OETCS/WP3/D3.5.3

openETCS

ITEA2 Project
Call 6 11025
2012 – 2015

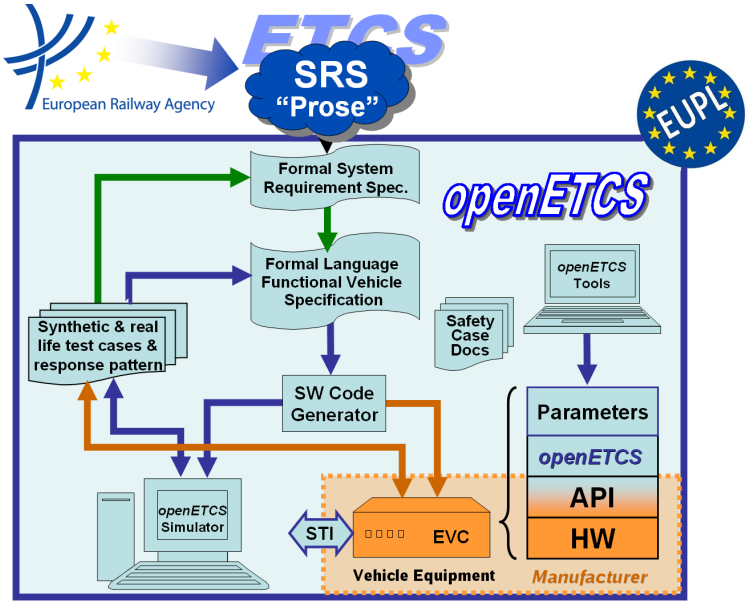
Work Package 3: “Modeling”

openETCS Architecture and Design Specification

Software Component Design and Internal Interface Specification

Peter Mahlmann, Bernd Hekele, Baseliyos Jacob, Peyman Farhangi, Stefan Karg, Valerio D´Angelo, Uwe Steinke, Christian Stahl, Jakob Gärtner, Mairamou Haman Adji, Jos Holtzer, Jan Welvaarts, Vincent Nuhaan, Thorsten Schulz, Benjamin Beichler, Marielle Petit-Doche, Matthias Güdemann, Veronique Gontier, Christian Giraud, Fausto Cochetti, Alexander Stante and David Mentre

August 2015



Funded by:

Federal Ministry of Education and Research
Région de Bruxelles-Capitale
GOBIERNO DE ESPAÑA
MINISTERIO DE INDUSTRIA, ENERGÍA Y TURISMO

This page is intentionally left blank

Work Package 3: “Modeling”

OETCS/WP3/D3.5.3
August 2015

openETCS Architecture and Design Specification

Software Component Design and Internal Interface Specification

Document approbation

Lead author:	Technical assessor:	Quality assessor:	Project lead:
location / date	location / date	location / date	location / date
signature Peter Mahlmann (DB Netz AG)	signature Jan Welte (Technische Universität Braunschweig)	signature Izaskun de la Torre (SQS)	signature Klaus-Rüdiger Hase (DB Netz)

Peter Mahlmann, Bernd Hekele, Baseliyos Jacob, Peyman Farhangi, Stefan Karg, Valerio D´Angelo

DB Netz AG

Uwe Steinke

Siemens AG

Christian Stahl

TWT GmbH

Jakob Gärtner, Mairamou Haman Adji

LEA Railergy

Jos Holtzer, Jan Welvaarts, Vincent Nuhaan

Nederlandss Spoorwegen

Thorsten Schulz, Benjamin Beichler

University of Rostock

Marielle Petit-Doche, Matthias Güdemann

Systerel

Veronique Gontier

All4Tec

Christian Giraud, Fausto Cochetti

Alstom

Alexander Stante

Fraunhofer ESK

David Mentre

MERCE

Architecture and Design Specification

Prepared for openETCS@ITEA2 Project

Abstract: This document describes the architecture and design specification of the openETCS onboard unit (OBU) model. The functional scope of the openETCS OBU model is to cover the functionality required for running on the ETCS level 2 Utrecht Amsterdam track. The OBU model is developed iteratively and the system model is documented in D3.5.x and the functional model is documented in D3.5.x, where x denotes the iteration.

