RCA

Reference CCS Architecture

An initiative of the ERTMS users group and the EULYNX consortium

Concept: Standard Communication Interface Operational Plan (SCI-OP)

Preliminary issue

Document id: RCA.Doc.31

Table of Contents

Version History

0.6	06.12.2019	P. Moosmann, Y. Wyder	First version for feedback RCA core group
0.7	14.12.2019	P. Moosmann	Feedback of RCA core group integrated: Added migration support. Complemented open points
Gamma.1	28.01.2020	P. Moosmann	Integrated review feedback from RCA Core Group
1.1	30.06.2021	René Schönemann, Martin Kemkemer, Jens Wieczorek, Rouzbeh Boloukian-Roudsari, and other members of the RCA-OPE-Cluster	Content restructured and fundamentally extended, chapters on up-stream messages and on overarching topics added
1.2	30.08.2021	René Schönemann, Martin Kemkemer, Jens Wieczorek, Rouzbeh Boloukian-Roudsari, and other members of the RCA-OPE-Cluster	Open points and new findings added
1.3	15.09.2021	René Schönemann, Martin Kemkemer, Jens Wieczorek, Gabriele Löber, Rouzbeh Boloukian-Roudsari, and Marcus Völcker as members of the RCA-OPE-Cluster	Preliminary version for RCA BL0 R3
1.4	30.11.2021	René Schönemann, Martin Kemkemer, Jens Wieczorek, Gabriele Löber, Rouzbeh Boloukian-Roudsari, and Marcus Völcker as members of the RCA-OPE-Cluster	Release version for RCA BL0 R3

Release information

Basic document information:

RCA.Doc.31

Concept Standard Communication Interface Operational Plan (SCI-OP)

Cenelec Phase: 1

Version: 0.3 (0.A)

RCA Baseline set: 0

Approval date: 30.11.2021

Disclaimer

This issue is a preliminary version of this document. The content of this document reflects the current ongoing specification work of RCA. Formal requirements management and change management will be introduced in future iterations. The content may be unfinished, will likely contain errors and can be changed without prior notice.

Imprint

Publisher:

RCA (an initiative of the ERTMS Users Group and EULYNX Consortium)

Support and Feedback

For feedback, or if you have trouble accessing the material, please contact rca@eulynx.eu

1 Introduction

1.1 Purpose of document

This concept document describes the abstract concepts, messages, and the data model of the Standard Communication Interface Operational Plan (SCI-OP).

1.2 Maturity and Related Topics

This document is the enhanced publication on this topic from RCA. It depicts the concept of SCI-OP. Its main purpose is to establish the topic further and to continue discussions with stakeholders in the sector.

Parts of the concept of the Operational Plan have been implemented in smartrail 4.0 prototypes as well as for existing interface implementation between the Rail Control System (RCS) and the CTC system (Leittechnik Iltis). This document is partly based on this implementation work and has been enhanced with knowledge from Digitale Schiene Deutschland and other stakeholders.

The concept is still work in progress. Whenever it is already known that a section needs further elaboration, this is marked with red italic notes as this: *This section will be further elaborated in future releases*.

1.3 Structure of document

This concept is structured as follows:

- Chapter 1 gives a short introduction to SCI-OP and this concept.
- Chapter 2 describes the role of SCI-OP for the RCA, the design principles and addresses migration.
- Chapter 3 gives an overview on the structure of abstract concepts and messages of SCI-OP.
- Chapter 4 describes the abstract concepts and messages of the Operational Plan Execution.
- Chapter 5 describes the abstract concepts and messages of the Operating State.
- Chapter 6 contains the data model for the implementation of abstract concepts and messages as described in chapters 4 and 5.
- Chapter 7 describes aspects of SCI-OP that address both cross-cutting issues and non-functional requirements.
- Chapter 8 describes the open points which will be addressed in future releases of this concept.
- Chapter 9 is the Appendix and holds complex illustrations and examples.

1.4 Related documents

The following related RCA documents provide further information and build on this concept:

- RCA System Architecture, RCA.Doc.35
- RCA Glossary, RCA.Doc.4
- RCA Domain Knowledge Specification, RCA.Doc.18
- Concept: Principles of the safety logic, RCA.Doc.30
- Concept: Plan Execution, RCA.Doc.47
- Concept: Architectural Design for Plan Execution, RCA.Doc.49

1.5 Writing conventions

The following writing conventions apply to this document

- Abstract concepts are written with capitalized starting letters and with spaces.
- Messages are written in lower case with underscores.
- Classes are written in camel case with initial uppercase letter (upper camel case)
- Attributes are written in camel case with initial lowercase letter (lower camel case)

2 Standard Communication Interface Operational Plan

This chapter describes the role of SCI-OP in the RCA, the design principles, and addresses migration.

2.1 Introduction

The Standard Communication Interface Operational Plan (SCI-OP) is part of the Reference CCS Architecture. It is located at the system border of RCA and connects the external Planning System (PAS) with the RCA SubSys ATO Execution (AE) and Plan Execution (PE).

Figure 1 depicts its role between the systems for planning and execution.

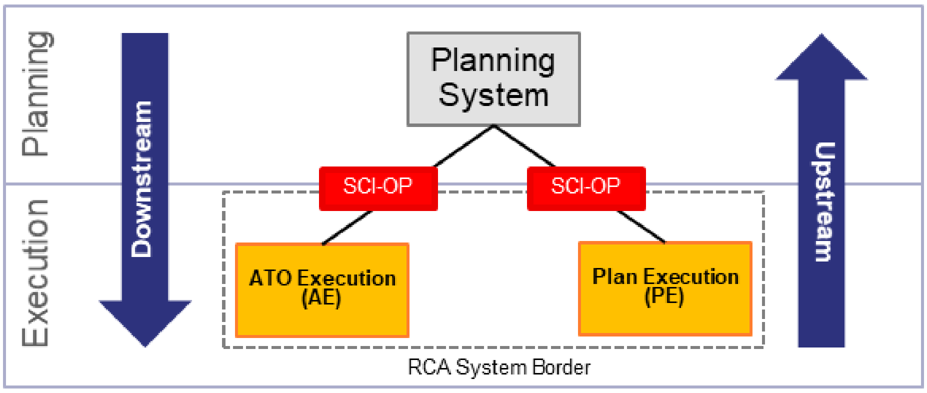


Figure 1: Scope of RCA interface SCI-OP

At this position the interface separates the two domains, Planning System and RCA, by encapsulation of RCA internals.

Primarily, the interface SCI-OP shall equally support the two RCA SubSys ATO Execution and SubSys Plan Execution as well as different kinds of external Planning Systems. Furthermore, it should support migration, i.e. it should enable the connection of existing CTC systems and associated interlockings via adapters. Accordingly, SCI-OP is the single interface between the external Planning System and the RCA system.

SCI-OP is a bidirectional interface. The downstream provides information from the Planning System to the RCA SubSys ATO Execution and SubSys Plan Execution. The upstream closes the feedback loop from the railway traffic execution to the Planning System.

The interface will be specified for and by RCA since no directly re-usable interface specification has been discovered in our research. These factors make this interface, and by so the messages sent over it, to a focal point of the architecture.

This concept contains the description of the abstract concepts of SCI-OP, a proposal for the messages to be exchanged, and the data model intended for the implementation of SCI-OP.

2.2 Objectives and System Requirements

The applicable objectives and system requirements can be found in *Concept: Plan Execution, RCA.Doc.47*, chapter: *Objectives and System Requirements*.

2.3 Design Principles

The elaboration of the message types and the data model of the interface SCI-OP considers the following design principles.

Design Principle: Every movement of a Physical Train Unit shall be planned with an Operational Plan Operational Movement.

Rationale: For effective schedule optimisation the Planning System needs to know and control every movement and every track usage. To use the tracks in the most efficient way, the Planning System creates Operational Plans for Operational Movement and observes their execution.

Design Principle: An Operational Plan defines exactly one Train Run.

Rationale: The granularity "one Train Run = one Operational Plan (per PAS - AoC)" has been chosen, because of several reasons. First it represents a Train Run that is a complete/self-contained entity. Second the data of an Operational Plan needs to be consistent and can be checked on this level. Third because of runtime efficiency. If a Train Run changes, only the directly and indirectly affected Operational Plans shall be recalculated and redistributed. This reduces the exchanged data volumes and makes it for consumer (SubSys Plan Execution and ATO Execution) easier to handle the changes. As a consequence, this means that running many trains simultaneously means to execute many Operational Plans at the same time.

Design Principle: All time specifications are clearly identifiable as operational or commercial

Rationale: There are commercial (e.g. published to customers) and operational (e.g. for performing railway operations timely) interests in scheduling activities. All time specifications in Operational Plans that are coming from an external Planning System into the RCA system are clearly identifiable as being either operational or commercial. The same applies to return messages from the RCA system to the Planning System.

Design Principle: Dependencies between Operational Plans are defined by Operational Plan Event Links, which are not changed by SubSys Plan Execution at any time.

Rationale: There are dependencies between Operational Movements. The most common cases of such a dependency are connecting trains or the defined order in which the trains should pass a given location. The concept of Operational Plan Event Links is an explicit way of defining such dependencies. The order of the Operational Movements resulting from these dependencies is not changed by the SubSys Plan Execution at any time but is the sole responsibility of the Planning System. However, manual intervention is possible in the event of an incident via the RCA Workbench through interface SWI-PE. Operational changes from the RCA Workbench will be passed through SWI-PE to SubSys Plan Execution and further via SCI-OP to the Planning System and to SubSys ATO Execution.

Design Principle: The interface SCI-OP treats Operational Plans as data objects in a dynamic environment

Operational Plans will be recalculated and redistributed based on the operational situation and the execution progress. This affects Operational Plans already in execution as well as those which will be carried out in the future. The interface SCI-OP and the SubSys connected with it will be able to handle versions of Operational Plans from a technical and a business perspective. From a technical point of view, corresponding data fields must be kept available for versioning of Operational Plans. From a business point of view, it must be ensured that updates of Operational Plans are executable and fit with the current operational situation.

Design Principle: Abstraction Level and product independent interface definition

Rationale: The RCA target picture shall be fully supported by the abstraction level and shall be product/system independent. For reasons of interoperability, different concepts (such as different GoA-Levels, fixed block or moving block, advanced safety systems and legacy interlockings) shall be supported directly or by introducing additional adaptors (mappers, translators).

Design Principle: Extensibility of the interface definition

Rationale: To support future changes, the design of the SCI-OP should be open for enhancements. It shall be possible to support a stepwise (for example regional) migration. This could allow to use Operational Plans with existing CTC systems and associated interlockings.

Design Principle: Consideration of Objectives and System Requirements of SubSys Plan Execution

A set of Objectives and System Requirements were designed for the RCA SubSys Plan Execution. These Objectives and System Requirements also apply without restriction to the interface SCI-OP. For details on Objectives and System Requirements of SubSys Plan Execution, please refer to document RCA.Doc.47 ”Concept: Operational Plan”.

2.4 SCI-OP in the Reference CCS Architecture (RCA)

Infrastructure management organisations typically run separate processes for railway traffic planning and railway traffic execution. The interface SCI-OP connects planning and execution processes and thus external, IM-specific systems with RCA.

The Planning System is for RCA an external, IM-specific system and not part of the RCA. The Planning System communicates over interface SCI-OP with RCA. SubSys Plan Execution and SubSys ATO Execution are the RCA SubSys, that are implementing SCI-OP. For an illustration of the interface SCI-OP in the RCA and the relation between its subsystems, it shall be referred again to Figure 1.

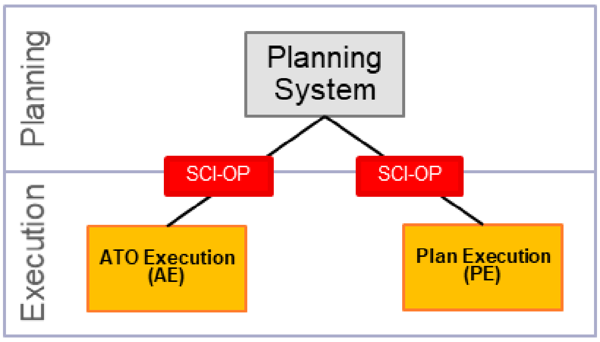
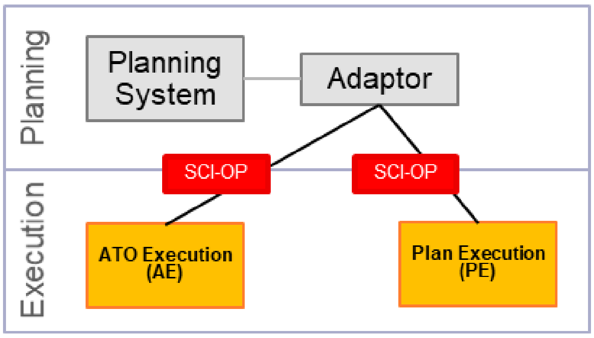
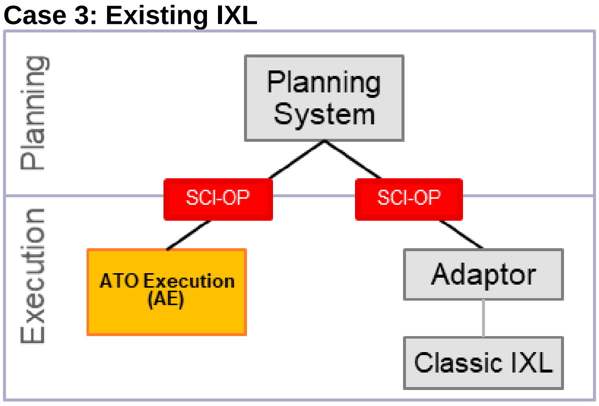
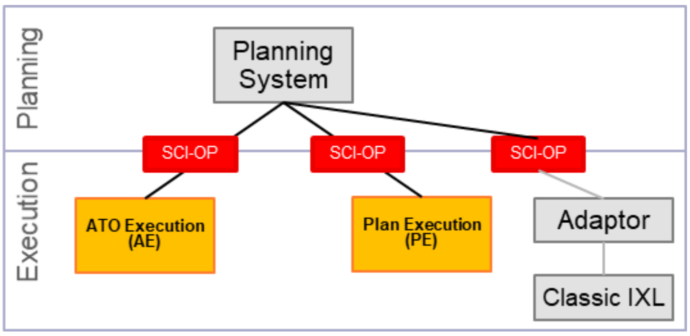
Based on the information taken from a received Operational Plan, SubSys Plan Execution calculates and triggers requests for Drive Protection Sections, Movement Permissions, Usage Restriction Areas and Warning Areas via SCI-CMD in order to implement Operational Movements as well as Operational Restrictions and Operational Warning Measures. SubSys ATO Execution generates instructions for ATO based on the information taken from the Operational Plan, so that Train Units autonomously drive the Operational Movements as planned. SubSys ATO Execution and SubSys Plan Execution report the execution progress of the Operational Plan back to the Planning System by the Operational Plan Execution Report and the Operating State.

Since SubSys ATO Execution and SubSys Plan Execution are part of the RCA specification, they are to be considered standard SubSys from an RCA perspective. The Planning System will be IM-specific. Hence, SCI-OP connects standard RCA SubSys and proprietary Planning Systems.

2.5 Role of SCI-OP during migration

SCI-OP has a crucial role during migration. It bridges different technologies of different development states and provides interoperability for the involved systems. The following table shows different blueprints how SCI-OP could be applied. SCI-OP provides the flexibility to connect a wide variety of Planning Systems to RCA and it allows to connect existing interlocking (IXL).

Table 1: Use cases of the application of the SCI-OP

Case 1: Planning System 	This is the desired final state. Any Planning System that implements SCI-OP can be connected to RCA.
Case 2: Existing Planning System 	If the Planning System cannot directly provide SCI-OP, an adaptor could translate from the existing Planning System interface to SCI-OP. This allows to use many existing Planning Systems, that are used today by the IMs.
Case 3: Existing IXL 	This case is primarily thought as a migration step. First SCI-OP would be introduced between the Planning System and the existing CTC/IXL by implementing an adaptor, that translates the Operational Plan to the language, which the CTC/IXL understands. In this case, it would be possible to already use SubSys ATO Execution.
Case 4: Partly existing IXL and Planning System already compatible with SCI-OP 	During migration, RCA infrastructure may be rolled out stepwise. This means that parts of the infrastructure will be equipped with RCA components, whereas at the same time, others won't. In this case, the Planning System will exchange data via SCI-OP with RCA components and to an existing CTC/IXL via an adaptor.

3 General Structure of the Interface SCI-OP

SCI-OP is a bidirectional interface. Its main purpose is the provision of Operational Plans to the RCA System. This results in the single, but very significant downstream message of the interface. The upstream messages sent from the RCA System are required by the Planning System to detect deviations between Operational Plans and their execution as well as to identify upcoming or existing conflicts between Operational Plans and to develop appropriate countermeasures. The upstream of the interface SCI-OP consists generally of the following streams:

- Operational Plan Execution Response (acceptance or rejection of execution request)
- Operational Plan Execution Report (progress of an accepted Operational Plan)
- Operating State (actual state of railway operations)

The messages transmitted via the interface SCI-OP are logically divided into two groups, Operational Plan Execution and Operating State. An overview on the main structure of SCI-OP is illustrated in Figure 2.

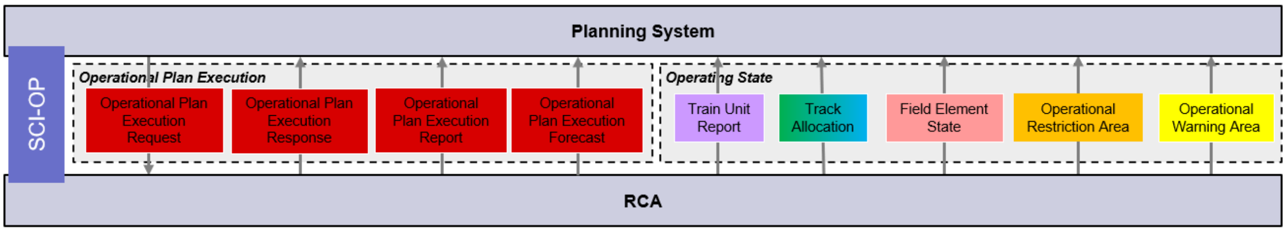


Figure 2: General structure of SCI-OP

Operational Plan Execution:

The main purpose of the SCI-OP is the provision of Operational Plans to the RCA System. Accordingly, the Operational Plan Execution comprises the major downstream message type from an external Planning System to the RCA System as well as relevant upstream return messages from the RCA System to the Planning System. All necessary abstract concepts, message type descriptions and further details of the Operational Plan Execution will be described in chapter 4.

Operating State:

For providing a better picture on the current state of the RCA System to the external Planning System, the Operating State delivers numerous upstream messages. They deliver, among others, information about the currently operating Train Units, the occupation of tracks, or the settings of Field Elements. All necessary abstract concepts, message type descriptions and further details of the Operating State will be described in chapter 5.

