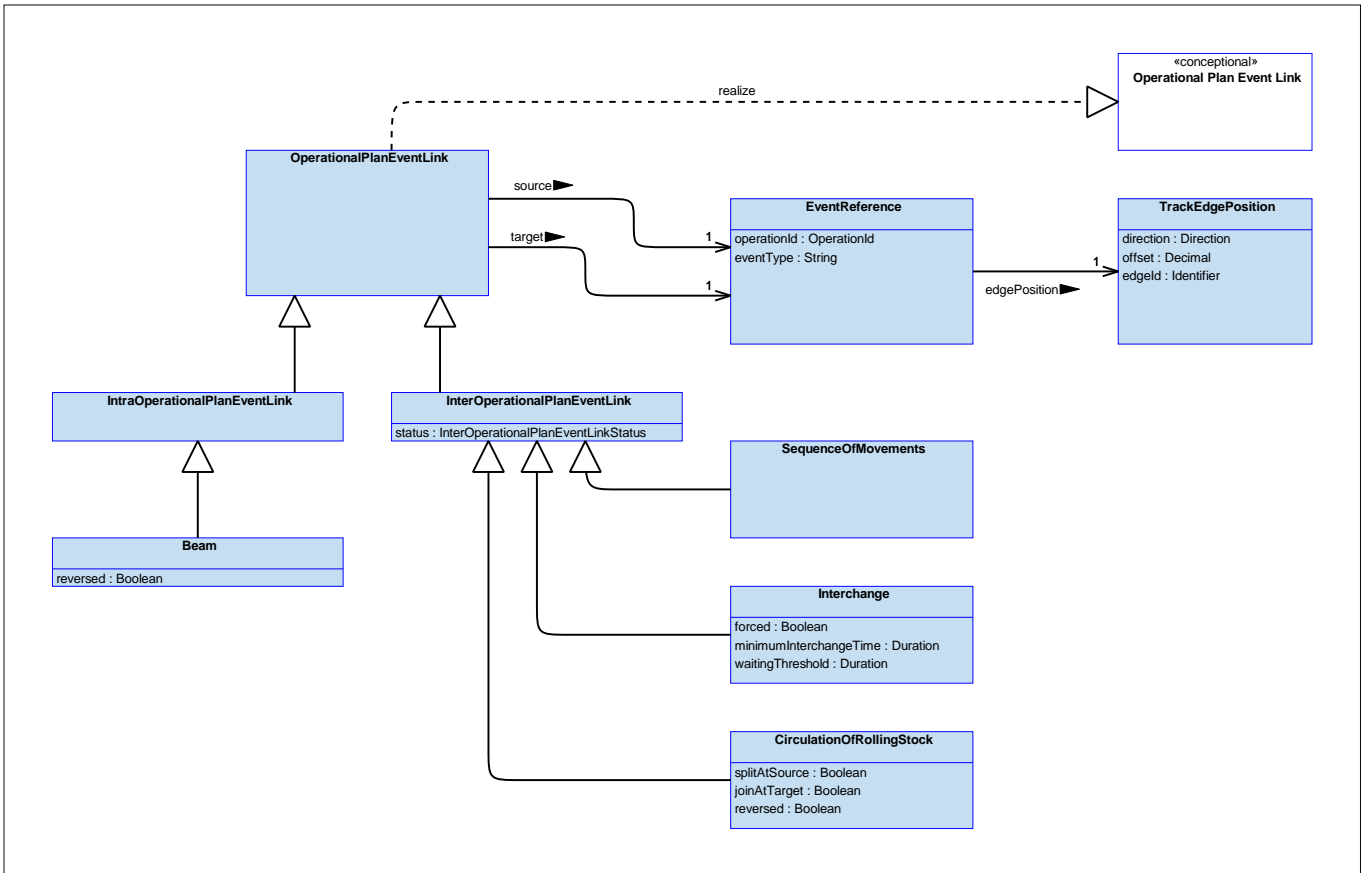


Figure 4 Class Diagram: SCI_1 OperationalPlanEventLink

Description: [Class Diagram: SCI_1 OperationalPlanEventLink](#) shows the structure of [OperationalPlanEventLinkUpdatedEvent](#) message exchanged on [SCI_1](#).

DynamicUmlClassName: Class Diagram

Name	Description
Beam	<p>Beam defines a relationship where the head of a train changes position without train run or shunting movement.</p> <p>Beams can occur in several situations. Examples:</p> <ul style="list-style-type: none"> • Turnback of a train • A train cannot continue because of malfunction, on opposite track a train replacement is ready. This train continues the planned path of the train. • In case of a strengthening of a train at the front of the train, the position of the head of train changes (jumps). • In the case that a train arrives on track 4 and leaves at track 13, there is a shunting movement necessary but in a first step not planned by an Operational Plan. These cases will disappear as soon as all movement are

	Document Number and Issue	RCA.Doc.25, Public Snapshot (v0.0.7)
	Date of Publish	06-12-2019
	Page No	19

Name	Description
	planned via Operational Plans .
CirculationOfRollingStock	<p>CirculationOfRollingStock describes how a Formation is passed from one Operational Plan to another. The CirculationOfRollingStock is defined between Operational Event of different Operational Plan. CirculationOfRollingStock links the following Operational Events together:</p> <p>With a train stop: source: ArrivalEvent or PassThroughEvent target: DepartureEvent or PassThroughEvent</p> <p>Without a train stop (e.g. change of an Operational Train Number without stopping): source: PassThroughEvent target: PassThroughEvent</p>
EventReference	This class identifies an Operational Event by the operationId , edgePosition and eventType .
Interchange	Interchange defines a connection relation between two Operational Plans where a train has to wait for the arrival of another train before departing. Conditions for the time between arrival and departure can be specified. These conditions can be enforced by a disposition.
InterOperationalPlanEventLink	Defines the relationship between two Operational Segments from different Operational Plans . Those different Operational Plans that represent different train runs.
IntraOperationalPlanEventLink	Defines the relationship between two Operational Segments within the same Operational Plan .
OperationalPlanEventLink	An implementation of the Operational Plan Event Link concept. The OperationalPlanEventLink describes the relationship between two OperationalEvents with the target and a source association.
SequenceOfMovements	SequenceOfMovements describes the chronological order of two linked Operational Events . The source Operational Event need to occur before the target Operational Event .
TrackEdgePosition	An implementation of the Track Edge Point concept.

	Document Number and Issue	RCA.Doc.25, Public Snapshot (v0.0.7)
	Date of Publish	06-12-2019
	Page No	20

Table 4 Description of Classes