Modification History

Version	Sections	Modification / Description	Author	Date
0.1	all	Initial document providing structure	Peter Mahlmann	27.05.2015
0.2	2	New template for design descriptions	Peter Mahlmann	10.06.2015
0.3	all	Transferred existing documentation to new template	Peter Mahlmann	22.06.2015

Table of Contents

Modification History iv

Figures and Tables viii

1 Introduction 1

1.1 Input Documents 1
1.2 Software and Tools used for Development 1
1.3 General Remarks on the openETCS OBU Model 2

I System Architecture and Functional Breakdown 3

2 System Architecture 4

2.1 Top Level Architecture and External Interfaces 4
2.2 Functional breakdown of the ETCS OBU 4

II Design Description 8

3 F1: Receive Information from Trackside 9

3.1 ETCS Messaging: TrackMessages 9
3.1.1 Component Requirements 9
3.1.2 Interface 10
3.1.2.1 Inputs 10
3.1.2.2 Outputs 11
3.1.3 Sub Components 13
3.1.3.1 Read_Packets 13
3.1.3.2 Component Requirements 13
3.1.3.3 Extract Packet 5 14
3.1.3.4 Component Requirements 14

4 F2: ETCS Kernel 16

4.1 Manage_TrackSideInformation_Integration 16
4.1.1 Component Requirements 16
4.1.2 Interface 16
4.1.2.1 Inputs 17
4.1.2.2 Outputs 22
4.1.3 Sub Components 27
4.1.3.1 Receive_TrackSide_Msg 27
4.1.3.2 CheckBGConsistency 29
4.1.3.3 CheckEuroradioMessage 30
4.1.3.4 ValidateDataDirection 30
4.1.3.5 InformationFilter 31
4.2 Speed and Distance Monitoring 32
4.2.1 Component Requirements 32
4.2.2 Interface 34
4.2.2.1 Inputs 34
4.2.2.2 Outputs 39

4.2.3 Sub Components ... 42
4.2.3.1 Receive_TrackSide_Msg ... 42
4.2.3.2 TargetManagement ... 43
4.2.3.3 CalcBrakingCurves_Integration ... 44
4.2.3.4 SDMLimitLocations ... 45
4.2.3.5 CalcSpeeds ... 46
4.2.3.6 ReleaseSpeed_Selection ... 47
4.2.3.7 SDM_Commands ... 47
4.2.3.8 SDM_OutputWrapper ... 48
4.3 Manage_ETCS_Procedures ... 49
4.3.1 Component Requirements ... 49
4.3.2 Interface ... 49
4.3.2.1 Inputs ... 49
4.3.2.2 Outputs ... 55
4.3.3 Sub Components ... 58
4.3.3.1 Awakness_of_Train ... 58
4.3.3.2 NP ... 58
4.3.3.3 SoM_L2_3_FS_SR_OS_LS_SH ... 59
4.3.3.4 SoM_NTC_SN ... 59
4.4 Manage_Track_Data ... 60
4.4.1 Component Requirements ... 60
4.4.2 Interface ... 60
4.4.2.1 Inputs ... 60
4.4.2.2 Outputs ... 61
4.4.3 Sub Components ... 62
4.4.3.1 Calculate_Train_Position ... 62
4.4.3.2 Inputs ... 63
4.4.3.3 Outputs ... 68
4.4.3.4 Provide_Position_Report ... 72
4.5 trainData ... 73
4.5.1 Component Requirements ... 73
4.5.2 Interface ... 74
4.5.2.1 Inputs ... 74
4.5.2.2 Outputs ... 75
4.5.3 Sub Components ... 76
4.5.3.1 Calculate_Train_Position ... 76
4.5.3.2 Inputs ... 77
4.5.3.3 Outputs ... 82
4.5.3.4 Provide_Position_Report ... 86
4.6 Mode_and_Level ... 87
4.6.1 Component Requirements ... 87
4.6.2 Interface ... 88
4.6.2.1 Inputs ... 88
4.6.2.2 Outputs ... 89
4.6.3 Sub Components ... 90
4.6.3.1 Level_Management ... 90
4.6.3.2 Mode_Management ... 90
4.6.3.3 Check_and_Provide_Mode_and_Level ... 91
4.7 Manage_Radio_Communication ... 92
4.7.1 Component Requirements ... 92
4.7.2 Interface ... 92
4.7.2.1 Inputs ... 92