4 Operational Plan Execution

➔ ***Abstract concept: Operational Plan Execution***
The Operational Plan Execution is the logical representation of all messages concerning the execution of Operational Plans in the Area of Control. This includes the Operational Plan Execution Request, the Operational Plan Execution Response, the Operational Plan Execution Report and the Operational Plan Execution Forecast.
(also refer to RCA Domain Knowledge Specification, RCA.Doc.18)

This chapter describes the abstract concepts as well as the messages of the Operational Plan Execution of SCI-OP. The messages of the Operational Plan Execution are embraced into four abstract concepts, as illustrated in Figure 3.

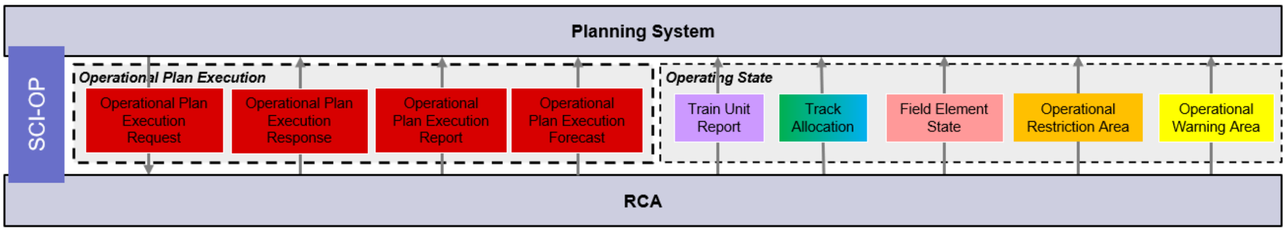


Figure 3: Main abstract concepts of SCI-OP with focus on the Operational Plan Execution

Section 4.1 contains the general definition and all necessary abstract concepts for the Operational Plan. It describes its different types and provides a first overview on the data structure of each type of Operational Plan. Section 4.2 comprises definitions of the downstream message and section 4.3 of the upstream messages, likewise.

4.1 Abstract Concepts Operational Plan

➔ ***Abstract Concept: Operational Plan***
The Operational Plan is the result of the planning process performed by the Planning System. It describes a planned Operational Movement, Operational Restriction or Operational Warning Measure on a specified area through a temporal sequence of Operational Events to be implemented by SubSys ATO Execution and/or SubSys Plan Execution in the Area of Control.
(also refer to RCA Domain Knowledge Specification, RCA.Doc.18)

➔ ***Abstract concept: Operational Plan Execution Request***
The Operational Plan Execution Request is the request to implement an Operational Plan. It is sent by the Planning System to SubSys ATO Execution and SubSys Plan Execution via SCI-OP.
(also refer to RCA Domain Knowledge Specification, RCA.Doc.18)

➔ ***Abstract concept: Operational Plan Execution Response***
The Operational Plan Execution Response is the response (acceptance or rejection) to an Operational Plan Execution Request. It is provided by SubSys ATO Execution and SubSys Plan Execution via SCI-OP.
(also refer to RCA Domain Knowledge Specification, RCA.Doc.18)

➔ ***Abstract concept: Operational Plan Execution Report***
The Operational Plan Execution Report describes the execution progress of the Operational Plan. It is provided by SubSys ATO Execution and SubSys Plan Execution via SCI-OP.
(also refer to RCA Domain Knowledge Specification, RCA.Doc.18)

➔ ***Abstract concept: Operational Plan Execution Forecast***
The Operational Plan Execution Forecast describes the execution forecast of the Operational Plan. It

is provided by SubSys ATO Execution via SCI-OP.
(also refer to RCA Domain Knowledge Specification, RCA.Doc.18)

An Operational Plan will be issued by the Planning System for any operationally relevant activity. This comprises all movements of Physical Train Units (incl. shunting operations), restrictions due to e.g. infrastructure maintenance and construction works, and warning measures during restrictions.

The implementation of an Operational Plan can be requested by an Operational Plan Execution Request (see details later in subsection 4.2) that is issued by the Planning System to the RCA System. The RCA reacts to the Operational Plan Execution Request with Operational Plan Execution Response and Operational Plan Execution Report (see details later in subsection 4.3).

An Operational Plan contains all necessary information for the RCA SubSys Plan Execution and SubSys ATO Execution to implement the Operational Plan. SubSys Plan Execution will be enabled to provide the capacity on the infrastructure for executing the Operational Plan. SubSys ATO Execution will be enabled to generate necessary driving instructions for the Physical Train Unit assigned to the respective Operational Plan.

An Operational Plan can have different characteristics depending on whether it is used to describe an Operational Movement, an Operational Restriction, or an Operational Warning Measure. The following figure shows the structure of the abstract concept Operational Plan:

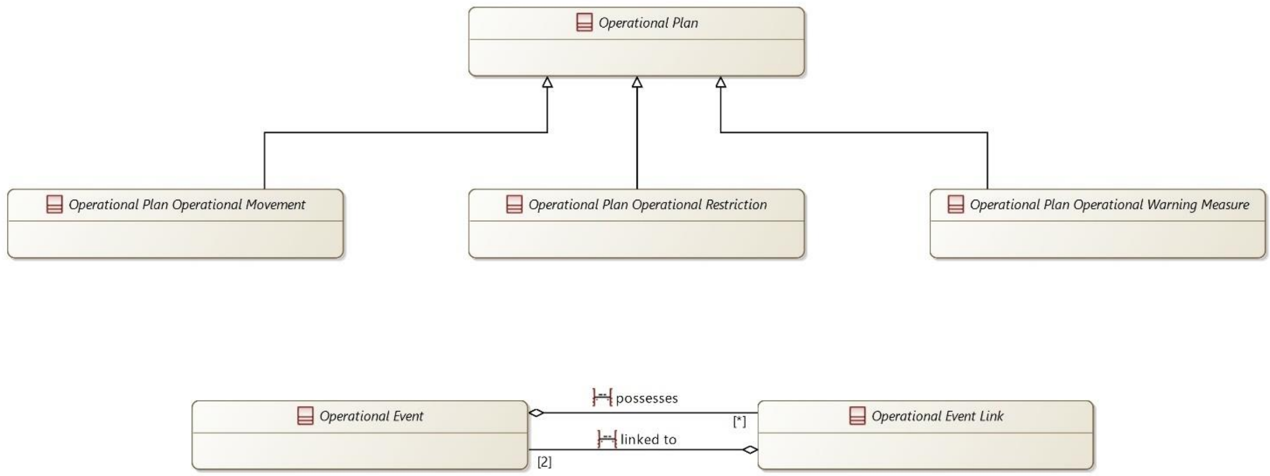


Figure 4: Structure of abstract concept Operational Plan

An Operational Plan for an Operational Movement, described in subsection 4.1.2, describes a Train Run by a precisely defined ordered set of Operational Events. The Operational Plan may be divided into different Operational Segments to represent different compositions of the Physical Train Unit during the Train Run.

An Operational Plan for an Operational Restriction described in subsection 4.1.3, is not bound to a Physical Train Unit and contains likewise a precisely defined ordered set of Operational Events and information on Operational Restrictions in an Operational Restriction Area.

An Operational Plan for an Operational Warning, described in subsection 4.1.4, is not bound to a Physical Train Unit, too. It contains likewise a precisely defined ordered set of Operational Events and information on Warning Areas from an operational perspective.

The detailed description of the different types of Operational Plans is preceded by subsection 4.1.1. It contains the abstract concepts which are applicable to all types of Operational Plans.

4.1.1 Abstract Concepts Applicable to all Types of Operational Plans

The following subsections describe abstract concepts of objects that are applicable to all types of Operational Plans. Operational Events, described in section 4.1.1.1, characterise specific actions of an Operational Plan. Operational Plan Event Links, described in section 4.1.1.2, are the instrument to link two Operational Events.

4.1.1.1 Operational Event

➔ ***Abstract Concept: Operational Event***
An Operational Event is a description of a single planned action (e.g. stop/passage of a Physical Train Unit, start/end of an Operational Restriction Area, start/end of an Operational Warning Area) defined in the Operational Plan.
(also refer to RCA Domain Knowledge Specification, RCA.Doc.18)

Depending on the type of an Operational Plan, different Operational Events with different properties are required. Therefore, the Operational Plan distinguishes between:

- Operational Movement Events (will be described in section 4.1.2.2)
- Operational Restriction Events (will be described in section 4.1.3.3)
- Operational Warning Measure Events (will be described in section 4.1.4.3)

An Operational Event contains the time and the position at which it shall be executed. Accordingly, all actions necessary for the event to be implemented shall be carried out before that defined time. It is in the responsibility of the consuming SubSys to ensure the timely triggering of necessary requests to RCA subsystems on other layers to ensure that the Operational Event will be implemented as planned.

4.1.1.2 Operation Plan Event Link

➔ ***Abstract Concept: Operational Plan Event Link***
An Operational Plan Event Link is a dependency on place and/or time between two Operational Events of the same or of two different Operational Plans.
(also refer to RCA Domain Knowledge Specification, RCA.Doc.18)

The concept allows to define dependencies between Operational Events of the same or different Operational Plans independent of their types (Operational Plan Operational Movement, Operational Plan Operational Restriction, Operational Plan Operational Warning).

The concept leaves the range of applications of the Operational Plan Event Links open. It allows the linking of any two Operational Events, independently of their technical plausibility or whether the linking makes sense from a business point of view. The correct setting of Operational Event Links is the sole responsibility of the Planning System.

The following table contains mandatory and exemplarily optional use cases for the linking of Operational Events across types of Operational Plans. It is not exhaustive and is intended to show the application range of Operational Plan Event Links in exemplary way.

Link from	to Operational Movement	to Operational Restriction	to Operational Warning
Operational Movement	• Dependencies between train runs of one Physical Train Unit: Succeeding train run cannot start if preceding train run is not finished. It is mandatory to create Operational Plan Event Links to the last (first) Operational Event of the preceding (succeeding) train run , if it exists for the considered Physical Train Unit. • Dependencies due to splitting/joining of Physical Consists (is expected by, is waiting for) • Dependencies due to passenger connections at a station (wait for other Physical Train Unit to perform passenger exchange)	• A construction site can only start, after the last scheduled (and eventually delayed) Train Run has passed the Track Edge Section.	---
Operational Restriction	• A Physical Train Unit can pass a Track Edge Section only after a construction site on it has ended.	• Dependencies between two construction sites (e.g. construction step 2 cannot start if construction step 1 has not been completed) • Dependencies between infrastructural insufficiencies and its repair (e.g. Temporary Speed Restriction (= Operational Restriction 1) due to a track geometry error and construction site for its correction (= Operational Restriction 2)	• A construction site on track 1 requires the establishment of a Warning Area on Track 2.
Operational Warning Measure	---	• A Warning Area on Track 2 can only be removed after the construction works on track 1 have ended.	• Two Warning Areas which work only in conjunction and shall be created / removed together

4.1.2 Operational Plan Operational Movement

➔ ***Abstract concept: Operational Movement***
An Operational Movement is a planned or already implemented train run, which runs a defined train, at defined times, along a defined track path within the Area of Control.
(also refer to RCA Domain Knowledge Specification, RCA.Doc.18)

➔ ***Abstract concept: Operational Plan Operational Movement***
The Operational Plan Operational Movement defines the parameters for the implementation of an Operational Movement.
(also refer to RCA Domain Knowledge Specification, RCA.Doc.18)

To make schedules effective, all deployments of Physical Train Units shall be considered in the Planning System and executed as planned. An Operational Plan Operational Movement defines precisely such a scheduled train run, which runs a defined train, at defined times along a defined track path from its point of departure to its point of arrival over the railway network.

This section explains the abstract concept Operational Plan Operational Movement in detail. Figure 5 shows the structure of the abstract concept Operational Plan Operational Movement.

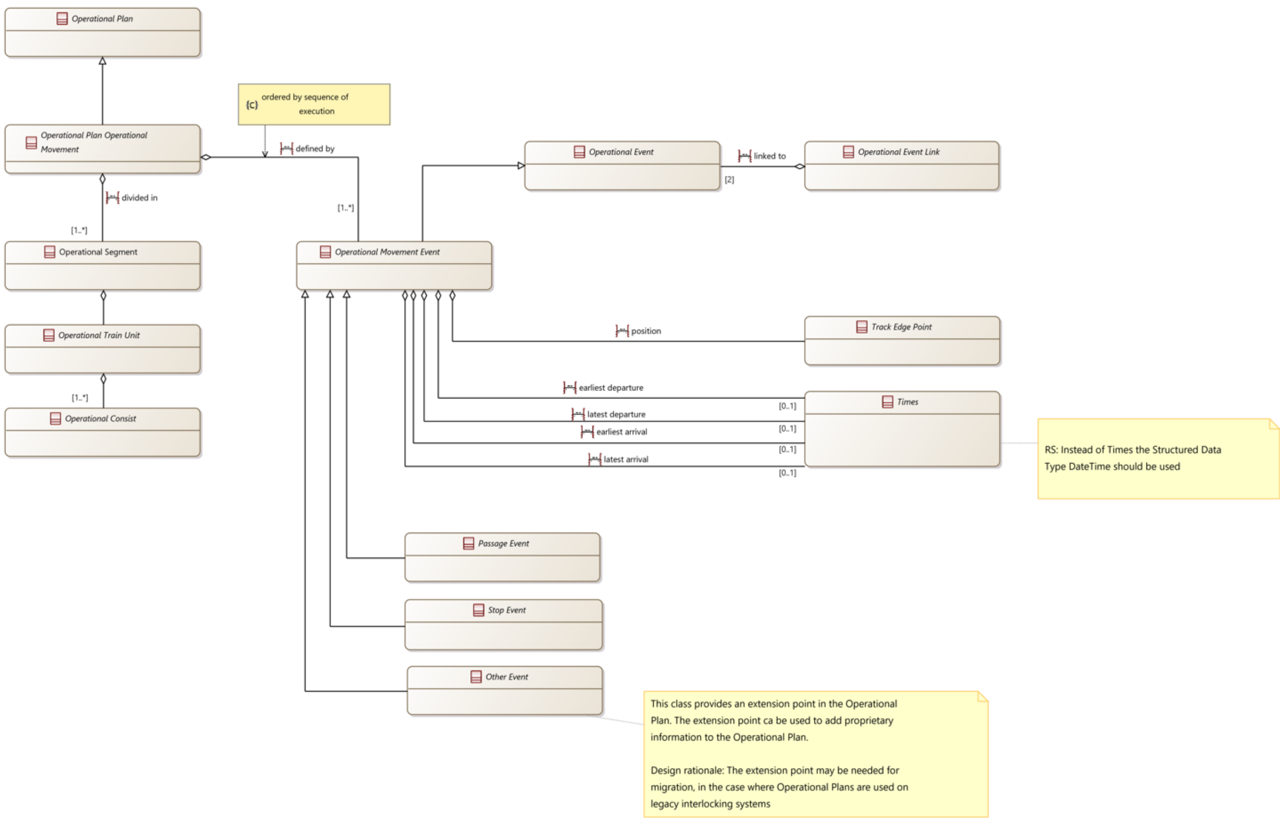


Figure 5: Structure of abstract concept Operational Plan Operational Movement

An Operational Plan Operational Movement has a unique identifier. It shall consist of one or more Operational Segments (described in section 4.1.2.1 and a set of Operational Movement Events (described in section 4.1.2.2). Operational Movement Events can have interdependencies via Operational Plan Event Links (already described in section 4.1.1.2). Section 4.1.2.3 contains an illustrated example of an Operational Plan Operational Movement.

4.1.2.1 Operational Segment

➔ ***Abstract concept: Operational Segment***
The Operational Segment divides an Operational Movement into several parts in order to represent different train compositions within the Operational Movement.
(also refer to RCA Domain Knowledge Specification, RCA.Doc.18)

➔ ***Abstract concept: Operational Train Unit***
The Operational Train Unit describes a sequence of Physical Consists and the operating parameters of the Physical Train Unit planned to be operated.
(also refer to RCA Domain Knowledge Specification, RCA.Doc.18)

➔ ***Abstract concept: Operational Consist***
The Operational Consist describes a sequence of the Physical Vehicles and the operating parameters of the Physical Consist planned to be operated.
(also refer to RCA Domain Knowledge Specification, RCA.Doc.18)

An Operational Plan Operational Movement consists of one or multiple Operational Segments. One Operational Segment refers to exactly one Operational Train Unit. A new Operational Segment has therefore to be created every time a new Operational Train Unit has to be referenced. This is the case in the following situations (list is not exhaustive):

- Adding Physical Consists or Physical Vehicles to a Physical Train Unit
- Changing the order of the Physical Consists of a Physical Train Unit
- Removing Physical Consists from a Physical Train Unit
- Direction change (alters the order of Physical Consists in the Physical Train Unit)
- Change of the Operational Train Category (see 6.2.4)
- Change of the train Number

Every Operational Segment has a sequence number which determines the order of the Operational Segments. The transition from one Operational Segment to another is determined by an Operational Movement Event (will be described in section 4.1.2.2).

4.1.2.2 Operational Movement Event

➔ ***Abstract concept: Operational Movement Event***
Operational Movement Events are Operational Events specified in the Operational Plan Operational Movement. They describe an ordered sequence of actions to take place at a certain position on the Track Path at a certain time.
(also refer to RCA Domain Knowledge Specification, RCA.Doc.18)

➔ ***Abstract concept: Track Path***
The Track Path is a gap-free and track-specific route on the railway network. It is used for describing the path of an Operational Movement in the Operational Plan Operational Movement.
(also refer to RCA Domain Knowledge Specification, RCA.Doc.18)

Each Operational Movement Event defines an action of an Operational Movement that should happen on a certain position at a certain time. Each Operational Movement Event is of a certain type. Currently, the following Operational Movement Events are defined:

Operational Event	Description
Passage Event	***Abstract concept: Passage Event*** *The Passage Event describes the planned passage of a Physical Train Unit. The Passage Event shall specify at least the position on the track and the time range (upper and lower bound) for the planned passage.* *(also refer to RCA Domain Knowledge Specification, RCA.Doc.18)*
Stop Event	***Abstract concept: Stop Event*** *The Stop Event describes the planned stop of a Physical Train Unit as well as all relevant planned actions to be carried out at planned stop of a Physical Train Unit.* *(also refer to RCA Domain Knowledge Specification, RCA.Doc.18)*
«Other» Event	***Abstract concept: Other Event*** *Other Event provides an extension point in the Operational Plan Operational Movement. The extension point can be used to add proprietary information to the Operational Plan.* *(also refer to RCA Domain Knowledge Specification, RCA.Doc.18)* It describes Events that are not required by RCA SubSys Plan Execution and SubSys ATO Execution but will be necessary for connected legacy IXL systems. If such events occur in an Operational Plan, they will be ignored by SubSys Plan Execution and SubSys ATO Execution.

Each Operational Movement Event shall be complemented with a reference to a position on the infrastructure and timing information at which the Operational Movement Event shall take place during plan execution.

Positions in Operational Movement Events

Each Operational Movement Event requires a precise reference to the position where the event shall be executed. This is a reference to a Track Edge Point based on Map Data in the Area of Control.

Timing Information in Operational Movement Events

With each Operational Movement Event, timing information (times) are provided. Four types of times are possible:

Times		Description
Earliest Arrival	[0..1]	Denotes the point in time when an arrival shall occur the earliest.
Latest Arrival	[0..1]	Denotes the point in time when an arrival shall occur the latest.
Earliest Departure	[0..1]	Denotes the point in time when a departure shall occur the earliest.
Latest Departure	[0..1]	Denotes the point in time when a departure shall occur the latest.

Overall multiplicity for timings in Operational Movement Events is [2..4]. How many timings are required, depends on the event type:

Timings for Passage Event: Latest Departure and Earliest Departure are required values.

Timings for Stop Events: A Stop Event can either denote an arrival, a departure, or both of a planned Mission. Accordingly, either both Arrival timings, or both Departure timings, or all four timings are required. If a Stop Event describes the first (last) stop of a Mission, then it contains only the departure (arrival) timings. In such cases it is mandatory to create Operational Plan Event Links (see section 4.1.1.2) to the last (first) Operational Event of the preceding (succeeding) Mission, if it exists for the considered Physical Train Unit. Summarising, a Stop Event can possess either:

- Timings for arrival and departure
- Timings for arrival and Operational Plan Event Link to subsequent Mission

- Timings for departure and Operational Plan Event Link to preceding Mission
- Timings for arrival only (if no Operational Plan Event Link to subsequent Mission can be created)
- Timings for departure only (if no Operational Plan Event Link to preceding Mission can be created)

Timings for Other Events: Latest Departure and Earliest Departure are required values. The other timings are optional.

With the specification of earliest and latest timings, time ranges can be specified. If the values of earliest and latest timings are equal, no degrees of freedom are in fact allowed. The time ranges are useful, because RCA component ATO-AT cannot drive an exact timeline and would get permanently re-controlled by the Planning System. Therefore, certain degrees of freedom should be provided for each Operational Movement Event. They give the consuming SubSys ATO Execution and Plan Execution certain leeway in determining the precise processing of Operational Events.

It is important to note that the timings of Operational Movement Events indicate the time at which APS shall be ready to execute an event. That means that all prerequisites (e.g. carrying out a sequence of events on SCI-CMD, like ensuring driveability and safety of the required track path, providing Movement Permissions to Physical Train Units) shall be carried out before that point in time. The consuming subsystems shall hold all information to determine the correct moments at which the necessary prerequisites for an Operational Movement Event shall be triggered.

4.1.2.3 Example of an Operational Plan Operational Movement

The properties of the Operational Plan Operational Movement described in this section are to be illustrated by means of an example. Therefore, Figure shows a simplified application of an Operational Plan.

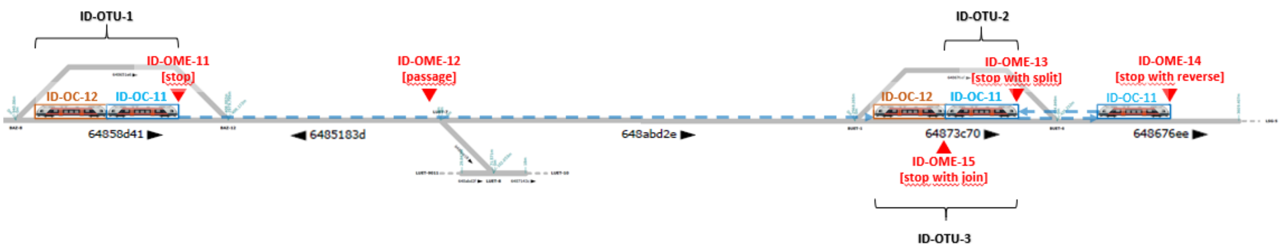


Figure 6: Simplified example of an Operational Plan Operational Movement

The depicted Physical Train Unit travels from left to right and its Operational Plan contains the following information.

- The Operational Train Unit (ID-OTU-1) which consists of 2 Operational Consists (ID-OC-11 at position 1 and ID-OC-12 at position 2) departs at the Operational Movement Event (ID-OME-11) at Track Edge (64858d41).
- The Operational Train Unit passes the Operational Movement Event (ID-OME-12) at Track Edge (6485183d).
- The Operational Train Unit stops at the Operational Movement Event (ID-OME-13) at Track Edge (64873c70) and performs a split. A new Operational Train Unit (ID-OTU-2) is defined, which contains only the Operational Consist (ID-OC-11 at position 1).
- The Operational Train Unit stops at the Operational Movement Event (ID-OME-14) at Track Edge (648676ee) and performs a reverse (in this case no new Operational Train Unit needs to be defined, because it's a single Operational Consist).
- The Operational Train Unit arrives at the Operational Movement Event (ID-OME-15) at Track Edge (64873c70) and performs a join. A new Operational Train Unit (ID-OTU-3) is defined, which contains both Operational Consists at reversed order (ID-OC-12 at position 2 and ID-OC-11 at position 1).

A data file printout of the depicted example is provided in the Appendix (section 9.1).

4.1.3 Operational Plan Operational Restriction

➔ ***Abstract concept: Operational Plan Operational Restriction***
The Operational Plan Operational Restrictions defines the parameters for the implementation of Operational Restriction Area(s) and the Operational Restriction(s) therein.
(also refer to RCA Domain Knowledge Specification, RCA.Doc.18)

The Operational Plan Operational Restriction is necessary to pursue maintenance, construction works, or safe operation in abnormal situations (e.g. obstacles on track or incidents). Operational Restrictions can either be created in the Planning System and provided as Operational Plan Operational Restriction to the RCA system, or in the RCA system. The latter are provided to the Planning System by the SubSys Plan Execution in the upstream (see section 5.4).

This chapter explains the abstract concept Operational Plan Operational Restriction in detail. Figure 7 shows the structure of the abstract concept Operational Plan Operational Restriction.

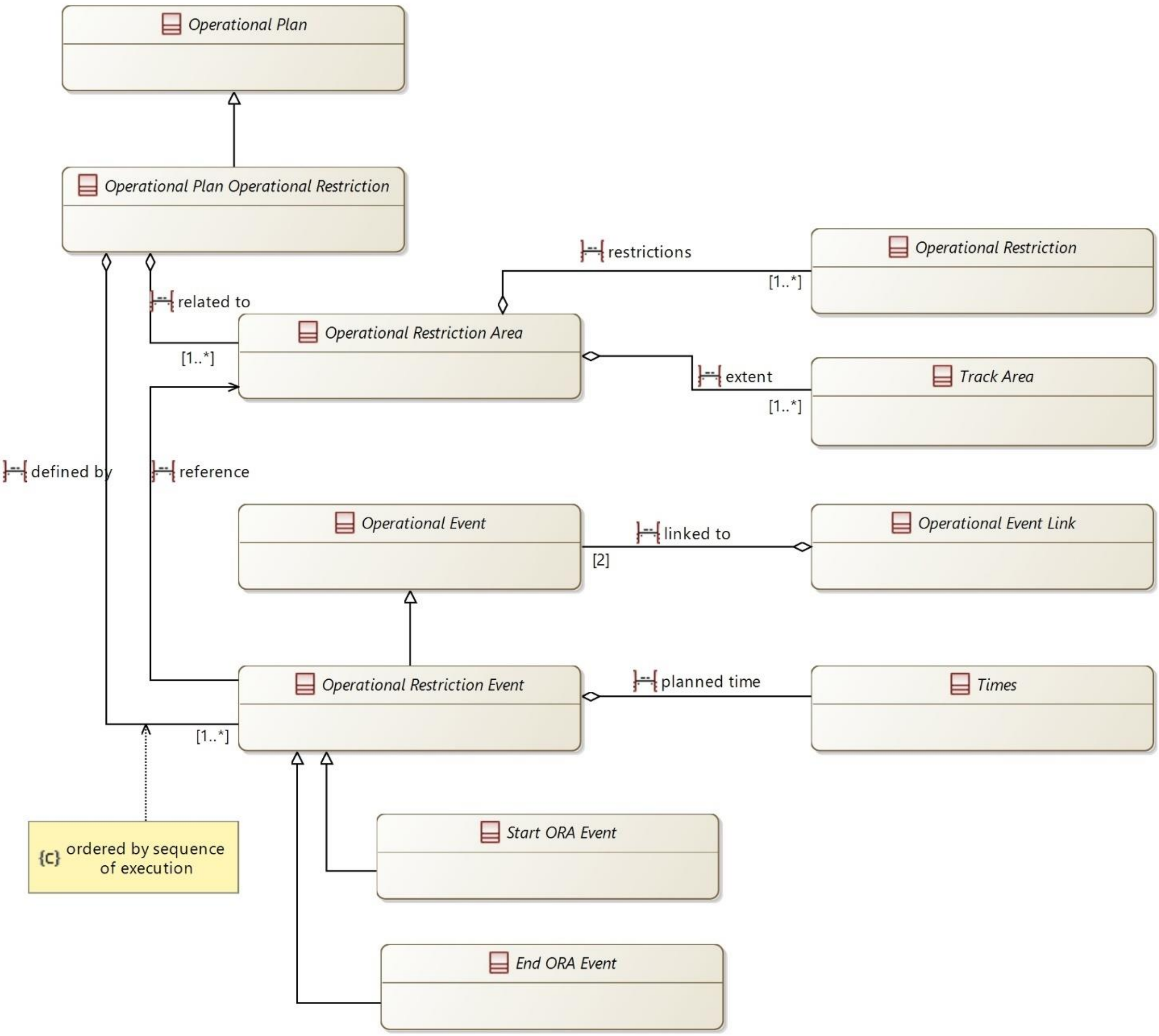


Figure 7: Structure of abstract concept Operational Plan Operational Restriction

The Operational Plan Operational Restriction has a unique identifier. An Operational Plan Operational Restriction contains one or more Operational Restriction Areas (described in section 4.1.3.1), one or more Operational Restrictions associated with the area (described in section 4.1.3.2), and one or more Operational Restriction Events (described in section 4.1.3.3). Each Operational Restriction Event refers to one Operational Restriction Area. Like all Operational Events, Operational Restriction Events can have interdependencies via Operational Plan Event Links (described in section 4.1.1.2).

4.1.3.1 Operational Restriction Area

➔ ***Abstract concept: Operational Restriction Area***
The Operational Restriction Area is an entity of the Operational Plan Operational Restriction and the Operating State. It describes the spatial dimensions of a planned or already implemented Operational Plan Operational Restriction within the Area of Control.
(also refer to RCA Domain Knowledge Specification, RCA.Doc.18)

The spatial dimension of an Operational Restriction Area is described as Track Edge Sections which provide the possibility to define a direction. Therefore, Operational Restrictions which are valid in one or both directions of travel can be represented. An Operational Restriction Area can contain one or more Track Edge Sections.

One or more Operational Restrictions are assigned to an Operational Restriction Area. These are described in the following subsection.

4.1.3.2 Operational Restrictions

➔ ***Abstract concept: Operational Restriction***
An Operational Restriction is a usage limitation on the railway network.
(also refer to RCA Domain Knowledge Specification, RCA.Doc.18)

An Operational Restriction can express different types of capacity reductions. By assigning an Operational Restriction to an Operational Restriction Area the type of usage restriction can be defined for the respective Operational Restriction Area.

The following table contains the available types of Operational Restrictions[1]:

Operational Restriction Type	Description
TrainType	Determines that the ORA shall be considered by specific train types only.
EngineeringTrainsOnly	Determines that the ORA can be accessed by specific Physical Train Units for maintenance or infrastructure constructions only.
TotalClosure	Determines that the ORA cannot be accessed by any Physical Train Unit.
AxleLoad	Determines that Physical Train Units with an axle load higher than indicated shall not enter the ORA.
CantDeficiency	Determines that Physical Train Units with a cant deficiency lower than indicated shall not enter the ORA.
OtherIntlTrainCat	Determines that Physical Train Units of a specific train category (from another AoC) shall not enter the ORA. *To be determined in coordination with APS cluster.*
TractionSystem	Determines that Physical Train Units with a specific traction system shall not enter the ORA. *To be determined in coordination with APS cluster.*

[1] *The listing is to be aligned regularly with the requirements and the capabilities of APS. Therefore, updates are to be expected in future versions of this document.*

TemporarySpeedRestriction	Determines that the Physical Train Unit's maximum velocity shall be limited to the indicated value.
DrivingModeRestriction	Determines that only Physical Train Units with a specific driving mode can enter the ORA. *To be determined in coordination with APS cluster, whether the Driving Modes that can be used, shall be defined in an OP.*
PermissiveMoveRestriction	Allows only permissive/on-sight movements (e.g. driver shall be present, move over a disturbed DPS)
AirTightness	Determines that only Physical Train Units with a specific air tightness can enter the ORA. *To be determined in coordination with APS cluster.*
LoadingGauge	Determines that Physical Train Units with a specific loading gauge shall not enter the ORA. *To be determined in coordination with APS cluster.*
LoadingType	Determines that Physical Train Units with a specific loading type shall not enter the ORA. *To be determined in coordination with APS cluster.*
NonStoppingArea	Determines that Physical Train Units shall not stop within the ORA.
OperationalRadioHole	Determines that radio communication is restricted in the ORA.
LessAdhesion	Determines that the Physical Train Unit's maximum adhesion shall be limited due to the indicated value. *To be determined in coordination with APS cluster.*
SoundHorn	Determines that Physical Train Units shall not use their sound horns in the ORA.
CrosswindSensitivity	Determines that Physical Train Units with a specific crosswind sensitivity shall not enter the ORA.

The types of Operational Restrictions are aligned with needed types of APS.

4.1.3.3 Operational Restriction Event

➔ ***Abstract concept: Operational Restriction Event***
Operational Restriction Events are Operational Events specified in the Operational Plan Operational Restriction.
(also refer to RCA Domain Knowledge Specification, RCA.Doc.18)

Each Operational Restriction Event defines an action of an Operational Plan Operational Restriction and is of a certain type. It shall be complemented with a reference to the Operational Restriction Area to which it belongs to. Further, an Operational Restriction Event shall contain timing information at which it shall take place during execution of the Operational Plan Operational Restriction. Details on the mentioned characteristics of Operational Restriction Events are outlined in the following subsections.

The following Operational Restriction Events are defined:

Operational Event	Description
Start ORA Event	***Abstract Concept: Start ORA Event*** *The Start ORA Event describes the planned time at which an Operational Restriction Area and all its containing Operational Restrictions shall become active.* *(also refer to RCA Domain Knowledge Specification, RCA.Doc.18)*
End ORA Event	***Abstract Concept: End ORA Event*** *The End ORA Event describes the planned time at which an Operational Restriction Area and*

	all its containing Operational Restrictions shall become inactive. *(also refer to RCA Domain Knowledge Specification, RCA.Doc.18)*

Timing Information in Operational Restriction Events

Timing information at which the Operational Restriction Event shall take place during execution of the Operational Plan Operational Restriction are required. As for the Operation Movement Events there are Degrees of Freedom foreseen for the timing points of Operational Restriction Events.

4.1.4 Operational Plan Operational Warning Measure

➔ ***Abstract concept: Operational Plan Operational Warning Measure***
The Operational Plan Operational Warning Measure defines the parameters for the implementation of Operational Warning Area(s) and the Operational Warning Measure(s) therein.
(also refer to RCA Domain Knowledge Specification, RCA.Doc.18)

An Operational Plan Operational Warning Measure describes the temporal marking of a part of the railway network where approaching Physical Train Units triggers the activation of a Warning System. This is usually required on or near construction sites to protect Trackside Workers from dangerous situations.

Operational Warning Measures can either be created in the Planning System and provided as Operational Plan Operational Warning Measure to the RCA system, or in the RCA system. The latter are provided to the Planning System by the SubSys Plan Execution in the upstream (see section 5.5).

An Operational Warning Measure does not impact the Operational Movements. Infrastructure limitations, such as Temporary Speed Restrictions across the affected part of the railway network, shall be issued by an Operational Plan Operational Restriction (section 4.1.3). The Operational Restriction Area, however, can spatially overlap with the Operational Warning Area.

This chapter explains the abstract concept Operational Plan Operational Warning Measure in detail. Figure 8 shows the structure of the abstract concept Operational Plan Operational Warning Measure.

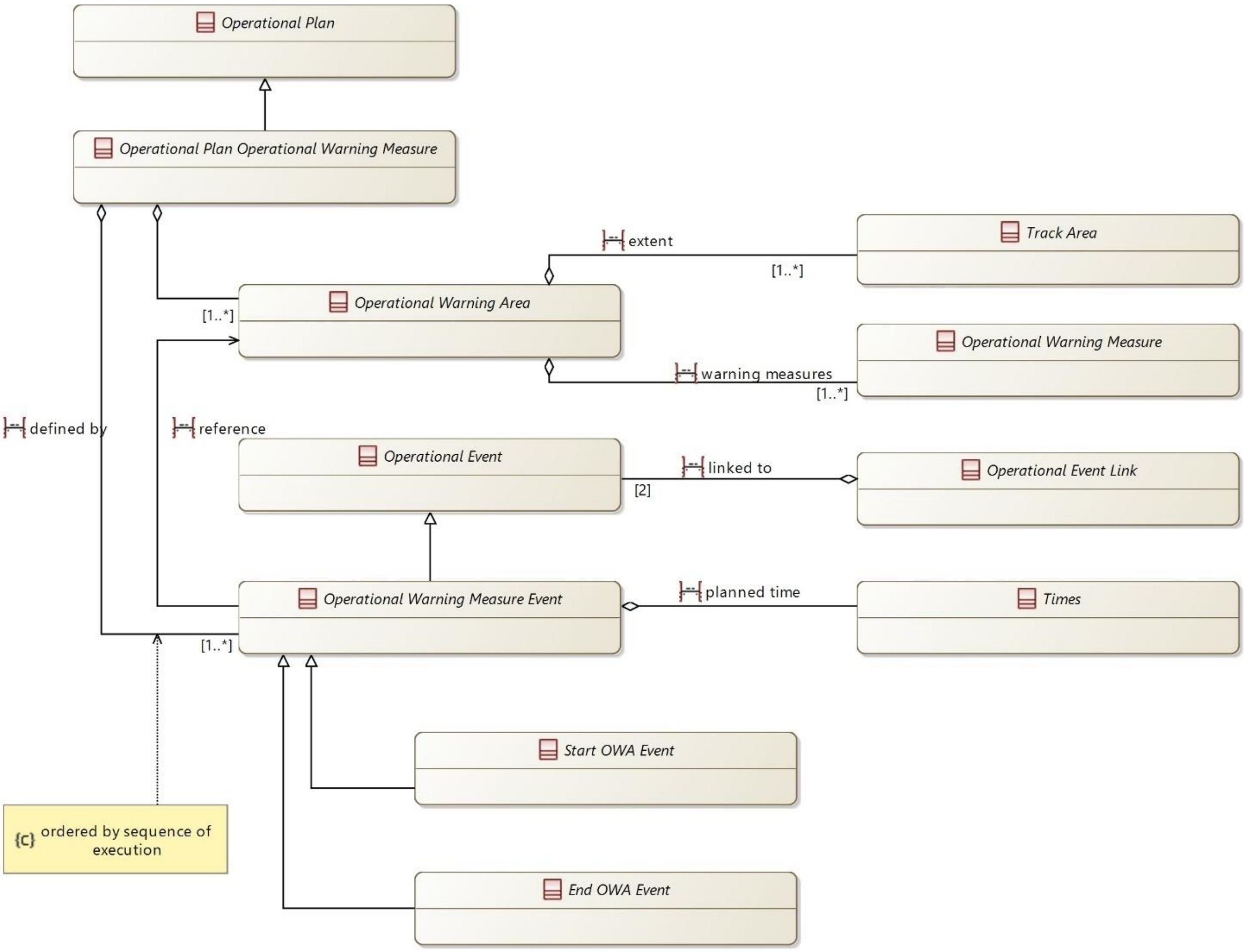


Figure 8: Structure of abstract concept Operational Plan Operational Warning Measure

The Operational Plan Operational Warning Measure has a unique identifier. The Operational Plan Operational Warning Measure contains one or more Operational Warning Areas (described in section 4.1.4.1), one or more Operational Warning Measures associated with the area (described in section 4.1.4.2), and one or more Operational Warning Measure Events (described in section 4.1.4.3). Each Operational Warning Measure Event refers to one Operational Warning Areas. Like all Operational Events, Operational Warning Measure Events can contain Operational Plan Event Links (described in section 4.1.1.2).