4.7.2.2 Outputs 93
4.7.3 Sub Components 94
4.7.3.1 Management_of_Radio_Communication 94

5 F3: Measure Train Movement 97

6 F4: Manage Radio Communication 98

7 F5: Manage JRU 99

8 F6: DMI Controller 100

8.1 DMI 100
8.1.1 Component Requirements 100
8.1.2 Interface 100
8.1.2.1 Inputs 100
8.1.2.2 Outputs 105

References 109

Figures and Tables

Figures

Figure 1. Scope of openETCS OBU model system according to ERA TSI Chapter 2.5. 5
Figure 2. Top level architecture with external interfaces. 6
Figure 3. DMI Interfaces. 6
Figure 4. 2nd level system architecture view. 7
Figure 5. Component SysML diagram 10
Figure 6. Manage_TrackSideInformation_Integration component SysML diagram. 17
Figure 7. High level overview of the InformationFilter components. 33
Figure 8. Structure of component SpeedSupervision_Integration. 38
Figure 9. Manage_ETCS_Procedures component SysML diagram 49
Figure 10. Manage_Track_Data component SysML diagram 65
Figure 11. Component SysML diagram 65
Figure 12. Provide Position Report component SysML diagram 73
Figure 13. Manage_Track_Data component SysML diagram 74
Figure 14. Component SysML diagram 79
Figure 15. Provide Position Report component SysML diagram 87
Figure 16. Mode_and_Level component SysML diagram 88
Figure 17. Manage_Radio_Communication component SysML diagram 96
Figure 18. DMI component SysML diagram 105

Tables

1 Introduction

A primary goal of the openETCS ITEA2 project is to provide a formal specification and a non-vital reference implementation of an ETCS onboard unit (OBU) according to the specification described in Subset-026 [1], defined the European Railway Agency (ERA).

This deliverable, i.e. D3.5.x, describes the architecture and design specification of the openETCS onboard (OBU) model. As the development of the OBU model is done iteratively according to a SCRUM process, the last digit of the deliverable identifier, i.e. x, denotes the current iteration of the model. This document should be considered as a complement to the following project outcomes respectively deliverables:

- the corresponding SysML and SCADE models, available at https://github.com/openETCS/modeling/tree/master/model/Scade/System,
- the corresponding functional design description, i.e. D3.6.x, and
- the documentation of the generic openETCS Application Programming Interface (API), available at https://github.com/openETCS/modeling/blob/master/API/description/api-description.pdf.

1.1 Input Documents

The following documents have been the basis for the analysis, functional decomposition, and design of the openETCS OBU model:

- ERA Subset-026 [1]
- ERA TSI CCS Documents
- openETCS API documentation, available at https://github.com/openETCS/modeling/blob/master/API/description/api-description.pdf
- openETCS requirements, i.e. D2.1...9, available at https://github.com/openETCS/requirements/tree/master/Reference

list has to be completed

1.2 Software and Tools used for Development

The following software and tools have been used in the openETCS development process:

list and descriptions have to be checked for completeness

SCADE System Version 16.1b of SCADE System has been used for the the genereation of SysML models.

SCADE Suite Version 16.1b of SCADE Suite has been used for the functional modelling of the openETCS OBU components. Executable models are generated via the SCADE Suite code generator (KCG), which has been certified for CENELEC EN 50128 at SIL 3/4.

SCADE Display Version 16.1b of SCADE Display has been used for the development of the Driver Machine Interface (DMI).

GitHub The web based Git repository hosting service GitHub has been used for distributed revision control and source code respectively model management.

1.3 General Remarks on the openETCS OBU Model

The openETCS OBU model has been developed according the specification given in ERA Subset-026 [1]. The software release of the openETCS OBU documented and described in this document is publicly available at https://github.com/openETCS/modeling/tree/master/model and refers to the commit corresponding to hashtag

```
1c06cc2d4a0d8f27569e065e2a9edf924b453ff1
```

In particular, the root of the SCADE System SysML model is located at https://github.com/openETCS/modeling/tree/master/model/system and the root of the functional SCADE Suite model is located at https://github.com/openETCS/modeling/tree/master/model/Scade.

Note that all components of the openETCS OBU have been developed from scratch, no existing components have been reused.

Part I

System Architecture and Functional Breakdown

2 System Architecture

The system architecture of the openETCS OBU is adopted from the system structure defined in ERA Subset-026, Chapter 2.5 [1]. Figure 1 shows which parts of the reference architecture are in the scope of the openETCS OBU model. Note that also specific parts of the ETCS trackside (e.g. Eurobalise and RBC blocks) have been modeled to have an integrated test environment, c.f. dashed blue line in Figure 1.

2.1 Top Level Architecture and External Interfaces

Figure 2 shows the top level architecture with external interfaces E1, E2,. . ., E9. The external interfaces are used for the communication between the openETCS OBU (dashed red line) and systems out of the scope of the openETCS project and the ETCS Onboard Unit System.In the following we give brief overview of the interfaces:

E1: In- and out flow between the Interlocking and the Eurobalise. Only relevant for controlled Eurobalises.