4.1.4.1 Operational Warning Area

➔ ***Abstract concept: Operational Warning Area***
The Operational Warning Area is an entity of the Operational Plan Operational Warning Measure and the Operating State. It describes the spatial dimensions of a planned or already implemented Operational Plan Operational Warning Measure within the Area of Control.
(also refer to RCA Domain Knowledge Specification, RCA.Doc.18)

The spatial dimensions of an Operational Warning Area are described as Track Edge Sections which provide the possibility to define a direction. Therefore, Operational Warning Measures which are valid in one or both directions of travel can be represented. An Operational Warning Area can contain one or more Track Edge Sections.

One or more Operational Warning Measures are assigned to an Operational Warning Area. These are described in the following subsection.

4.1.4.2 Operational Warning Measure

➔ ***Abstract concept: Operational Warning Measure***
An Operational Warning Measure is a measure to warn people about dangers-on the railway network.
(also refer to RCA Domain Knowledge Specification, RCA.Doc.18)

An Operational Warning Measure describes a type of warning measure within an Operational Warning Area. Operational Warning Measures are implemented by Automatic Warning Systems (AWS). An AWS consists of a set of collective Warning Devices and/or individual Warning Devices.

The available Operational Warning Measures depend on local safety regulations (defining, among others, early warning thresholds, warning duration, etc.), Warning Devices at hand, and their specific configuration.

Operational Warning Measures do not influence Operational Movements.

The precise definition of applicable Operational Warning Measures, their control, etc. is still under development and will be elaborated in further releases based on the outcomes from System Capabilities 65.1 to 65.4.

4.1.4.3 Operational Warning Measure Event

➔ ***Abstract concept: Operational Warning Measure Event***
Operational Warning Measure Events are Operational Events specified in the Operational Plan Operational Warning Measure
(also refer to RCA Domain Knowledge Specification, RCA.Doc.18)

Each Operational Warning Measure Event defines an action of an Operational Plan Operational Warning Measure and is of a certain type. It shall be complemented with a reference to the Operational Warning Area to which it belongs to. Further, an Operational Warning Measure Event shall contain timing information at which it shall take place during execution of the Operational Plan Operational Warning Measure. Details on the mentioned characteristics of Operational Warning Measure Events are outlined in the following subsections.

The following Operational Warning Measure Events are defined:

Operational Event	Description
Start OWA Event	***Abstract Concept: Start OWA Event*** *The Start OWA Event describes the planned temporal start of an Operational Warning Area.* *(also refer to RCA Domain Knowledge Specification, RCA.Doc.18)*
End OWA Event	***Abstract Concept: End OWA Event*** *The End OWA Event describes the planned temporal end of an Operational Warning Area.* *(also refer to RCA Domain Knowledge Specification, RCA.Doc.18)*

Timing Information in Operational Warning Measure Events

Timing information at which the Operational Restriction Event shall take place during execution of the Operational Plan Operational Restriction are required. As for the Operation Movement Events there is are Degrees of Freedom foreseen for the timing points of Operational Warning Measure Events.

4.2 Messages Downstream of Operational Plan Execution

The downstream messages sent from the Planning System are required by SubSys Plan Execution and SubSys ATO Execution in order to implement the requested Operational Plan. There is only one downstream message type on the interface SCI-OP, as depicted in Figure 9.

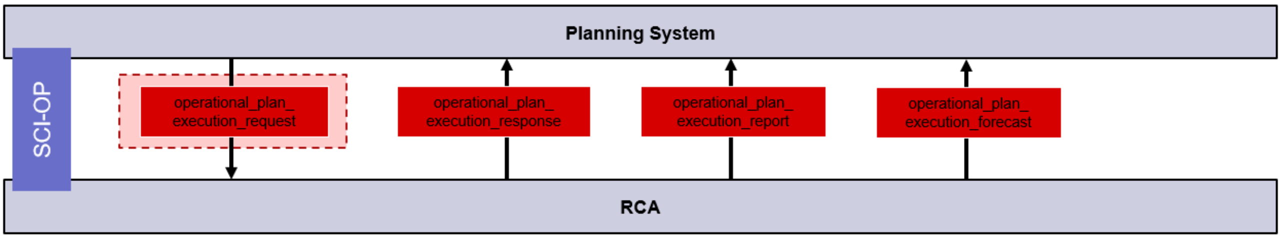


Figure 9: Downstream message Operational Plan Execution Request (in the bright red box) and, informatively, corresponding upstream messages.

4.2.1 Message: operational_plan_execution_request

Description:

This message is sent to request the execution of an Operational Plan (initial or update) by RCA system, in detail by SubSys Plan Execution and/or SubSys ATO Execution.

Attributes:

Name	Type	Multiplicity	Description
operationalPlan	OperationalPlan	1	Operational Plan to be executed by SubSys Plan Execution and/or SubSys ATO Execution

For further details on the message, please refer to details in chapter 0.

4.3 Messages Upstream of Operational Plan Execution

The upstream messages sent from the RCA System are required by the Planning System to detect deviations between Operational Plans and the Plan Execution as well as to identify upcoming or existing conflicts between Operational Plans and to develop appropriate countermeasures. The upstream for Operational Plan Execution consists of three message types, the operational_plan_execution_response, the operational_plan_execution_report and the operational_plan_execution_forecast, as illustrated in Figure 10.

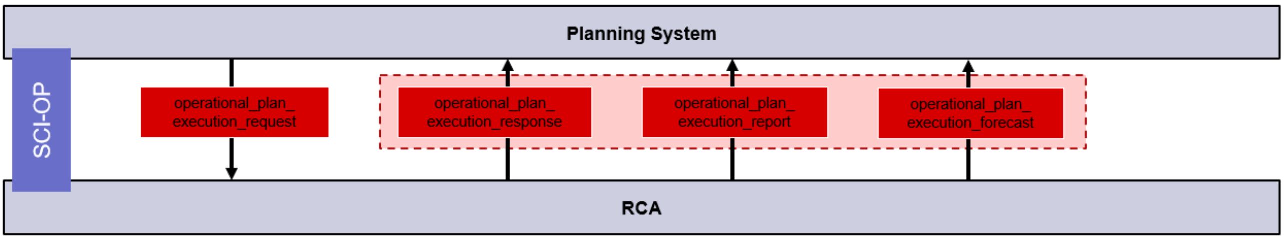


Figure 10: Upstream messages of the Operational Plan Execution (marked in the bright red box)

The operational_plan_execution_response provides the acceptance or rejection of an Operational Plan based on a feasibility check (details will be described in subsection 4.3.1.)

The operational_plan_execution_report provides the progress of the Operational Events of an Operational Plan during plan execution (details will be described in subsection 4.3.2.)

The operational_plan_execution_forecast provides the forecast of the Operational Movement Events of an Operational Plan Operational Movement during plan execution (details will be described in subsection 4.3.3)

The first two message types are issued for all three types of an Operational Plan, Operational Movement (described in section 4.1.2), Operational Restriction (described in section 4.1.3), and Operational Warning Measure (described in section 4.1.4) and are provided by both RCA SubSys Plan Execution and SubSys ATO Execution.

The last message type is issued only for Operational Plan Operational Movement (described in section 4.1.2) and is provided by RCA SubSys ATO Execution only.

4.3.1 Message: operational_plan_execution_response

Description:

This message is the response to an OperationalPlanExecutionRequest to report either the acceptance or the rejection of an Operational Plan. A rejection is only reported if the Operational Plan is not feasible.

Attributes:

Name	Type	Multiplicity	Description
updatedAt	DateTime	1	Timestamp when associated Operational Plan Execution Request was accepted or rejected
operationalPlanIdentifier	OperationalPlan-Identifier	1	Reference to the associated Operational Plan
rcaSubSys	Enum	1	Responding RCA SubSys. Possible values are: • “PE”: for RCA SubSys Plan Execution (PE) • “AE”: for RCA SubSys ATO Execution (AE)
returnCode	Enum	1	Code indicating the acceptance or the reason in case of a rejection. Possible values are: - 0: Execution Request of Operational Plan was accepted - <> 0: Execution Request of Operational Plan was rejected *The Operational Plan Feasibility Check and its details will be described here in a later version of this document:* - *Details of returnCode to be defined*

4.3.2 Message: operational_plan_execution_report

Description:

This message reports an update on the execution progress of a previously accepted Operational Plan. An operational_plan_execution_report shall be sent at each status change of any Operational Event for each accepted Operational Plan by the subsystems responsible for its implementation.

Each status change of an Operational Event issues a new Operational Plan Execution Report to the related Operational Plan. Accordingly, Operational Plan Execution Reports will be sent to the Planning System based on discrete events but not otherwise, e.g. by fixed time intervals.

With the discrete updates of Operational Events, the Operational Plan Execution Report does not provide a real-time or near-time picture of the operational situation. More detailed (near-time) updates on the progress of an Operational Movement will be provided via dedicated messages of the Operating State, that is to say the Train Unit Report (cf. section 5.1), the Track Allocation (cf. section 5.2) and the Field Element State (cf. section 5.3). Detailed updates on Operational Restrictions will be provided via dedicated messages of Operational Restriction Areas (cf. 5.4) and Operational Warning Areas (cf. 5.5).

The Operational Plan Execution Report will be returned by RCA subsystems implementing the Operational Plans. The subsystems will provide their own Operational Plan Execution Reports to the Planning System and the Planning System is responsible for combining the information provided together with the Operating State (see section 5), into a coherent image.

The Operational Plan Execution Report enables the SubSys ATO Execution and SubSys Plan Execution to inform the Planning System about existing conflicts. Additionally, it allows to inform the Planning System about the completion and forecasted delays concerning specific Operational Events.

Attributes:

Name	Type	Multiplicity	Description
operationalEventTime	OperationalEventTime	1..2	Timestamps when the activities within an Operational Event were updated by subsystem
operationalPlanIdentifier	OperationalPlanIdentifier	1	Reference to associated Operational Plan
operationalEventRef	GenericRef	1	Operational Event in referenced Operational Plan which changed its execution state
executionStatePE	ExecutionStatePE	0..1	Execution State of Operational Event by SubSys Plan Execution
executionStateAE	ExecutionStateAE	0..1	Execution State of Operational Event by SubSys ATO Execution

4.3.3 Message: operational_plan_execution_forecast

Description:

This message reports an execution forecast of Operational Movement Events of an Operational Plan Operational Movement. An operational_plan_execution_forecast shall be sent for each forecast update of Operational Movement Events for an accepted Operational Plan by the subsystems ATO Execution.

Attributes:

Name	Type	Multiplicity	Description
operationalEventTime	OperationalEventTime	1..2	Timestamps when the forecasts of the activities within an Operational Event were updated by subsystem
operationalPlanIdentifier	OperationalPlanIdentifier	1	Reference to associated Operational Plan
executionForecast	ExecutionForecast	1..*	Execution Forecast for Operational Movement Event(s) by SubSys ATO Execution

5 Operating State

➔ ***Abstract concept: Operating State***

The Operating State is the logical representation of the actual state of railway operations in the Area of Control. This includes for example the settings and states of Field Elements or the position of Physical Train Units.
(also refer to RCA Domain Knowledge Specification, RCA.Doc.18)

The knowledge about the Operating State enables the Planning System to keep current with the operational situation in the Area of Control and to recognise deviations from an Operational Plan during execution. Further, it allows for identifying upcoming or existing conflicts between Operational Plans and developing appropriate countermeasures. The messages of the Operating State from SubSys Plan Execution are embraced into five abstract concepts, as illustrated in Figure 11.

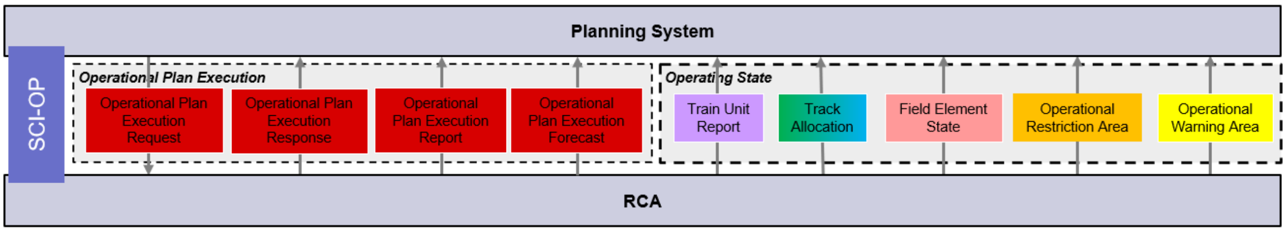


Figure 11: Main abstract concepts of SCI-OP with focus on the Operating State

The Operating State provides information of the positions and states of Physical Train Units, track occupancies and other limitations of the railway network, and the states of Field Elements. The messages are embraced in five groups. For every group, one or more message types exist. They are described in detail in the following sections.

5.1 Train Unit Report

➔ ***Abstract concept: Train Unit Report***

The Train Unit Report is an entity of the Operating State. It describes the position, properties, and state of any identified Physical Train Unit in the Area of Control.
(also refer to RCA Domain Knowledge Specification, RCA.Doc.18)

The Train Unit Report consists of four upstream messages as shown in Figure 12:

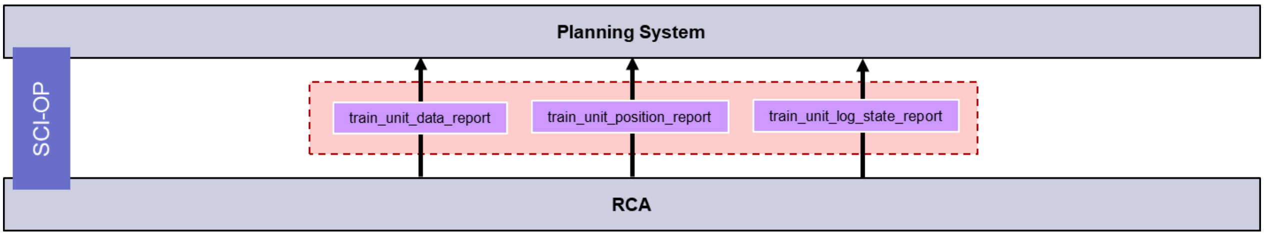


Figure 12: Upstream messages of Train Vehicle Report (marked in the red box)

The messages of a Train Unit Report shall include a reference to the corresponding Operational Plan, if it exists. A Train Unit Report message without reference to any Operational Plan also enables the Planning System to react on it, e.g. by issuing a suitable Operational Plan or by triggering actions for incident prevention.

The three message types of the Train Unit Report are described in the following subsections.

5.1.1 Message: train_unit_data_report

Description:

This message reports updates to the states of an identified Physical Train Unit. For each train_unit_data_report sent, the SubSys Plan Execution provides the reference to the corresponding Operational Plan, if available.

Attributes:

Name	Type	Multiplicity	Description
updatedAt	DateTime	1	Timestamp when the state dataset of the Physical Train Unit was updated.
operationalPlanIdentifier	OperationalPlanI-dentifier	0..1	Reference to associated Operational Plan, if available.
operationalTrainUnitRef	GenericRef	0..1	Reference to associated Operational Train Unit, if available
physicalTrainUnitIdentifier	PhysicalTrainUni-tIdentifier	1	Reference to associated Physical Train Unit.
trainUnitCharacteristics	TrainUnitCharacter-istics	1	Updated state dataset of a Physical Train Unit.

5.1.2 Message: train_unit_position_report

Description:

This message reports updates to the position, extent, and velocity of an identified Physical Train Unit. For each train_unit_position_report sent, the SubSys Plan Execution provides the reference to the corresponding Operational Plan, if available. This functionality does not provide high-precision positioning to the Planning System, since APS, as well as SubSys Plan Execution and the Planning System, shall consider the inaccuracy of the localisation technology used to safely represent the Physical Train Unit.

Attributes:

Name	Type	Multiplicity	Description
updatedAt	DateTime	1	Timestamp when the position dataset of the Physical Train Unit was updated.
operationalPlanIdentifier	OperationalPlanI-dentifier	0..1	Reference to associated Operational Plan, if available.
operationalTrainUnitRef	GenericRef	0..1	Reference to associated Operational Train Unit, if available
physicalTrainUnitIdentifier	PhysicalTrainUni-tIdentifier	1	Reference to associated Physical Train Unit.
trainUnitMovementData	TrainUnitMove-mentData	1	Updated position dataset of a Physical Train Unit

5.1.3 Message: train_unit_log_state_report

Description:

This message reports the logon or logoff and thus the identification of a Physical Train Unit to the Area of Control. For each train_unit_log_state_report sent, the SubSys Plan Execution provides the reference to the corresponding Operational Plan, if available. The message type provided for this purpose shall contain the following information:

Attributes:

Name	Type	Multiplicity	Description
updatedAt	DateTime	1	Timestamp when the Physical Train Unit was reported as logged on
operationalPlanIdentifier	OperationalPlan-Identifier	0..1	Reference to associated Operational Plan, if available
operationalTrainUnitRef	GenericRef	0..1	Reference to associated Operational Train Unit, if available
physicalTrainUnitIdentifier	Physical-TrainUnitIdenti-fier	1	Reference to associated Physical Train Unit
logState	binary	1	Determines the logon or the logoff of a Physical Train Unit Possible values: • 0: logoff • 1: logon

5.2 Track Allocation

➔ ***Abstract concept: Track Allocation***
The Track Allocation describes the assignment of an extent on the railway network to either a Movement Permission or an Unresolved Trackbound Movable Object.
(also refer to RCA Domain Knowledge Specification, RCA.Doc.18)

A Track Allocation describes the assignment of a fraction of the railway network, i.e. a Track Path, to either a Movement Permission or an Unresolved Trackbound Movable Object. In both cases, a Track Allocation denotes the temporal limitation of the railway network. Track Allocation consists of three upstream message types as shown in Figure 13.