E2: In- and out flow between the Interlocking and Radio Block Control. This interface ensures the states or logics directly to the Radio Block Control and the other way back from the train to the interlocking.

E3: Input flow from the Eurobalise to the Balise Transmission Module or Antenna Unit (BTM) into the ETCS OBU.

E4: In- and out flow between the Radio Block Control and the Euroradio modul into the ETCS OBU. This interface is not active in ETCS levels 0 and 1 since there is no ETCS radio interaction between track and train in these levels.

E5: This interface is used for the interaction between the driver and the display (Driver Machine Interface, DMI), c.f. Figure 3.

E6: This interface is a compound structure and combines the interfaces E3 and E4.

E7: Input interface to the odometry subsystem of the ETCS OBU. Used for sending information to the train if there is any movement outside the ETCS System, e.g. "cold movement".

E8: Input interface to the ETCS OBU to set configuration data such as fixed values, system values, national values and train configuration.

E9: In- and Out flow between the ETCS OBU and the train. This interface is used for the interaction between the Train and the ETCS OBU such as brake control, traction control, door control, etc.

2.2 Functional breakdown of the ETCS OBU

Figure 4 depicts the functional breakdown of the ETCS OBU block shown in Figure 2.The internal interfaces

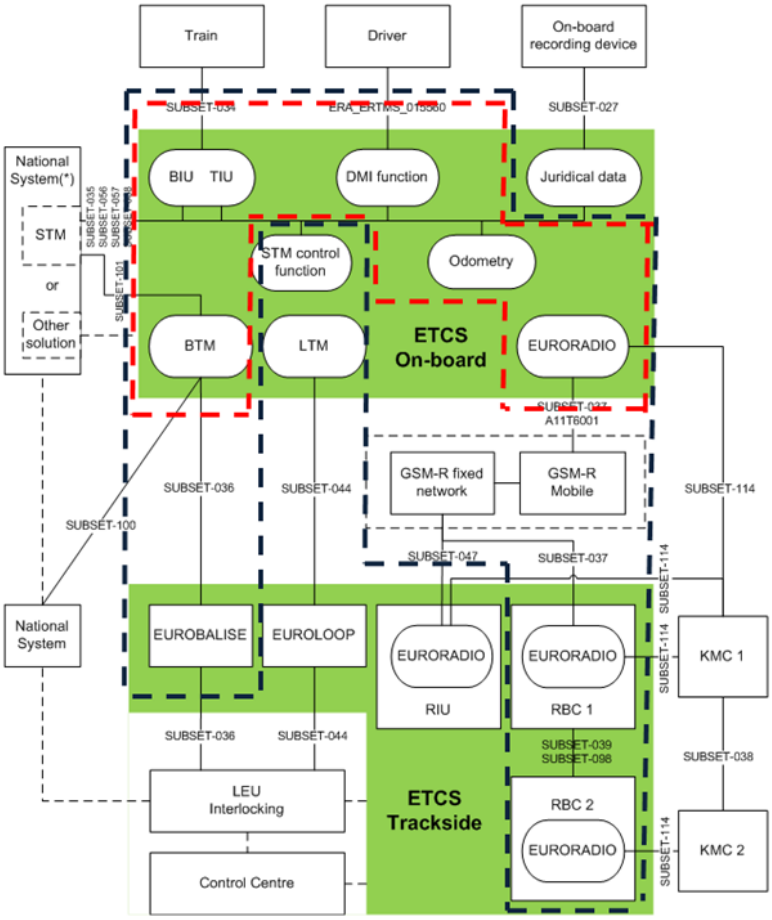


Figure 1. Scope of openETCS OBU model system according to ERA TSI Chapter 2.5. Functional blocks in the scope of openETCS have been marked by the dashed blue line. The dashed red line shows the OBU blocks in the scope of openETCS.

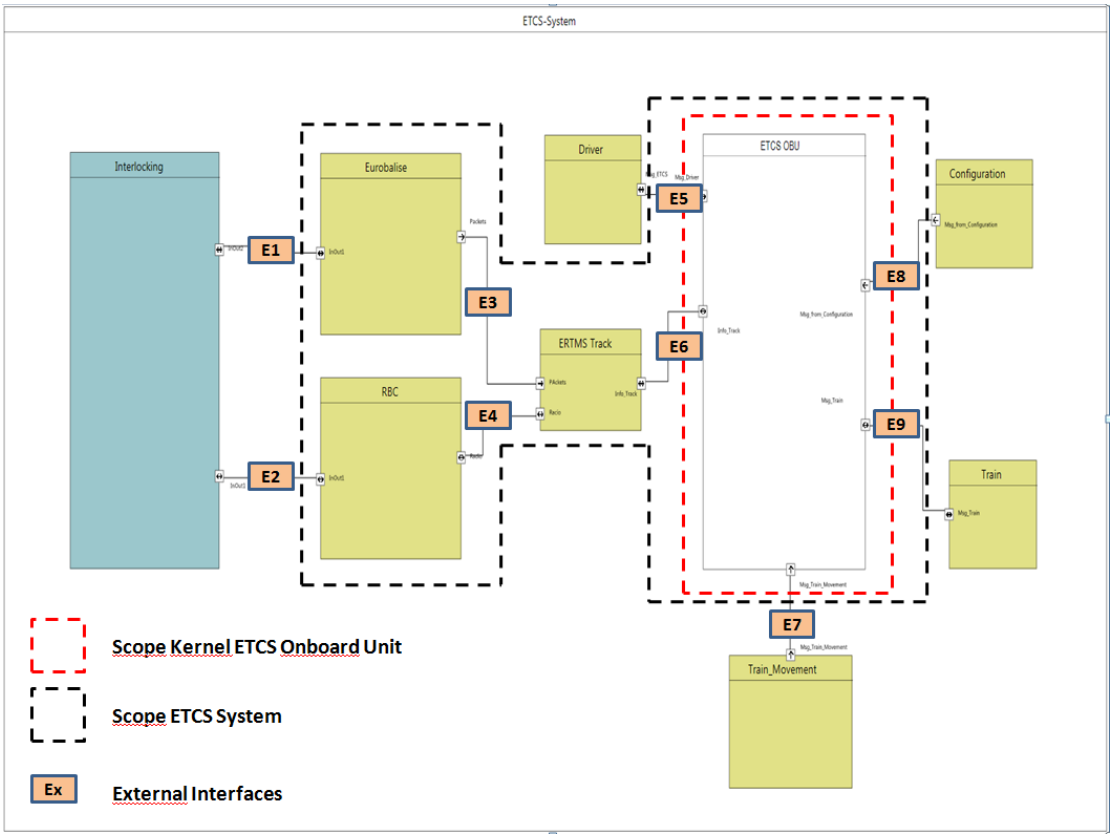


Figure 2. Top level architecture with external interfaces.

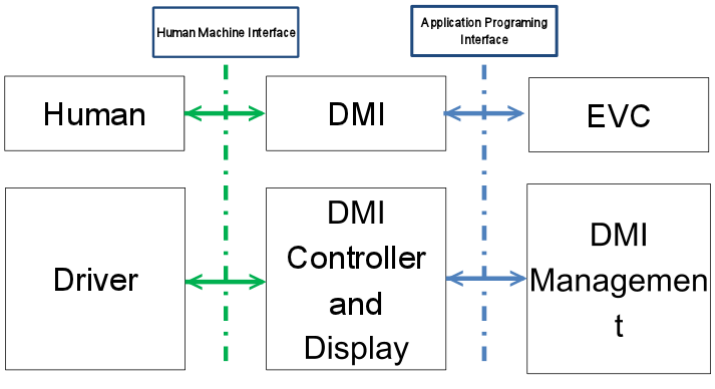


Figure 3. DMI Interfaces.