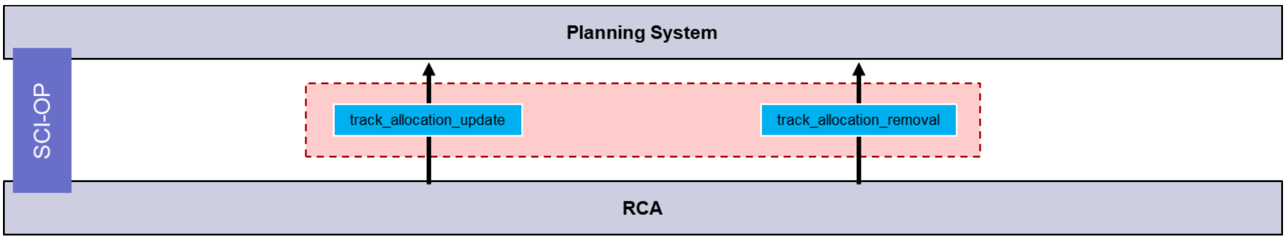


Figure 13: Upstream messages of Track Allocation (marked in the red box)

The Track Allocation provides the Planning System information about the usage of parts of the railway network. In the case of moving Train Units, the Track Allocations are frequently updated.

Messages of Track Allocations shall include a reference to the corresponding Operational Plan if it exists. A Track Allocation message without reference to any Operational Plan enables the Planning System to react on it, e.g. by issuing a suitable Operational Plan or by triggering actions at Incident Prevention Management.

5.2.1 Message: track_allocation_update

Description:

This message reports the reservation of a Track Edge(s) within the Area of Control due to either a Movement Permission or of an Unresolved Trackbound Movable Object. The message is used for creation and for updates of the same Track Allocation. The receiving Planning System shall realise that the first message of a Track Allocation denotes a creation, any subsequent message is an update. For each track_allocation_update sent, the SubSys Plan Execution provides the reference to the corresponding Operational Plan, if available.

Attributes:

Name	Type	Multiplicity	Description
updateTime	DateTime	1	Timestamp when Track Allocation was created or updated
operationalPlanIdentifier	OperationalPlan-Identifier	0..1	Reference to associated Operational Plan, if available
operationalTrainUnitRef	GenericRef	0..1	Reference to associated Operational Train Unit, if available
physicalTrainUnitIdentifier	Physical-TrainUnitIdenti-fier	0..1	Reference to associated Physical Train Unit, if available
trackAllocation	TrackAllocation	1	Describes the created Track Allocation

5.2.2 Message: track_allocation_removal

Description:

This message reports the release of a Track Edge(s) within the Area of Control due to either the removal of a Movement Permission or the disappearance of an Unresolved Trackbound Movable Object.

Attributes:

Name	Type	Multiplicity	Description
updateTime	DateTime	1	Timestamp when Track Allocation was removed
operationalPlanIdentifier	OperationalPlan-Identifier	0..1	Reference to associated Operational Plan, if available
trackAllocationRef	GenericRef	1	Reference to the removed Track Allocation

5.3 Field Element State

➔ **Abstract concept: Field Element State**
The Field Element State is an entity of the Operating State. It describes the state of any Field Element in the Area of Control.
(also refer to RCA Domain Knowledge Specification, RCA.Doc.18)

The Field Element State provides the states of Field Elements to the Planning System. For a point, for example, the position of the point blades is indicated. Each Field Element is uniquely identifiable by an ID and belongs to one of predefined Field Element Types (see section 6.9.2 for existing Field Element Types).

The Field Element State possesses only one upstream message: Field_element_state_update, as illustrated in Figure 14.

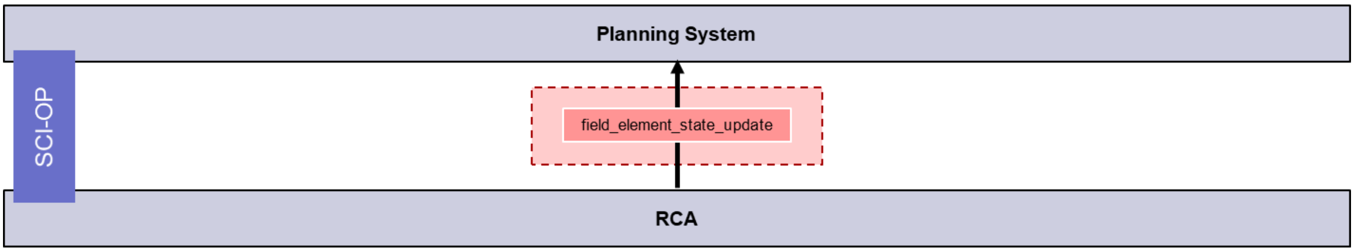


Figure 14: Upstream message of Field Element State (marked in the red box)

It is described in the following subsection.

5.3.1 Message: Field_element_state_update

Description:

This message reports an update of the states of a Field Element. For each Field_element_state_update sent, the SubSys Plan Execution provides the reference to the corresponding Operational Plan, if available.

Attributes:

Name	Type	Multiplicity	Description
updatedAt	DateTime	1	Timestamp when Field Element State was updated.
operationalPlanIdentifier	OperationalPlan-Identifier	0..1	Reference to associated Operational Plan, if available.
fieldElementIdentifier	FieldElemen-tIdentifier	1	Reference to updated Field Element.
fieldElementState	FieldElement-State	1	Type and state of updated Field Element.

5.4 Operational Restriction Area

An Operational Restriction limits the railway network within an Operational Restriction Area. It consists of two upstream message types, as shown in Figure 15.

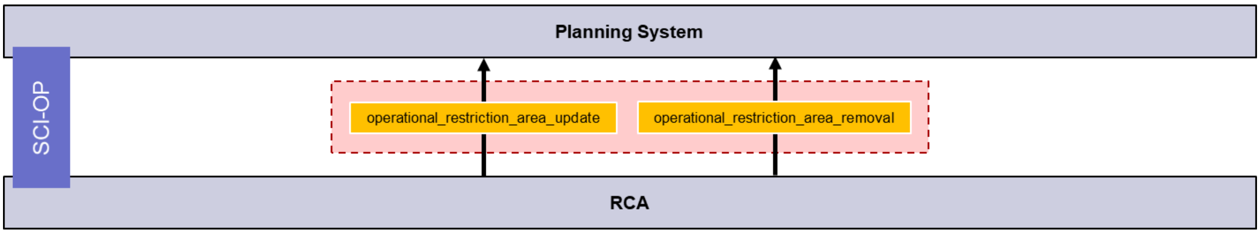


Figure 15: Upstream messages of Operational Restriction Area

Messages regarding Operational Restriction Areas are directly created as a reaction to messages safety issues and are fed by messages of APS concerning Usage Restriction Areas. Usage Restriction Areas are intentionally not to be modified, both, modification and removal are assumed to be safety-critical measures and are therefore limited to what is most necessary. This means that only upstream messages for creation and removal of Operational Restriction Areas are defined.

Messages of Operational Restriction Area shall include a reference to the corresponding Operational Plan if it exists. An Operational Restriction Area message without reference to any Operational Plan enables the Planning System to react on it, e.g. by altering Operational Plans affected by it. The two message types of the Operational Restriction Area are described in the following subsections.

5.4.1 Message: operational_restriction_area_update

Description:

This message reports the creation of an Operational Restriction Area. For each operational_restriction_area_update sent, the SubSys Plan Execution provides the reference to the corresponding Operational Plan, if available. The moment of the effective Operational Restriction Area creation can deviate from the planned creation of the corresponding Operational Restriction Area in the Planning System. This is useful information to identify differences between the planned and the effective start of an Operational Restriction Area.

Attributes:

Name	Type	Multiplicity	Description
operationalRestrictionArea	OperationalRestriction-Area	1	Operational Restrictions Area which was created
operationalPlanIdentifier	OperationalPlanIdentifier	0..1	Reference to associated Operational Plan, if available
operationalRestriction-EventRef	GenericRef	0..1	Reference to associated Operational Restriction Event, if available
plannedRemovalTime	DateTime	0..1	Timestamp at which the Operational Restriction Area is planned/forecasted to be removed
updatedAt	DateTime	1	Timestamp at which the Operational Restriction Area was created

5.4.2 Message: operational_restriction_area_removal

Description:

This message reports the removal of an Operational Restriction Area. For each operational_restriction_area_removal sent, SubSys Plan Execution provides the reference to the corresponding Operational Plan, if available. The moment of the effective Operational Restriction Area removal can deviate from the planned removal of the corresponding Operational Restriction Area in the Planning System. This is useful information to identify differences between the planned and the effective end of an Operational Restriction Area.

Attributes:

Name	Type	Multiplicity	Description
operationalRestriction-AreaRef	GenericRef	1	Reference to the removed Operational Restriction Area
operationalPlanIdentifier	OperationalPlanIdentifier	0..1	Reference to associated Operational Plan, if available
operationalRestriction-EventRef	GenericRef	0..1	Reference to associated Operational Restriction Event, if available
updatedAt	DateTime	1	Timestamp when the Operational Restriction Area was removed

5.5 Operational Warning Measure Area

The Operational Warning Area describes an area on the railway network in which driving trains can potentially impose danger, to Authorised Trackside Persons (e.g. to infrastructure maintenance staff). It consists of two upstream message types, as shown in Figure 16.

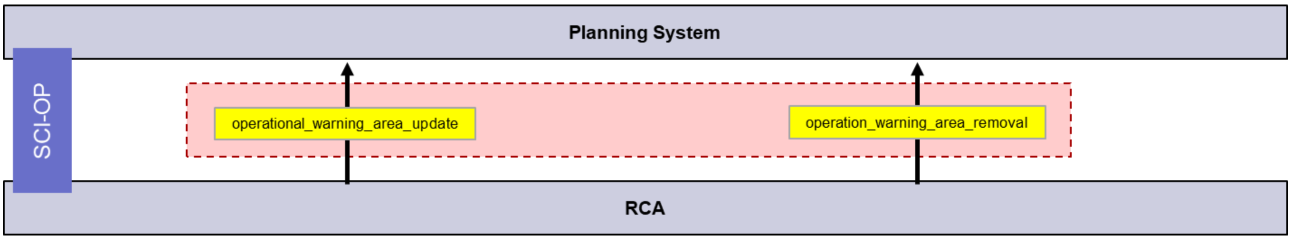


Figure 16: Upstream messages of Operational Warning Area (marked in the red box)

Messages regarding Operational Warning Areas are directly fed from Warning Areas of APS. Messages of Operational Warning Area shall include a reference to the corresponding Operational Plan if it exists. An Operational Warning Area message without reference to any Operational Plan enables the Planning System to react on it, e.g. by creating a new Operational Warning Area. The two message types of the Operational Warning Area are described in the following subsections.

5.5.1 Message: operational_warning_area_update

Description:

This message reports the creation or the update of an Operational Warning Area. For each operational_warning_area_update sent, the SubSys Plan Execution provides the reference to the corresponding Operational Plan, if available.

An Operational Warning Area has two types of states:

- Activation State: Active (can warn) / Inactive (cannot warn)
- Warning State: When Activation State = Active: Is Warning / Is Not Warning

The operational_warning_area_update reports the Activation State of an Operational Warning Area. It was determined that the Warning State is not required by the Planning System and shall therefore not be delivered in the upstream.

Attributes:

Name	Type	Multiplicity	Description
operationalPlanIdentifier	OperationalPlanIdentifier	0..1	Reference to associated Operational Plan, if available
operationalWarning-MeasureEventRef	GenericRef	0..1	Reference to associated Operational Warning Measure Event, if available
plannedRemovalTime	DateTime	0..1	Timestamp at which the Operational Warning Area is planned/forecasted to be removed
updatedAt	DateTime	1	Timestamp when the Operational Warning Area was created or updated
operationalWarningArea	OperationalWarningArea	1	Operational Warning Area which was created or updated incl. the current Activation State of the Operational Warning Area

5.5.2 Message: operational_warning_area_removal

Description:

This message reports the removal of an Operational Warning Area. For each operational_warning_area_removal sent, the SubSys Plan Execution provides the reference to the corresponding Operational Plan, if available. An operational_warning_area_removal message shall contain the following information:

Attributes:

Name	Type	Multiplicity	Description
operationalWarningAreaRef	GenericRef	1	Reference to the removed Operational Warning Area
operationalPlanIdentifier	OperationalPlanIdentifier	0..1	Reference to associated Operational Plan, if available
operationalWarningMeasureEventRef	GenericRef	0..1	Reference to associated Operational Warning Measure Event, if available
updatedAt	DateTime	1	Timestamp when the Operational Warning Area was removed

The moment of the effective Warning Area removal can deviate from the planned removal of the corresponding Operational Warning Area in the Planning System. This is useful information to identify differences between the planned and the effective end of an Operational Warning Area.

6 Data Model

This chapter describes the defined data model for the implementation of abstract concepts and messages which have been described in chapters 4 and 5.

6.1 Operational Plan

6.1.1 Abstract Class: OperationalPlan

Description:

This abstract class shall be realised by one of the following classes:

- OperationalMovementPlan
- OperationalRestrictionPlan
- OperationalWarningMeasurePlan

6.1.2 Class: OperationalEventLink

Description:

This class describes the dependency between the origin Operational Event and any other Operational Event of the same or another Operational Plan.

Attributes:

Name	Type	Multiplicity	Description	Reason
type	Enum	1	Describes the type of dependency between the two Operational Events. Possible values are: • "IS_WAITING_FOR": The Operational Event cannot be processed before another Operational Event has been successfully processed. • "IS_EXPECTED_BY": The Operational Event shall be successfully processed before another Operational Event can be processed.	For chronological triggering of requests via SCI-CMD. For chronological triggering of requests via AoE (ATO over ETCS) SS-131.
operationalPlanRef	Operational-PlanIdentifier	1	Reference to the Operational Plan which contains the dependent Operational Event	For implementation and triggering of requests via SCI-CMD.
operationalEventRef	GenericRef	1	Reference to the dependent Operational Event	For implementation and triggering of requests via SCI-CMD.

Example:

```
"operationalEventLink": {
    "type": "IsWaitingFor",
    "operationalPlanRef": "UUID-OP-2"
    "operationalEventRef": "UUID-OE-23"
}
```

6.1.3 Class: OperationalEventTime

Description:

This class contains the timing information of an Operational Event with its scope.

Attributes:

Name	Type	Multi-plicity	Description	Reason
scope	Enum	1	An Operational Event can have several timings. This attribute determines the scope of one timing. For Operational Movement Events, the following values are possible: • “EARLIEST_OPERATIONAL_DEPARTURE”: The Physical Train Unit shall leave from or pass the Operational Movement Event (type: “stop” or “passage”) not earlier than at the given date time. • “LATEST_OPERATIONAL_DEPARTURE”: The Physical Train Unit shall depart from or pass the Operational Movement Event (type: “stop” or “passage”) not later than at the given date time. • “EARLIEST_OPERATIONAL_ARRIVAL”: The Physical Train Unit shall arrive at the Operational Movement Event (type: “stop” not earlier than at the given date time. • “LATEST_OPERATIONAL_ARRIVAL”: The Physical Train Unit shall arrive at the Operational Movement Event (type: “stop”) not later than at the given date time. • OPERATIONAL_ARRIVAL: Time scheduled by PAS for the arrival of the Physical Train Unit. The time shall be between or equal to EARLIEST_ARRIVAL and LATEST_ARRIVAL (used by SubSys ATO Execution in downstream). • OPERATIONAL_DEPARTURE: Time scheduled set by PAS for the departure of the Physical Train Unit. The time shall be between or equal to EARLIEST_DEPARTURE and LATEST_DEPARTURE (used by SubSys ATO Execution in downstream). • PUBLISHED_ARRIVAL: Train arrival time published in the timetable (used by SubSys ATO Execution in downstream). • PUBLISHED_DEPARTURE: Train departure time published in the timetable (used by SubSys ATO Execution in downstream)	For implementation and triggering of requests via SCI-CMD. For implementation and triggering of requests via AoE SS-131.

			• "REALISED_ARRIVAL": DateTime provided in upstream by SubSys ATO Execution when Physical Train Unit arrived • "REALISED_DEPARTURE": DateTime provided in upstream by SubSys ATO Execution when Physical Train Unit departed For Operational Restriction Events and Operational Warning Events, the following values are possible: • "CREATION": The Operational Restriction Area or Operational Warning Area shall be created at the given DateTime • "REMOVAL": The Operational Restriction Area or Operational Warning Area shall be removed at the given DateTime	
Time	DateTime	1	Timestamp at which the respective Operational Event shall be executed	For implementation and triggering of requests via SCI-CMD.

Example for Operation Movement Event:

```
"times": [
  {
    "scope": "EARLIEST_ARRIVAL",
    "dateTime": "2020-03-11 03:34:00 PM"
  },
  {
    "scope": "LATEST_ARRIVAL",
    "dateTime": "2020-03-11 03:25:00 PM"
  }
],
```

6.1.4 Class: OperationalPlanIdentifier

Description:

This class describes the identification parameters of an Operational Plan.

Attributes:

Name	Type	Multiplicity	Description	Reason
uuid	GenericUuid	1	Unique identifier of a Track Allocation for the communication across systems.	Each Operational Plan shall be uniquely identifiable
version	Enum	1	Version of an Operational Plan issued by the Planning System	When an Operational Plan will be updated, it must be clearly apparent which is the latest version of this Plan.
name	String	1	Human-readable identifier of the Operational Plan (not necessarily unique)	Identifier for manual control

6.2 Operational Plan Operational Movement

6.2.1 Class: OperationalPlanOperationalMovement

Description:

This class defines the implementation of an Operational Movement.

Attributes:

Name	Type	Multi-plicity	Description	Reason
id	GenericId	1	File-wide unique, machine-interpretable identifier.	Unique identifier within the transmitted data set
uuid	GenericUuid	1	Universally unique identifier used for identification of the object across systems	Unique identifier for the communication across systems
versionId	GenericId	1	Current version of Operational Plan	Maintain the current version of Operational Plan in case of updates
mapDataVersionRef	GenericRef	1	Reference to the associated Map Data version used by the Operation Plan	Feasibility check of the requested Operational Plan
operationalSegments	OperationalSegment	1..*	Sequence of Operational Train Units used in the Operational Plan	Identification of different Train Units during Operational Movement
operationalTrainUnits	Operational-TrainUnit	1..*	List of Operational Train Units belonging to the Operational Plan Operational Movement	Characteristics of Operational Train Units are required for implementation and triggering of requests via SCI-CMD
operationalConsists	OperationalConsist	1..*	List of Operational Consists belonging to the Operational Plan Operational Movement	Characteristics of Operational Consists are required for implementation and triggering of requests via SCI-CMD
operationalMovementEvents	OperationalMovementEvent	1..*	List of Operational Movement Events to be implemented	Characteristics of Operational Movement Events are required for implementation and triggering of requests via SCI-CMD

Example:

```
"operationalPlanOperationalMovement": [
  {
    "id": "ID-OPOM-1",
    "uuid": "UUID-OPOM-1",
    "versionId": "VID-OPOM-1",
    "mapDataVersionRef": "ID-INST-1",
    "operationalSegments": [],
    "operationalTrainUnits": [],
```

```
    "operationalConsists": [],
    "operationalMovementEvents": [],
  }
],
```

6.2.2 Class: OperationalSegment

Description:

This class describes the order of Operational Segments within an Operational Plan. Each Operational Segment has a reference to a valid Operational Train Unit.