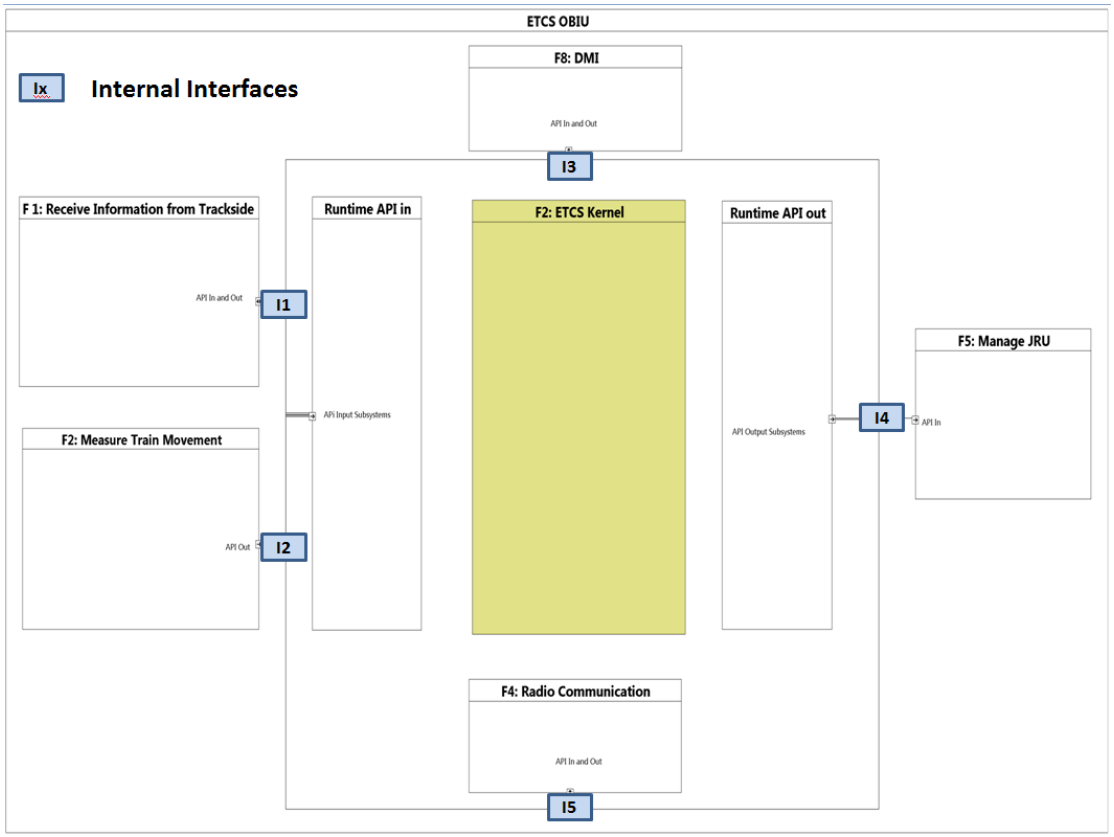


Figure 4. 2nd level system architecture view.

I1: In flow from the Balise Transmission Module (BTM or Antenna) to the "F2 ETCS Kernel" trough Runtime API in. Transmitted data are information from the Eurobalise.

I2: In flow from the Odometrie (ODO) to the "F2 ETCS Kernel" trough Runtime API in. Transmitted data are information from the movement of the train.

I3: In- and Out flow between the DMI Controller and the "F2 ETCS Kernel" trough Runtime API in and out. Transmitted data are information of driver action and display. See description in figure of "External Interface E5".

I4: Out flow from "F2 ETCS Kernel" to the JRU manager trough Runtime API out. Transmitted data are all necessary information for a juridical recorder unit "black box".

I5: In- and Out flow between the Euroradio and "F2 ETCS Kernel" trough Runtime API in and out. Transmitted data are radio track information (RBC) and information to the track (RBC).

Part II

Design Description

3 F1: Receive Information from Trackside

3.1100 ETCS Messaging: TrackMessages

3.1.1 Component Requirements

Component name	TrackMessages::Read_P005
Link to SCADE model	https://github.com/openETCS/modeling/tree/master/model/Scade/System/ObuFunctions/ETCS_Messaging/TrackMessages
SCADE designer	Jakob Gärtner, LEA Railergy Mairamou Haman Adji, LEA Railergy
Description	TrackMessages is a library containing functionality to: • Transport TrainToTrack and TrackToTrain messages and packets using a compressed format which is conceptually close to the ETCS language as defined in Subset-026 • Compress trackside information and decompress it in the on-board unit, taking into account different baseline versions and providing transparent translation. • Compress trainside information and decompress it in the trackside simulation models, taking into account different baseline versions and providing transparent translation. As TrackMessages is a library with various components supporting all packets and messages defined in Subset-026, we have selected one exemplary function to document the concept. As only the packet/ message- related functionality is specific, this approach will allow a first understanding of the concept and the related interfaces. For a full discussion of the library, refer to the [specifc chapter? document?] The function Read_P005 extracts a packet 5 (Gradient Profile) from the compressed packets data flow, if present. It translates the integer-coded compressed data with the help of the metadata in the header section of the CompressedPackets_T formatted data flow. After performing variable-level translation and exception detection, a baseline-3 conformal packet 5 is available for use within the relevant OBU functions.

Input documents	Subset-026, Chapter 6 Subset-026, Chapter 7 Subset-026, Chapter 8 The objective of this component (the full TrackMessages library) is to provide a full formalisation of above chapters in Subset-026
Safety integrity level	4
Time constraints	n/a (for the provided example function)
API requirements	In the demonstrator context, the API is fully defined on SCADE model level. For integration with external systems (BTM, Radio, Subset-076 or Subset-94), additional conversion to/ from bit-level representation will be required

3.1.2 Interface

An overview of the interface of component [component name] is shown in Figure 5. The inputs and outputs are described in detail in Section 3.1.2.1 respectively 3.1.2.2.

105 #### 3.1.2.1 Inputs

3.1.2.1.1 Message_In

Input name	Message_In
Description	Message_In takes the compressed track-to-train messages that have either been compressed by the trackside simulation components of the TrackMessages library, or have been filled by the API. All packets that are part of the same message are transmitted within one cycle of the model's execution. Message_IN is taking the compressed packet information from the track to train dataflow.
Source	Manage_TrackSideInformation_Integration
Type	Common_Types_Pkg::CompressedPackets_T

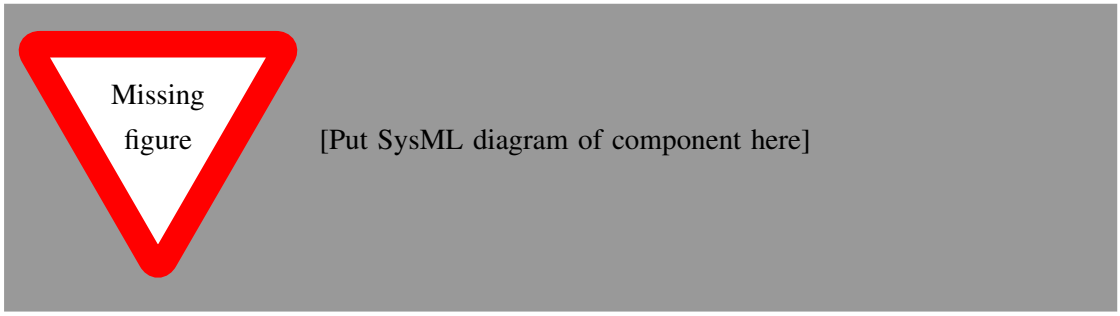


Figure 5. Component SysML diagram

Valid range of values	n/a The consistency of the metadata is checked at the input side. The ranges of the transported variables are checked at the conversion step (from integer format to SRS- conformal format)
Behaviour when value is at boundary	n/a
Behaviour for values out of valid range	n/a The content of this input is not checked, as any issues will be found at conversion level. If the metadata are not matching the search criteria the packet will be considered as non existent and will therefore be ignored.

3.1.2.2 Outputs

3.1.2.2.1 received

Output name	received
Description	Flag to indicate reception of a packet 5 from trackside in the current cycle
Destination	Any calling component
Type	bool
Valid range of values	[true | false]
Behaviour when value is at boundary	n/a
Behaviour for values out of valid range	n/a
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	n/a

3.1.2.2.2 P005_OBU_out

Output name	P005_OBU_out
Description	Gradient Profile (Packet 5) according to 7.4.2.2
Destination	Any calling operator
Type	TM::P005_OBU_T

Valid range of values

TM::P005_OBU_T is a complex data type. Values are given for each element.
Format is: Type Name: range/ list of values

- bool valid: [true | false]
- q_dir Q_DIR:
  [Q_DIR_Both_directions |
  Q_DIR_Nominal |
  Q_DIR_Reverse]
- l_packet L_PACKET: (0-8191)
- q_scale Q_SCALE:
  [ENUM_Q_SCALE_10cm |
  ENUM_Q_SCALE_1m |
  ENUM_Q_SCALE_10m]
- n_iter N_ITER: (0-33)
  (Remark: start section from the original packet is integrated into the list of sections)

The structured element sections is an array of type P005_section_enum_T. For each element, the valid range of values is as follows:

- bool valid: [true | false]
  (Remark: Check for consistency with the value of n_iter)
- d_link D_LINK: (0-32767)
- q_newcountry Q_NEWCOUNTRY:
  [TM_conversions::ENUM_Q_NEWCOUNTRY_same |
  TM_conversions::ENUM_Q_NEWCOUNTRY_not_same]
- nid_c NID_C: (0-1023)
- nid_bg NID_BG: (0-16383)
- q_linkorientation Q_LINKORIENTATION:
  [TM_conversions::ENUM_Q_LINKORIENTATION_reverse|
  TM_conversions::ENUM_Q_LINKORIENTATION_nominal]
- q_linkreaction Q_LINKREACTION:
  [TM_conversions::ENUM_Q_LINKREACTION_Train_trip |
  TM_conversions::ENUM_Q_LINKREACTION_Apply_service_brake
  TM_conversions::ENUM_Q_LINKREACTION_No_Reaction]
- q_locacc Q_LOCACC: (0-63)

Only an output structure with the structured element "valid" set to "true" is to be considered as received. If this field is set to true, the Output 1 (received) must equally be set to "true".