Attributes:

Name	Type	Multiplicity	Description	Reason
id	GenericId	1	File-wide unique, machine-interpretable identifier.	Unique identifier within the transmitted data set
uuid	GenericUuid	1	Universally unique identifier used for identification of the object across systems	Unique identifier for the communication across systems
sequence	PositiveCounter	1	Sequence of successive Operational Train Units along the Track Path	-
operationalTrainUnitRef	GenericRef	1	Reference to Operational Train Unit	-

Example:

```
"operationalSegments": [
  {
    "id": "ID-OS-1",
    "uuid": "UUID-OS-1",
    "sequence": 1,
    "operationalTrainUnitRef": "ID-OTU-1"
  }
]
```

6.2.3 Class: OperationalTrainUnit

Description:

This class describes operational characteristics of an Operational Train Unit.

Attributes:

Name	Type	Multiplicity	Description	Reason
id	GenericId	1	File-wide unique, machine-interpretable identifier.	Unique identifier within the transmitted data set
uuid	GenericUuid	1	Universally unique identifier used for identification of the object across systems	Unique identifier for the communication across systems
physicalTrainUnitIdentifier	PhysicalTrainUnitIdentifier	1	Reference to associated Physical Train Unit	activeOnBoardUnitId: Information shall be available in SubSys Plan Execution to uniquely identify the

				relevant Physical Train Unit in the Operating State provided via SCI-CMD. trainNumber/additionalTrainNumber/tafTapTsiTrainID: Information shall be available in SubSys Plan Execution user interface of Workbench (fallback level for PAS)
operationalTrainCategory	Operational-TrainCategory	0..1	Description of the train category.	Information shall be available in SubSys Plan Execution user interface of Workbench (fallback level for PAS).
operationalConsistRefs	Operational-ConsistRef	1..*	List of references to Operational Consists belonging to the same Operational Train Unit.	Characteristics of Operational Consists are required for implementation and triggering of requests via SCI-CMD.
brakeWeightPercentage	Brake-WeightPercentage	0..1	Definition of the brake system for the whole Physical Train Unit as a "fixed formation".	For implementation and triggering of requests via SCI-CMD (Movement Permission).

Example:

```
"operationalTrainUnits": [
  {
    "id": "ID-OTU-1",
    "uuid": "UUID-OTU-1",
    "physicalTrainUnitIdentifier": {},
    "operationalTrainCategory ": {},
    "operationalConsistRefs": []
    "brakeWeightPercentage": {}
  }
]
```

6.2.4 Class: OperationalTrainCategory

Description:

This class describes the train category of an Operational Train Unit.

Attributes:

Name	Type	Multiplicity	Description	Reason
id	GenericId	1	File-wide unique, machine-interpretable identifier.	Unique identifier within the transmitted data set
uuid	GenericUuid	1	Universally unique identifier used for identification of the object across systems	Unique identifier for the communication across systems

name	String	0..1	Established, human-readable short string, giving the object a name. Not intended for machine interpretation. Examples: • Regionalbahn • Regionaltog • Snabbtåg • Intercity • Frecciarossa • Railjet • Eurostar • ...	Information shall be available in SubSys Plan Execution user interface of Workbench (fallback level for PAS)
description	String	0..1	Human-readable, more detailed description as addition to the name. Not intended for machine interpretation.	Information shall be available in SubSys Plan Execution user interface of Workbench (fallback level for PAS)
trainUsage	Enum	0..1	Purpose of the train journey for service, representation, and information applications. Possible values are: • COMMERCIAL_PASSENGER_TRAIN • COMMERCIAL_CARGO_TRAIN • LIGHT_RUN (LEERFAHRT) • LOCOMOTIVE_RUNNING_LIGHT (LOKLEERFAHRT) • ENGINEERING_TRAIN • BREAKDOWN_TRAIN (HILFSZUG)	Information shall be available in SubSys Plan Execution user interface of Workbench (fallback level for PAS). It is not intended to pass this information to other SubSys.

Example:

```
"operationalTrainCategory": {
    "id": "ID-CAT-1",
    "uuid": "UUID-CAT-1",
    "name": "RegionalTrain",
    "description": "",
    "trainUsage": "COMMERCIAL_PASSENGER_TRAIN"
}
```

6.2.5 Class: OperationalConsistRef

Description:

The class references an Operational Consist and defines its position in the Operational Train Unit.

Attributes:

Name	Type	Multiplicity	Description	Reason
operationalConsistRef	GenericRef	1	Reference to the associated Operational Consist	-

position	Integer	1	Position of the Operational Consist in the Operational Train Unit	-

Example:

```
"operationalConsistRef": [
  {
    "operationalConsistRef": "ID-OC-11",
    "position": 1
  }
]
```

6.2.6 Class: BrakeWeightPercentage

Description:

This class defines the braked weight percentages for the whole Physical Train Unit as a “fixed formation”.

Attributes:

Name	Type	Multiplicity	Description	Reason
regularBrakeWeightPercentage	Percentage	0..1	Brake weight percentage related to brake mass for normal brake operations of non-automatic brakes	For implementation and triggering of requests via SCI-CMD (Movement Permission)
emergencyBrakeWeightPercentage	Percentage	0..1	Brake weight percentage related to brake mass for emergency brake operations, differs from regular brake percentage depending on auxiliary brakes	For implementation and triggering of requests via SCI-CMD (Movement Permission)

Example:

```
"brakeWeightPercentage": {
  "regularBrakeWeightPercentage": 150,
  "emergencyWeightBrakePercentage": 200
}
```

6.2.7 Class: OperationalConsist

Description:

This class describes the characteristics of an Operational Consist.

Attributes:

Name	Type	Multiplicity	Description	Reason
id	GenericId	1	File-wide unique, machine-interpretable identifier	Unique identifier within the transmitted data set
uuid	GenericUuid	1	Universally unique identifier used for identification of the object across systems	Unique identifier for the communication across systems
supportedOnBoardEquipment	SupportedOnBoardEquipment	1	List of the on-board equipment supported by the Consist, such as safety control system	For implementation and triggering of MP requests via SCI-CMD

Example:

```
"operationalConsist": [
  {
    "id": "ID-OC-11",
    "uuid": "UUID-OC-11",
    "supportedOnBoardEquipment": []
  }
]
```

6.2.8 Class: SupportedOnBoardEquipment

Description:

This class describes the technical on-board equipment supported by an Operational Consist such as the installed safety control system equipment or ATO-related equipment. It is provided to the RCA System in an Operational Plan Operational Movement.

Attributes:

Name	Type	Multiplicity	Description	Reason
type	Enum	1	This specifies the type of the train protection system using an enumeration to have unique names: Possible values are: • "ETCS" • Further types to be defined	For implementation and triggering of requests via SCI-CMD (MP-Request)
etcsSystemVersions	ETCS-SystemVersion	1..*	Delivers the list of ETCS System Versions supported by the Operational Consist	For implementation and triggering of requests via SCI-CMD (MP-Request)
anyClass	AnyClass	0..*	Defines an extension point for a non-SCI-OP class	Extension point for e.g. for further used on-board equipment

Example:

```
"supportedOnBoardEquipment": [
  {
    "type": "ETCS",
    "etcsSystemVersions": []
  }
]
```

6.2.9 Class: ETCSSystemVersion

Description:

This class describes the ETCS system version.

Attributes:

Name	Type	Multiplicity	Description	Reason
etcsSystemVersion	Enum	1	ETCS System Version. Possible values are:	For implementation and triggering of requests via SCI-CMD (MP-Request)

			• "1.0" • "1.1" • "2.0"	

Example:

```
"etcsSystemVersions": [
  {
    "etcsSystemVersion": "1.0"
  },
  {
    "etcsSystemVersion": "1.1"
  },
  {
    "etcsSystemVersion": "2.0"
  }
],
```

6.2.10 Class: OperationalMovementEvent

Description:

This class describes the sequence number, position and characteristics of an Operational Movement Event.

Attributes:

Name	Type	Multiplicity	Description	Reason
id	GenericId	1	File-wide unique, machine-interpretable identifier.	Unique identifier within the transmitted data set.
uuid	GenericUuid	1	Universally unique identifier used for identification of the object across systems.	Unique identifier for the communication across systems.
sequence	Integer	1	Defines the chronological order of the Operational Movement Events inside the Operational Plan.	To execute the Operation Plan in the correct order.
type	Enum	1	Describes the planned action of the Physical Train Unit at this Operational Movement Event. Possible values are: • "STOP": The Physical Train Unit shall stop at this event. • "PASSAGE": The Physical Train Unit shall not stop at this event. • "OTHER": This is a placeholder for IM-specific events. Please check with section 4.1.2.2.	For implementation and triggering of requests via SCI-CMD.
alignment	Enum	1	Describes the alignment of the arriving train in relation to the place on a track referenced by	For implementation and triggering of requests via SCI-CMD.

			Operational Movement Event. Possible values are: • "HEAD": The head of the train (front of first vehicle) shall be located at the position specified by the Operational Movement Event. • "CENTER": The center of the train (mid point concerning the train length) shall be located at the position specified by the Operational Movement Event. • "REAR": The end of the train (back of last vehicle) shall be located at the position specified by the Operational Movement Event.	
position	OperationalMovementEventPosition	1	Describes the position referenced by the Operational Movement Event.	For implementation and triggering of requests via SCI-CMD.
operationalEventTimes	OperationalEventTime	2, 4	Describes the arrival and departure time and day with their scope.	For implementation and triggering of requests via SCI-CMD.
operationalEventLinks	OperationalEventLink	0..*	List of dependencies between this Operational Event and any other Operational Event of the same or another Operational Plan.	For implementation and triggering of requests via SCI-CMD.
trackPathToSubsequentEvent	TrackEdgeSection	0..*	Describes the Track Path from one Operational Movement Event to the subsequent one, by a sorted list of Track Edge Sections lying between the specified event and the subsequent one. The Track Edges referenced by the TrackEdgePosition of the two events are excluded from the list.	For implementation and triggering of requests via SCI-CMD.
stopDescription	StopDescription	0..1	Describes the characteristics and activities to be carried out at planned stop of a Physical Train Unit	For implementation and triggering of requests via SCI-CMD.

				For implementation and triggering of requests via AoE SS-131.

Example:

```
"operationalMovementEvents": [
  {
    "id": "ID-OME-11",
    "uuid": "UUID-OME-11",
    "sequence": 1,
    "toBeExecuted": true,
    "type": "STOP",
    "alignment": "HEAD",
    "operationalMovementEventPosition": {},
    "operationalEventTimes": [],
    "operationalEventLinks":[],
    "trackPathToSubsequentEvent":[
      {
        "trackEdgeRef": "648390da",
        "direction": "UP"
      },
      {
        "trackEdgeRef": "6485183d",
        "direction": "DOWN"
      }
    ],
    "stopDescription": {}
  }
]
```

6.2.11 Class: OperationalMovementEventPosition

Description:

This class describes the position referenced by the Operational Movement Event.

Attributes:

Name	Type	Multiplicity	Description	Reason
type	Enum	1	Type of position Possible values are: • "LOGICAL_POSITION": used if physical position of the Operational Movement Event is not known to SubSys Plan Execution. • "POSITION_ON_TRACK": used if physical position of the Operational Movement Event is known to SubSys Plan Execution. If the type is "POSITION_ON_TRACK", the attribute trackEdgePosition is mandatory.	For implementation and triggering of requests via SCI-CMD. The type "logicalPosition" might be required if the physical position of the Operational Movement Event is not known to SubSys Plan Execution, but there are temporal dependencies to this event. (e.g. connecting train in foreign country).

trackEdgePosition	TrackEdgePosition	0..1	Reference to the associated Track Edge Position	For implementation and triggering of requests via SCI-CMD.

Example:

```
"operationalMovementEventPosition": {
  "type": "POSITION_ON_TRACK",
  "trackEdgePosition": {
    "trackEdgeRef": "6485183d",
    "offset": 127.000
  }
}
```

6.2.12 Class: StopDescription

Description:

This class describes the characteristics and activities to be carried out at planned stop of a Physical Train Unit.

Attributes:

Name	Type	Multiplicity	Description	Reason
commercial	Boolean	1	This characterizes the reason for the stop. It can either be “commercial” or “operational”. In case of a passenger train, a commercial stop would be a stop, where passengers can enter or alight from the train. An operational stop might for example be used for changing the staff	For implementation and triggering of requests via AoE SS-131.
stopOnRequest	Boolean	1	This indicates a commercial stop, depending on a certain demand, such as passengers or goods waiting for the train to stop. A stop on request is a special kind of a commercial stop. This attribute is to be defined only if commercial="true".	For implementation and triggering of requests via AoE SS-131.
onOff	Enum	1	This describes, if passengers/goods are not supposed to enter or leave the train if it is a commercial stop (commercial="true"). The attribute is used for passenger information and with the operational background to encode whether a train may possibly depart before the scheduled time. Concerning semantical constraints, see below. Possible values are:	For implementation and triggering of requests via AoE SS-131.

			• "BOTH": Train stops to alight as well as to be entered. This is the normal case for most of the commercial stops and therefore the default value. • "ON": Train stops to be entered only or to receive goods only but not to alight/unload. • "OFF": Train stops to alight from the train only or to unload goods but not to be entered/receive.	
stopTimes	StopTimes	0..1	Composition of the stopping durations a Physical Train Unit spends at a planned stop	For implementation and triggering of requests via SCI-CMD. For implementation and triggering of requests via AoE SS-131.
stopActivities	StopActivity	0..*	List of stop activities. They describe activities to be carried out at planned stop of a Physical Train Unit.	For implementation and triggering of requests via AoE SS-131.
platformEdgeRefs	PlatformEdgeRef	0..*	List of platform edges used by the Physical Train Unit. In special cases, there could occur more than one platform edge.	For implementation and triggering of requests via AoE SS-131.

Example:

```
"stopDescription": {
  "commercial": true,
  "stopOnRequest": true,
  "onOff": "BOTH",
  "stopTimes": {},
  "stopActivities": [],
  "platformEdgeRefs": []
}
```

6.2.13 Class: StopTimes

Description:

This class describes the set of components of the time period between arrival and departure of a Physical Train Unit at a planned stop. It should contain: minimalTime + operationalReserve + additionalReserve.

Attributes:

Name	Type	Multiplicity	Description	Reason
minimalTime	Duration	0..1	Minimal stop time.	For implementation and triggering of requests via SCI-CMD.

				For implementation and triggering of requests via AoE SS-131.
operationalReserve	Duration	0..1	Additional stop time for operational purposes	For implementation and triggering of requests via SCI-CMD. For implementation and triggering of requests via AoE SS-131
additionalReserve	Duration	0..1	Additional unspecified stop time, e.g. waiting for a connection	For implementation and triggering of requests via SCI-CMD. For implementation and triggering of requests via AoE SS-131
anyAttribute	AnyAttribute	0..1	Defines an extension point for a non-SCI-OP attribute	Extension point for e.g. for further stop times

Example:

```
"stopTimes": {
  "minimalTime": "00:01:00"
  "operationalReserve": "00:00:30"
  "additionalReserve": "00:00:30"
}
```

6.2.14 Class: StopActivity

Description:

This class describes an activity carried out at a stop.

Attributes:

Name	Type	Multiplicity	Description	Reason

type	Enum	1	Describes the activity carried out at this planned stop. Possible Values are: • "JOIN": Couple vehicles / train parts - intended for self-propelling train parts. Please consider relation to formations (as far as used) • "SPLIT": Uncouple vehicles / train parts - intended for self-propelling train parts. Please consider relation to formations (as far as used) • "COLLECT": Couple vehicles / train parts - intended for non-self-propelling train parts. Please consider relation to formations (as far as used) • "DROP": Uncouple vehicles / train parts - intended for non-self-propelling train parts. Please consider relation to formations (as far as used) • "REVERSE": Stop to change driving direction of train • "STABLE": Notes the necessity of stabling track(s) during the stop time	For implementation and triggering of requests via SCI-CMD. "STABLE": ATO could use the information for shutting down the PTU. PE could release a too long MP.
actualOperationalSegmentRef	GenericRef	0..1	Reference to associated Operational Segment which is valid before the Stop Activity	For implementation and triggering of requests via SCI-CMD.
targetOperationalSegmentRef	GenericRef	0..1	Reference to the associated Operational Segment which will be valid after the Stop Activity	For implementation and triggering of requests via SCI-CMD.

Example:

```
"stopActivities": [
  {
    "type": "split",
    "actualOperationalSegmentRef": "ID-OS-1",
    "targetOperationalSegmentRef": "ID-OS-2"
  }
]
```

6.2.15 Class: PlatformEdgeRef

Description:

This class references the platform edge element to be used by the Train Unit at the stop. In special cases, there could occur more than one platform edge.

Attributes:

Name	Type	Multiplicity	Description	Reason
platformEdgeRef	GenericRef	1	Reference to the associated Platform Edge provided by the version of Map Data currently in use.	For implementation and triggering of requests via AoE SS-131 (automatic door opening and closing).

Example:

```
"platformEdgeRefs": [
  {
    "platformEdgeRef": "UUID-PE-1",
  }
]
```

6.2.16 Class: PhysicalTrainUnitIdentifier

Description:

This class describes the identification parameters of a Physical Train Unit.

Attributes:

Name	Type	Multiplicity	Description	Reason
trainNumber	String	0..1	Operational identification of the Physical Train Unit (please note that it has not necessarily to be unique)	Information shall be available in SubSys Plan Execution user interface of Workbench (fallback level for PAS)
additionalTrainNumber	String	0..1	Distinction between Operational Train Units with the same trainNumber. This may occur if a trainNumber is reused for different spare trains.	Information shall be available in SubSys Plan Execution user interface of Workbench (fallback level for PAS)
tafTapTsiTrainID	String	0..1	Unique TrainID, used for the explicit identification of a Physical Train Unit, especially for international train runs. The number will be implemented during the implementation of the TAF/TAP TSI.	Information shall be available in SubSys Plan Execution user interface of Workbench (fallback level for PAS)
activeOnBoardUnitId	String	0..1	Reference to active OBU of Physical Train Unit	Information shall be available in to uniquely identify the relevant Physical Train Unit in the Operating State provided via SCI-CMD.

Example:

```
"physicalTrainUnitIdentifier": {
  "trainNumber": "4711",
  "additionalTrainNumber": "",
  "tafTapTsiTrainID": "",
  "activeOnBoardUnitId": "9920021",
}
```

6.3 Operational Plan Operational Restriction

6.3.1 Class: OperationalPlanOperationalRestriction

Description:

This class defines the implementation of one or more Operational Restrictions. It relates to one or more Operational Restriction Areas and it contains references to the Operational Restriction Events of these Operational Restriction Areas.

Attributes:

Name	Type	Multiplicity	Description	Reason
id	GenericId	1	File-wide unique, machine-interpretable identifier.	Unique identifier within the transmitted data set
uuid	GenericUuid	1	Universally unique identifier used for identification of the object across systems	Unique identifier for the communication across systems
versionId	GenericId	1	Current version of Operational Plan	Maintain the current version of Operational Plan in case of updates
mapDataVersionRef	GenericRef	1	Reference to the associated Map Data version used by the Operation Plan	Feasibility check of requested Operational Plan
operationalRestrictionAreas	OperationalRestrictionArea	1..*	List of Operational Restriction Areas to be implemented	Characteristics of Operational Restriction Areas are required for implementation of requests via SCI-CMD
operationalRestrictionEvents	OperationalRestrictionEvent	1..*	List of Operational Restriction Events to be implemented	Characteristics of Operational Restriction Events are required for implementation and triggering of requests via SCI-CMD

6.3.2 Class: OperationalRestrictionArea

Description:

This class describes the characteristics of an Operational Restriction Area.

Attributes:

Name	Type	Multiplicity	Description	Reason

id	GenericId	1	File-wide unique, machine-interpretable identifier.	Unique identifier within the transmitted data set.
uuid	GenericUuid	1	Universally unique identifier used for identification of the object across systems.	Unique identifier for the communication across systems.
operationalRestrictions	OperationalRestriction	1..*	List of Operational Restrictions to be implemented within one Operational Restriction Area	For implementation of requests via SCI-CMD
extent	TrackArea	1	Extent of the Operational Restriction Area on the railway network.	For implementation of requests via SCI-CMD

6.3.3 Class: OperationalRestriction

Description:

This class describes an Operational Restriction to be implemented for a certain Operational Restriction Area.

Attributes:

Name	Type	Multiplicity	Description	Reason
id	GenericId	1	File-wide unique, machine-interpretable identifier.	Unique identifier within the transmitted data set.
uuid	GenericUuid	1	Universally unique identifier used for identification of the object across systems.	Unique identifier for the communication across systems.
operationalRestrictionType	Enum	1	Type of the usage restriction to be implemented: Possible values are: • "MAX_SPEED_RESTRICTION": restrict the maximum allowed speed • Further restrictions are to be defined	For implementation of requests via SCI-CMD.
maximumSpeedRestriction	Integer	0..1	Restricts the maximum speed allowed (unit: km/h).	For implementation of requests via SCI-CMD.
anyClass	AnyClass	0..1	Defines an extension point for a non-SCI-OP class	Extension point for e.g. for further used restrictions.

6.3.4 Class: OperationalRestrictionEvent

Description:

This class describes the characteristics of an Operational Restriction Event.

Attributes:

Name	Type	Multiplicity	Description	Reason

id	GenericId	1	File-wide unique, machine-interpretable identifier.	Unique identifier within the transmitted data set.
uuid	GenericUuid	1	Universally unique identifier used for identification of the object across systems.	Unique identifier for the communication across systems.
OperationalRestrictionAreaRef	GenericRef	1	Reference to the Operational Restriction Area.	For implementation of requests via SCI-CMD
OperationalRestriction-EventType	Enum	1	Type of an Operational Restriction Event. Possible values are: • "START_ORA_EVENT": Operational Restriction Area with defined Operational Restrictions shall be created • "END_ORA_EVENT": Existing Operational Restriction Area shall be removed	Defines whether an Operational Restriction Event determines the temporal start or the temporal end of an ORA.
operationalEventTimes	OperationalEventTime	2	Describes the planned creation or removal times and day with their scope.	For implementation and triggering of requests via SCI-CMD.
operationalEventLinks	OperationalEventLink	0..*	List of dependencies between this Operational Event and any other Operational Event of the same or another Operational Plan.	For implementation and triggering of requests via SCI-CMD.

6.4 Operational Plan Warning Measure

6.4.1 Class: OperationalPlanOperationalWarningMeasure

Description:

This class defines the implementation of one or more Operational Warning Measures. It relates to one or more Operational Warning Areas and it contains references to the Operational Warning Measure Events of these Operational Warning Areas.

Attributes:

Name	Type	Multiplicity	Description	Reason
id	GenericId	1	File-wide unique, machine-interpretable identifier.	Unique identifier within the transmitted data set
uuid	GenericUuid	1	Universally unique identifier used for identification of the object across systems	Unique identifier for the communication across systems

versionId	GenericId	1	Current version of Operational Plan	Maintain the current version of Operational Plan in case of updates
mapDataVersionRef	GenericRef	1	Reference to the associated Map Data version used by the Operation Plan	Feasibility check of requested Operational Plan
operationalWarningAreas	Operational-WarningArea	1..*	List of Operational Warning Areas to be implemented	Characteristics of Operational Warning Areas are required for implementation of requests via SCI-CMD
operationalWarning-MeasureEvents	Operational-Warning-MeasureEvent	1..*	List of Operational Warning Measure Events to be implemented	Characteristics of Operational Warning Measure Events are required for implementation and triggering of requests via SCI-CMD

6.4.2 Class: OperationalWarningArea

Description:

This class describes the characteristics of an Operational Warning Area.

Attributes:

Name	Type	Multiplicity	Description	Reason
id	GenericId	1	File-wide unique, machine-interpretable identifier.	Unique identifier within the transmitted data set.
uuid	GenericUuid	1	Universally unique identifier used for identification of the object across systems.	Unique identifier for the communication across systems.
operationalWarningMeasures	Operational-Warning-Measures	1..*	List of Operational Warning Measures to be implemented within one Operational Restriction Area	For implementation of requests via SCI-CMD
extent	TrackArea	1	Extent of the Operational Warning Area on the railway network.	For implementation of requests via SCI-CMD
operationalWarningAreaActivationState	Operational-WarningArea-aActivation-State	1	State of Operational Warning Area	For implementation of requests via SCI-CMD

6.4.3 Class: OperationalWarningAreaActivationState

Description:

This class describes the state of an Operational Warning Area.

Attributes:

Name	Type	Multiplicity	Description	Reason
activationState	Enum	1	Activation State of the Operational Warning Area. Possible values are: • "INACTIVE": Not all listed warning devices assigned to the Operational Warning Area are "ready to warn". • "READY_TO_WARN": All listed warning devices assigned to the Operational Warning Area are actively connected and "ready to warn". They will do so with the change of the *Warning State* which is not part of this attribute.	For implementation of requests via SCI-CMD
plannedWarningDevices	Operational-WarningDevice	1..*	List of planned warning devices assigned to the Operational Warning Area that shall be "READY_TO_WARN_IN_OWA" for an Operational Movement to enter the Operational Warning Area. *This attribute is subject to change, depending on the final decision whether Warning Devices shall be integrated in RCA.*	For implementation of requests via SCI-CMD

6.4.4 Class: OperationalWarningMeasure

Description:

This class describes an Operational Warning Measure to be implemented for a certain Operational Warning Area.

Attributes:

Name	Type	Multiplicity	Description	Reason
id	GenericId	1	File-wide unique, machine-interpretable identifier.	Unique identifier within the transmitted data set.
uuid	GenericUuid	1	Universally unique identifier used for identification of the object across systems.	Unique identifier for the communication across systems.

plannedWarningDevices	Operational-WarningDe-vice	1..*	List of planned warning devices assigned to the Operational Warning Area that shall be “ready to warn” for an Operational Movement to enter the Opera-tional Warning Area.	For implementation of re-quests via SCI-CMD.
anyClass	AnyClass	0..1	Defines an extension point for a non-SCI-OP class.	Extension point for e.g. for further used warning measures.

6.4.5 Class: OperationalWarningMeasureEvent

Description:

This class describes the characteristics of an Operational Warning Measure Event.

Attributes:

Name	Type	Multiplicity	Description	Reason
id	GenericId	1	File-wide unique, machine-inter-pretable identifier.	Unique identifier within the transmitted data set.
uuid	GenericUuid	1	Universally unique identifier used for identification of the object across systems.	Unique identifier for the com-munication across systems.
OperationalWarningAreaRef	GenericRef	1	Reference to the Operational Warning Area.	For implementation of re-quests via SCI-CMD.
OperationalWarning-MeasureEventType	Enum	1	Type of an Operational Warning Measure Event. Possible values are: • “START_OWA_EVENT”: Operational Warning Area with defined Operational Restrictions shall be cre-ated • “END_OWA_EVENT”: Ex-isting Operational Warning Area shall be removed	Defines whether an Opera-tional Warning Measure Event determines the tem-poral start or the temporal end of an OWA.
operationalEventTimes	Opera-tionalEv-entTime	2	Describes the planned creation or removal times and day with their scope.	For implementation and trig-gering of requests via SCI-CMD.
operationalEventLinks	Opera-tionalEvent-Link	0..*	List of dependencies between this Operational Event and any other Operational Event of the same or another Operational Plan.	For implementation and trig-gering of requests via SCI-CMD.

6.4.6 Class: OperationalWarningDevice

Description:

This class describes the characteristics of an Operational Warning Device as an abstract description of a Physical Warning Device.

Attributes:

Name	Type	Multiplicity	Description
warningDeviceRef	GenericRef	1	Reference to physical warning device.
warningDeviceType	Enum	0..1	Type of physical warning device. Possible values are: • "WARNING_DEVICE_ACOUSTICAL_INDICATION" • "WARNING_DEVICE_OPTICAL_INDICATION" • "WARNING_DEVICE_HAPTIC_INDICATION"
warningDeviceState	Enum	0..1	• "NOT_LOGGED_IN": physical warning device not logged in and therefore not known to RCA • "LOGGED_IN": physical warning device is logged in and therefore known and identified by RCA • "ASSIGNED_TO_OWA": physical warning device is assigned to Operational Warning Area by responsible safety actor (precondition: LOGGED_IN) • "READY_TO_WARN_IN_OWA": physical warning device is ready to warn for assigned Operational Warning Area (precondition: ASSIGNED_TO_OWA). This means that the physical warning device is technically ready and all safety-related parameters of the warning (e.g. timing, volume, geographical position of the physical warning device, etc.) are granted by the responsible safety actor who shall be on site.

6.5 Operational Plan Execution Report

6.5.1 Class: ExecutionStatePE

Description:

The class describes the Execution State of a referenced Operational Event from PE's perspective.

Attributes:

Name	Type	Multiplicity	Description
executionStatePE	Enum	1	Execution State of Operational Event. Possible value are: • "COMPLETED": The Operational Event has been successfully completed. ○ An Operational Movement Event is completed for SubSys PE, if a Movement Permission was granted by SubSys SL which allocates the position of the Operational Movement Event, allowing the departure, arrival or

			passage of the Train Unit from/at this Operational Movement Event, and if the Train Unit has started moving within this Movement Permission.
			• "FAILED":
			  ○ Refer to Abstract concept: ***Execution Failure***
			• "WARNING":
			  ○ Refer to Abstract concept: ***Execution Warning***
			This section will be further elaborated in future releases.

failedCode	Enum	0..1	Code indicating the reason for failing to execute associated Operational Event. Possible values are: • to be defined *This section will be further elaborated in future releases.*
warningCode	Enum	0..1	Code indicating the reason for the warning Possible values are: • to be defined This section will be further elaborated in future releases.

6.5.2 Class: ExecutionStateAE

Description:

The class describes the Execution State of a referenced Operational Event from SubSys ATO Execution's perspective.

Attributes:

Name	Type	Multiplicity	Description

executionStateAE	Enum	1	Execution State of Operational Event.
			Possible values are: (to be clarified)
			• "COMPLETED": The Operational Event has been successfully completed.
			  ○ An Operational Movement Event is completed by SubSys AE, if the Train Unit passed, stopped at, or departed from the corresponding ATO Timing Point.
			• "DELAYED": The Operational Event cannot be completed in time but will be processed further.
			• "FAILED":
			  ○ Refer to Abstract concept: ***Execution Failure***
			• "WARNING":
			  ○ Refer to Abstract concept: ***Execution Warning***

			This section will be further elaborated in future re-leases.
failedCode	Enum	0..1	Code indicating the reason for failing to execute associated Operational Event. Possible values are: • to be defined *This section will be further elaborated in future re-leases.*
warningCode	Enum	0..1	Code indicating the reason for the warning Possible values are: • to be defined This section will be further elaborated in future releases.

6.6 Operational Plan Execution Forecast

6.6.1 Class: ExecutionForecast

Description:

The class describes the Execution Forecast of an Operational Movement Event from SubSys ATO Execution's perspective.

Attributes:

Name	Type	Multiplicity	Description
operationalMovementE-ventRef	GenericRef	1	Operational Movement Event in referenced Operational Plan which execution time is forecasted
executionForecastTime	ExecutionForecastTime	1..2	By SubSys ATO Execution forecasted execution time for Op-erational Movement Events; to be used to inform about esti-mated arrival or/and departure time

6.6.2 Class: ExecutionForecastTime

Description:

This class contains the forecast timing information of an Operational Movement Event with its scope.

Attributes:

Name	Type	Multiplicity	Description	Reason
scope	Enum	1	For Operational Movement Events, the following values are possible: • "PREDICTED_ARRIVAL": DateTime predicted by SubSys ATO Execution when Physical Train Unit will arrive (provided in upstream by SubSys ATO Exe-cution)	As feedback to Planning System via SCI-OP

			• "PREDICTED_DEPARTURE": DateTime predicted by SubSys ATO Execution when Physical Train Unit will depart (provided in upstream by SubSys ATO Execution)	
Time	DateTime	1	Timestamp at which the respective Operational Movement Event is forecasted to be executed	As feedback to Planning System via SCI-OP

6.7 Train Unit Report

6.7.1 Class: TrainUnitDataCharacteristics

Description:

This class describes the state dataset of a Physical Train Unit provided in a Train Unit Report via SCI-OP.

Attributes:

Name	Type	Multiplicity	Description
maximumSpeed	Integer	0..1	Maximum speed of Physical Train Unit
etcsOperationalTrain-Category	Enum	0..1	Provides the train type (freight, passenger), brake position, tilting capabilities and cant deficiency to which the Physical Train Unit belongs. Possible values are: • "PASS1", "PASS2", "PASS3" • "TILT1", "TILT2", "TILT3", "TILT4", "TILT5" , "TILT6" , "TILT7" • "FP1", "FP2", "FP3" , "FG4" • "FG1", "FG2", "FG3", "FG4"
axleLoadCategory	Enum	0..1	Axle Load category to which the train belongs. Possible values are: • "A" • "HS17" • "B1", "B2" • "C2", "C3", "C4" • "D2", "D3", "D4", "D4XL" • "E4", "E5"
operationalState	Enum	0..1	Describes the operational state of a Physical Train Unit. Possible values are: • "ACTIVE": Identified and localised Physical Train Unit which active OBU has established a session with subsystem of RCA • "READY_FOR_MP": Identified and localised Physical Train Unit which active OBU is ready to receive a Movement Authority

			• "INACTIVE": Identified and localised Physical Train Unit, whose OBU is inactive
unsafeTrainLength	Integer	0..1	Unsafe length of Physical Train Unit manually entered by driver (unit: meter).
safeTrainLength	Integer	0..1	Save length of Physical Train Unit provided by a system/process which guarantees safe length and train integrity (unit: meter)
movementMode	Enum	0..1	Describes the current mode of movement. Possible values are: • "STANDSTILL": Currently not moving • "ON_SIGHT": Driving by driver's safety responsibility • "FULL_SUPERVISION": Driving full supervised by the vehicle supervisor • "REVERSING": Driving opposite direction to the cab direction

6.7.2 Class: TrainUnitMovementData

Description:

This message reports updates to the position, extent and velocity of a logged-on Physical Train Unit. For each Train_unit_position_report sent, the SubSys Plan Execution provides the reference to the corresponding Operational Plan, if available. This functionality does not provide high-precision positioning to the Planning System, since APS, as well as SubSys Plan Execution and the Planning System, shall consider the inaccuracy of the localisation technology used to safely represent the Physical Train Unit.

Attributes:

Name	Type	Multiplicity	Description
extentTrainUnit	DirectedTrackPath	0..1	Describes the safe extent of the Physical Train Unit on the railway network
velocityTrainUnit	MeasuredVelocity	0..1	Measured current velocity of Physical Train Unit
verifiedPosition	Boolean	1	• TRUE: position of the Physical Train Unit is verified by the OBU • FALSE: position of the Physical Train Unit was assessed as invalid/unknown (e.g. cold movement detection) or was set manually

6.7.3 Class: MeasuredVelocity

Description:

This class describes the measured velocity of a Physical Train Unit.

Attributes:

Name	Type	Multiplicity	Description
velocity	Integer	1	Current measured velocity.

velocityConfidenceInter-valMax	Integer	1	Higher boarder of confidence interval.
velocityConfidenceInter-valMin	Integer	1	Lower boarder of confidence interval.

6.8 Track Allocation

6.8.1 Class: TrackAllocation

Description:

This class describes a track area on the railway which is allocated.

Attributes:

Name	Type	Multiplicity	Description
uuid	GenericUuid	1	Unique identifier of a Track Allocation for the communication across systems.
type	Enum	1	Type of Track Allocation. Possible values are: • “MOVEMENT”: The track is allocated for a movement • “OCCUPANCY”: The track is allocated to an occupancy
extentTrackAllocation	ContiguousTrackArea	1	Describes the extent of the Track Allocation on the railway network.

6.9 Field Element State

6.9.1 Class: FieldElementIdentifier

Description:

This class describes the identification parameters of a Field Element.

Attributes:

Name	Type	Multiplicity	Description
mapDataVersionRef	GenericRef	1	Reference to the associated Map Data version
fieldElementRef	GenericRef	1	Reference to associated Field Element (corresponds to DPSGroup in Map Data).

6.9.2 Class: FieldElementState

Description:

This class describes the type and states of a Field Element.

Attributes:

Name	Type	Multiplicity	Description

fieldElementType	Enum	1	Type of Field Element. Possible values are: (list is not exhaustive) • LEVEL_CROSSING • POINTS_WITH_TRAILING_SENSOR • POINTS_WITHOUT_TRAILING_SENSOR • POINTS_ WITH _MOVABLE_FROG • POINTS_ WITHOUT_MOVABLE_FROG • ONE_WAY_SPRING_POINT • DERAILER_DIRECTIONAL • DERAILER_ NON_DIRECTIONAL • DERAILMENT_POINT • CROSSING_WITH_MOVABLE_FROG • MOVABLE_BRIDGE • MOVABLE_BUFFER_STOP • GATE_INFRINGING_THE_STRUCTURE_GAUGE • TRAVERSER • TURNTABLE • MOVABLE_RAMPS_WITH_MONITORING_SEN-SORS • WEIR_CHAMBER • LANDSLIDE_AND_AVALANCHE_DETECTION • POSITION_DETECTION_OF_CONSTRUCTIONS • KEY_LOCKS_WITH_SENSORS • CATENARY_SECTION_WHERE_TRAC-TION_IS_CHANGED *The non-exhaustive list of Field Element Types is prone to changes and shall be re-checked for modifications made by the RCA APS Cluster. Source.*
fieldElementState	Enum	1	State of the Field Element (corresponds to the DPSGroupState of DPSGroup on SCI-CMD). Possible values are: • "PROCESSING": Field Element is not in end position (e.g. point is turning, level crossing is opening/closing) • "READY": Field Element is in end position • "FAILURE": Field Element has a failure
fieldElementDriveabilityState	Enum	1	Driveability State of the Field Element (corresponds to the DPSDriveabilityState of the DPSGroup on SCI-CMD). Possible values are: • to be defined
fieldElementFlankProtection-State	Enum	0..1	Flank protection state of the Field Element (corresponds to the DPSFlankprotectionState of the DPSGroup on SCI-CMD) Possible values are: • to be defined
failureCode	Enum	0..1	Failure code of Field Element which was reported on SCI-CMD Possible values are:

			• to be defined

6.10 Common

6.10.1 Class: TrackEdgePosition

Description:

This class describes a position on a Track Edge within the Map Data.

Attributes:

Name	Type	Multiplicity	Description	Reason
trackEdgeRef	GenericRef	1	Reference to the associated Track Edge provided by the version of Map Data currently in use.	-
offset	Integer	1	Describes the offset of the event position on the referenced Track Edge (unit: meters). The offset is always defined in relation to the start node of the referenced Track Edge.	-

6.10.2 Class: TrackEdgeSection

Description:

This class describes a section on a Track Edge within the Map Data.

Attributes:

Name	Type	Multiplicity	Description	Reason
trackEdgeRef	GenericUuid	1	Reference to the associated Track Edge within the version of Map Data currently in use.	-
offsetA	Integer	0..1	Describes the offset on the referenced Track Edge (unit: meters). The offset is always defined in relation to the start node of the referenced Track Edge. Offset A shall be numerically smaller than Offset B. If offsetA is not given, this means that Track Section begins at start node of the Track Edge (offsetA = 0)	Something on the track does not cover/allocate a whole trackEdge, but only a section of this TrackEdge. To define the extent of the section, an offset on both ends of the TrackEdge is needed.
offsetB	Integer	0..1	Describes the offset on the referenced Track Edge (unit: me-	-

			ters). The offset is always defined in relation to the start node of the referenced Track Edge. Offset B shall be numerically larger than Offset A. If offsetB is not given, this means that Track Section ends at end node of the Track Edge (offsetB = trackEdge.length)	
direction	Enum	1	Direction relative to the given direction of the Track Edge (start node, end node). Possible values are: • "UP": This denotes the direction from the start node of the referenced Track Edge to the end node of the referenced Track Edge • "DOWN": This goes opposite to "up" • "NONE": no defined direction • "BOTH": both directions are given	-

6.10.3 Class: TrackArea

Description:

This class describes the extent of a Track Area on the railway network.

Attributes:

Name	Type	Multiplicity	Description
trackEdgeSections	TrackEdgeSection	1..*	List of Track Edge Sections which define the extent of the Track Area

6.10.4 Class: ContiguousTrackArea

Description:

This class describes the extent of a Contiguous Track Area on the railway network.

Attributes:

Name	Type	Multiplicity	Description
trackEdgeSections	TrackEdgeSection	1..*	List of contiguous (gap-free and overlap-free) Track Edge Sections which define the extent of the Contiguous Track Area

6.10.5 Class: DirectedTrackPath

Description:

This class describes the extent of a navigable, gap-free, and overlap-free path on the railway network in a defined direction. In practice, navigability means that railway network offers the possibility for a Physical Train Unit to run over the Track Path without direction change.

Attributes:

Name	Type	Multiplicity	Description
rearPosition	TrackEdgePosition	1	Rear position of Track Path
rearPositionMovementDirection	Enum	1	Movement direction along the Track Edge. Possible values are: • "UP": This denotes the direction from the start node to the end node (increasing relative position values) • "DOWN": This goes opposite to up (decreasing relative position values)
viaTracks	TrackEdgeSection	0..*	Sorted list of Track Edge Sections lying between the Track Edge of the rear position and the Track Edge of the front position along the Track Path. The Track Edges referenced by the rear- and front position are excluded from the list.
frontPosition	TrackEdgePosition	1	Front position of Track Path

7 Cross-cutting issues and NFRs

This chapter describes aspects of SCI-OP that address both cross-cutting issues as well as non-functional requirements.

7.1 Versioning of Messages

As the traffic situation may change during execution because of conditions in the physical world (e.g. train delays, infrastructure element malfunctions), the Operational Plans need to be adapted regularly. By receiving feedback from the SubSys's via the upstream, the Planning System knows about the actual operational situation and can determine the deviation from the Operational Plan. Based on that, the Planning System (or responsible staff) might decide to re-optimise the schedule (e.g. change the order of trains) and to elaborate a new version of the Operational Plan.

7.1.1 Versioning of Downstream Messages

Version management is to be carried out by the Planning System. Therefore, a new version of an Operational Plan will be generated by the Planning System and sent to SubSys Plan Execution and SubSys ATO Execution via SCI-OP.

All details about versioning, update etc. of an Operational Plan is provided in the Architectural Design for Plan Execution (RCA.Doc. 49), section 4.1.7. On the interface SCI-OP it shall be assured that all downstream messages coming from the Planning System shall be equipped with a unique ID of the Operational Plan as well as a version number of that Operational Plan. The relevant data objects (uuid, versionId) are already considered in the data model (chapter 0):

7.1.2 Versioning of Upstream Messages

All upstream messages provide a reference to the corresponding Operational Plan, if applicable and available. If this reference can be stated by the sending SubSys, it is required to enrich this message with the corresponding version number. Therefore, the data object versionID is to be provided in upstream messages.

7.2 Message Quantity Structure

Assessment on number of messages transferred over SCI-OP within a time range

This section will be further elaborated in future releases.

7.3 Integration of Map Data in Messages on SCI-OP

- Analyse and integrate requirement for SCI-OP regarding Map Data activation

This section will be further elaborated in future releases.

7.4 Timing of Messages

- To determine until when it makes sense to send upstream messages before it is too late for the Planning System to react on them

This section will be further elaborated in future releases.

8 Open Points

This chapter describes the open points which will be addressed in future releases of this concept.

8.1 Prio 1

- Finalise consolidation of URA types with APS cluster
    - Update table in 4.1.3.2
    - Harmonise with 6.3.3
    - Modify data object structure: Presumably, each operationalRestrictionType requires his specific attributes
- Analyse and integrate requirements for SCI-OP regarding SubSys ATO Execution
- Develop Return States for Class: ExecutionStateAE
    - Develop Return States for Class: ExecutionStatePE, e.g. completed or delayed
- Add Operational Plan Event Reference to Operational Plan Movement Events based on Abstract Concept in Architectural Design
- Forecast of Operational Events: Define new message type or new state for Operational Event

8.2 Prio 2

- Full Operating State for initialisation of the Planning System:
    - Do we need messages for an initial operating state? For instance, to report the actual state during initialisation of the Planning System. Do we might need commands in the downstream to request e.g. the actual track allocations?
    - For the Field Element State, too.
- Comments by Jens in model of Operational Plan Operational Warning Measure (Figure):
    - Do we need to show Start and End events as types of the Operational Event?
    - I assume that it is up to the planning system to “group” areas in one Operational Plan Operational Warning Measure.
- Where shall WA be planned?
    - Should WA be planned in the Planning System or somewhere else?
    - Where would that be?
    - What are the impacts on SCI-OP?
- Challenge return codes of feasibility checks of Operational Plan Execution Response
- Peter M: There is an inconsistency between the Architecture Poster (SCI-OP/SCI-PS/SCI-OS) and this document (SCI-OP). We will need to align somehow on these interfaces.
- Check the necessity of uuid's for all SCI-OP data objects

8.3 Prio 3

- Incorporate versioning of MAP data in all messages
    - among others: Map data version in Operational Events
- Challenge whether PE Alarms must be considered in Upstream. Open: Will we get all necessary data from APS?
- Update Class: StopTimes
- Update Class: StopActivity
- Integrate message descriptions and attributes, as well as all classes of data model to RCA Capella-model as “candidates”
- DSO Trasse: Add Extension point for solving differences between OP and manual settings performed on the Workbench
- Introduction of underlying design pattern “Published Language (DDD pattern)” of the Operational Plan
- Address further open points in the chapters 3 and 4 (marked in red italic text)

9 Appendix

This chapter holds an example of an Operational Plan Operational Movement in JSON-Format.

9.1 Example of an Operational Plan Operational Movement as JSON


```json
{
  "operationalPlanOperationalMovement": [
    {
      "id": "ID-OPOM-1",
      "uuid": "UUID-OPOM-1",
      "versionId": "VID-OPOM-1",
      "mapDataVersionRef": "ID-INST-1",
      "operationalSegments": [
        {
          "id": "ID-OS-1",
          "uuid": "UUID-OS-1",
          "sequence": 1,
          "operationalTrainUnitRef": "ID-OTU-1"
        },
        {
          "id": "ID-OS-2",
          "uuid": "UUID-OS-2",
          "sequence": 2,
          "operationalTrainUnitRef": "ID-OTU-2"
        },
        {
          "id": "ID-OS-3",
          "uuid": "UUID-OS-3",
          "sequence": 3,
          "operationalTrainUnitRef": "ID-OTU-3"
        }
      ],
      "operationalTrainUnits": [
        {
          "id": "ID-OTU-1",
          "uuid": "UUID-OTU-1",
          "physicalTrainUnitIdentifier": {
            "trainNumber": "4711",
            "additionalTrainNumber": "",
            "tafTapTsiTrainID": "",
            "activeOnBoardUnitId": "9920021"
          },
          "operationalTrainCategory": {
            "id": "ID-CAT-1",
            "uuid": "UUID-CAT-1",
            "name": "RegionalTrain",
            "description": "",
            "trainUsage": "COMMERCIAL_PASSENGER_TRAIN"
          },
          "operationalConsistRefs": [
            {
              "operationalConsistRef": "ID-OC-11",
              "position": 1
            },
            {
              "operationalConsistRef": "ID-OC-12",
              "position": 2
            }
          ],
          "brakeWeightPercentage": {
            "regularBrakeWeightPercentage": 150,
            "emergencyBrakeWeightPercentage": 200
          }
        },
        {
          "id": "ID-OTU-2",
          "uuid": "UUID-OTU-2",
          "physicalTrainUnitIdentifier": {
```

```
          "trainNumber": "4711",
          "additionalTrainNumber": "",
          "tafTapTsiTrainID": "",
          "activeOnBoardUnitId": "9920021"
        },
        "operationalTrainCategory": {
          "id": "ID-CAT-1",
          "uuid": "UUID-CAT-1",
          "name": "RegionalTrain",
          "description": "",
          "trainUsage": "COMMERCIAL_PASSENGER_TRAIN"
        },
        "operationalConsistRefs": [
          {
            "operationalConsistRef": "ID-OC-11",
            "position": 1
          }
        ],
        "brakeWeightPercentage": {
          "regularBrakeWeightPercentage": 150,
          "emergencyBrakeWeightPercentage": 200
        }
      },
      {
        "id": "ID-OTU-3",
        "uuid": "UUID-OTU-3",
        "physicalTrainUnitIdentifier": {
          "trainNumber": "4711",
          "additionalTrainNumber": "",
          "tafTapTsiTrainID": "",
          "activeOnBoardUnitId": "9920021"
        },
        "operationalTrainCategory": {
          "id": "ID-CAT-1",
          "uuid": "UUID-CAT-1",
          "name": "RegionalTrain",
          "description": "",
          "trainUsage": "COMMERCIAL_PASSENGER_TRAIN"
        },
        "operationalConsistRefs": [
          {
            "operationalConsistRef": "ID-OC-12",
            "position": 1
          },
          {
            "operationalConsistRef": "ID-OC-11",
            "position": 2
          }
        ],
        "brakeWeightPercentage": {
          "regularBrakeWeightPercentage": 150,
          "emergencyBrakeWeightPercentage": 200
        }
      }
    ],
    "operationalConsists": [
      {
        "id": "ID-OC-11",
        "uuid": "UUID-OC-11",
        "supportedOnBoardEquipment": [
          {
            "type": "ETCS",
            "etcsSystemVersions": [
              {
                "etcsSystemVersion": "1.0"
              },
              {
                "etcsSystemVersion": "1.1"
              },
              {
```

```
              "etcsSystemVersion": "2.0"
            }
          ]
        }
      ]
    },
    {
      "id": "ID-OC-12",
      "uuid": "UUID-OC-12",
      "supportedOnBoardEquipment": [
        {
          "type": "ETCS",
          "etcsSystemVersions": [
            {
              "etcsSystemVersion": "1.0"
            },
            {
              "etcsSystemVersion": "1.1"
            },
            {
              "etcsSystemVersion": "2.0"
            }
          ]
        }
      ]
    }
  ],
  "operationalMovementEvents": [
    {
      "id": "ID-OME-11",
      "uuid": "UUID-OME-11",
      "sequence": 1,
      "toBeExecuted": true,
      "type": "STOP",
      "alignment": "HEAD",
      "operationalMovementEventPosition": {
        "type": "POSITION_ON_TRACK",
        "trackEdgePosition": {
          "trackEdgeRef": "648390da",
          "offset": 267.000
        }
      },
      "operationalEventTimes": [
        {
          "scope": "EARLIEST_DEPARTURE",
          "dateTime": "2020-03-11 03:07:00 PM"
        },
        {
          "scope": "LATEST_DEPARTURE",
          "dateTime": "2020-03-11 03:08:30 PM"
        }
      ],
      "operationalEventLinks": [],
      "trackPathToSubsequentEvent": [
        {
          "trackEdgeRef": "648390da",
          "direction": "UP"
        },
        {
          "trackEdgeRef": "6485183d",
          "direction": "DOWN"
        }
      ],
      "stopDescription": {}
    },
    {
      "id": "ID-OME-12",
      "uuid": "UUID-OME-12",
      "sequence": 2,
      "toBeExecuted": true,
```

```
          "type": "PASSAGE",
          "alignment": "HEAD",
          "operationalMovementEventPosition": {
            "type": "POSITION_ON_TRACK",
            "trackEdgePosition": {
              "trackRef": "6485183d",
              "offset": 127.000
            }
          },
          "operationalEventTimes": [
            {
              "scope": "EARLIEST_DEPARTURE",
              "dateTime": "2020-03-11 03:12:30 PM"
            },
            {
              "scope": "LATEST_DEPARTURE",
              "dateTime": "2020-03-11 03:14:00 PM"
            }
          ],
          "operationalEventLinks": [],
          "trackPathToSubsequentEvent": [
            {
              "trackEdgeRef": "6485183d",
              "direction": "DOWN"
            },
            {
              "trackEdgeRef": "648abd2e",
              "direction": "UP"
            },
            {
              "trackEdgeRef": "64873c70",
              "direction": "UP"
            }
          ],
          "stopDescription": {}
        },
        {
          "id": "ID-OME-13",
          "uuid": "UUID-OME-13",
          "sequence": 3,
          "toBeExecuted": true,
          "type": "STOP",
          "alignment": "HEAD",
          "operationalMovementEventPosition": {
            "type": "POSITION_ON_TRACK",
            "trackEdgePosition": {
              "trackRef": "64873c70",
              "offset": 300000
            }
          },
          "operationalEventTimes": [
            {
              "scope": "EARLIEST_ARRIVAL",
              "dateTime": "2020-03-11 03:19:00 PM"
            },
            {
              "scope": "LATEST_ARRIVAL",
              "dateTime": "2020-03-11 03:20:00 PM"
            },
            {
              "scope": "EARLIEST_DEPARTURE",
              "dateTime": "2020-03-11 03:21:00 PM"
            },
            {
              "scope": "ATEST_DEPARTURE",
              "dateTime": "2020-03-11 03:22:00 PM"
            }
          ],
          "operationalEventLinks": [],
          "trackPathToSubsequentEvent": [
```



```
          {
            "trackEdgeRef": "64873c70",
            "direction": "UP"
          },
          {
            "trackEdgeRef": "648676ee",
            "direction": "UP"
          }
        ],
        "stopDescription": {
          "commercial": true,
          "stopOnRequest": true,
          "onOff": "BOTH",
          "stopTimes": {
            "minimalTime": "00:01:00",
            "operationalReserve:": "00:00:30",
            "additionalReserve": "00:00:30"
          },
          "stopActivities": [
            {
              "type": "SPLIT",
              "actualOperationalSegmentRef": "ID-OS-1",
              "targetOperationalSegmentRef": "ID-OS-2"
            }
          ],
          "platformEdgeRefs": [
            {
              "platformEdgeRef": "UUID-PE-1"
            }
          ]
        }
      },
      {
        "id": "ID-OPE-14",
        "uuid": "UUID-OPE-14",
        "sequence": 4,
        "toBeExecuted": true,
        "type": "STOP",
        "trainReverse": true,
        "alignment": "REAR",
        "operationalMovementEventPosition": {
          "type": "POSITION_ON_TRACK",
          "trackEdgePosition": {
            "trackRef": "648676ee",
            "offset": 1550.000
          }
        },
        "operationalEventTimes": [
          {
            "scope": "EARLIEST_ARRIVAL",
            "dateTime": "2020-03-11 03:25:00 PM"
          },
          {
            "scope": "LATEST_ARRIVAL",
            "dateTime": "2020-03-11 03:26:00 PM"
          },
          {
            "scope": "EARLIEST_DEPARTURE",
            "dateTime": "2020-03-11 03:30:00 PM"
          },
          {
            "scope": "LATEST_DEPARTURE",
            "dateTime": "2020-03-11 03:31:00 PM"
          }
        ],
        "operationalEventLinks": [],
        "trackPathToSubsequentEvent": [
          {
            "trackEdgeRef": "648676ee",
            "direction": "DOWN"
```



```
          },
          {
            "trackEdgeRef": "64873c70",
            "direction": "DOWN"
          }
        ],
        "stopDescription": {
          "commercial": true,
          "stopOnRequest": true,
          "onOff": "BOTH",
          "stopTimes": {
            "minimalTime": "00:03:00",
            "operationalReserve:": "00:01:30",
            "additionalReserve": "00:01:30"
          },
          "stopActivities": [
            {
              "type": "REVERSE"
            }
          ],
          "platformEdgeRefs": [
            {
              "platformEdgeRef": "UUID-PE-2"
            }
          ]
        }
      },
      {
        "id": "ID-OPE-15",
        "uuid": "UUID-OPE-15",
        "sequence": 5,
        "toBeExecuted": true,
        "type": "STOP",
        "alignment": "HEAD",
        "operationalMovementEventPosition": {
          "type": "POSITION_ON_TRACK",
          "trackEdgePosition": {
            "trackRef": "64873c70",
            "offset": 287.500
          }
        },
        "operationalEventTimes": [
          {
            "scope": "EARLIEST_ARRIVAL",
            "dateTime": "2020-03-11 03:34:00 PM"
          },
          {
            "scope": "LATEST_ARRIVAL",
            "dateTime": "2020-03-11 03:25:00 PM"
          }
        ],
        "operationalEventLinks": [],
        "trackPathToSubsequentEvent": {},
        "stopDescription": {
          "commercial": true,
          "stopOnRequest": true,
          "onOff": "BOTH",
          "stopTimes": {
            "minimalTime": "00:03:00",
            "operationalReserve:": "00:01:30",
            "additionalReserve": "00:01:30"
          },
          "stopActivities": [
            {
              "type": "JOIN",
              "actualOperationalSegmentRef": "ID-OS-2",
              "targetOperationalSegmentRef": "ID-OS-3"
            }
          ],
          "platformEdgeRefs": [
```



```
                {
                  "platformEdgeRef": "UUID-PE-3"
                }
              ]
            }
          }
        ]
      }
    ]
}
```