Behaviour when value is at boundary	n/a
Behaviour for values out of valid range	The component is prepared for the upcoming error/exception handling concept. An error flag is, at the moment, raised internally if any of the compressed input values is out of range. A hierarchical error processing is foreseen. *The types that have been defined in the package S026_7 do not provide any default/ invalid value. The following fields are therefore set to an arbitrary value upon reception of an out-of-range value from track side, and the internal error flag is raised:* • q_dir Q_DIR: set to: Q_DIR_Both_directions • q_scale Q_SCALE: set to: ENUM_Q_SCALE_10cm • q_newcountry Q_NEWCOUNTRY: set to:[TM_conversions::ENUM_Q_NEWCOUNTRY_same | TM_conversions::ENUM_Q_NEWCOUNTRY_not_same] • q_newcountry Q_NEWCOUNTRY: set to: TM_conversions::ENUM_Q_NEWCOUNTRY_not_same • q_linkorientation Q_LINKORIENTATION: set to: TM_conversions::ENUM_Q_LINKORIENTATION_reverse • q_linkreaction Q_LINKREACTION: set to: TM_conversions::ENUM_Q_LINKREACTION_Train_trip
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	n/a

3.1.3 Sub Components

3.1.3.1 Read_Packets

3.1.3.2 Component Requirements

Component name	TM_lib_internal::RECV_ReadPackets
Link to SCADE model	https://github.com/openETCS/modeling/tree/master/model/Scade/System/ObuFunctions/ETCS_Messaging/TrackMessages
SCADE designer	Jakob Gärtner, LEA Railergy

Description	RECV_ReadPackets extracts packet data information and raw compressed packet data from the compressed packets data flow, using filter criteria provided through parameter inputs: • NID_PACKET: search for a specific packet • Version Number: search for a specific version number • Q_DIR: search for packets that are only valid for a specific direction • Serial number: search for a specific packet instance, if several instances of a given packet type exist • F_Version: Flag to decide whether to evaluate or ignore packet version information. • F_id: Flag whether to evaluate or ignore packet serial number information. The operator TM_lib_internal::RECV_ReadPackets takes a set of parameter data to • 1. Search the metadata of the compressed packets data flow using the provided parameters to determine if a matching packet is contained in any given cycle • 2. Output the flag "received" exactly in any cycle a matching packet is found • 3. Output an array of compressed packet data that is filled with the data from the identified packet
Input documents	Subset-026, Chapter 7 This function is not directly traceable to Subset-026, but is built from derived requirements
Safety integrity level	4
Time constraints	n/a
API requirements	In the demonstrator context, the API is fully defined on SCADE model level. For integration with external systems (BTM, Radio, Subset-076 or Subset-94), additional conversion to/ from bit-level representation will be required

3.1.3.3 Extract Packet 5

3.1.3.4 Component Requirements

Component name	TM_conversions::trackside.C_P005_compr_onboard
Link to SCADE model	https://github.com/openETCS/modeling/tree/master/model/Scade/System/ObuFunctions/ETCS_Messaging/TrackMessages
SCADE designer	Jakob Gärtner, LEA Railergy
Description	If a matching packet 5 has been received, TM_conversions::trackside.C_P005_compr_onboard: takes the compressed packet data and converts them to an SRS conformal onboard packet format. Trailing 0 beyond the valid length of the packet are ignored.
Input documents	Subset-026, Chapter 7
Safety integrity level	4
Time constraints	n/a
API requirements	n/a

4 115 F2: ETCS Kernel

4.1 Manage_TrackSideInformation_Integration

4.1.1 Component Requirements

detail for
ntation
e trackMes-
oncept

Component name	Manage_TrackSideInformation_Integration
Link to SCADE model	https://github.com/openETCS/modeling/blob/master/model/Scade/System/ObuFunctions/ManageLocationRelatedInformation/BaliseGroup/Manage_TrackSideInformation_Integration/Manage_TrackSideInformation_Integration.etp
SCADE designer	Bernd Hekele, DB Netz AG
Description	The block "Manage_TrackSideInformation_Integration" is responsible for receiving Eurobalise telegrams and Euroradio messages from the API and performs several consistency checks on the inputs. The block collects the telegrams of balises in order to build balise group messages. Euroradio messages are always delivered as a whole message. On each message, a consistency check is performed, before the data is validated according to the driving direction of the train. In general, messages not designated for the current driving direction of the train are not forwarded to the further processing. After applying consistency checks, the data direction is validated.
Input documents	See sub-components.
Safety integrity level	4
Time constraints	The component has to be able to receive balise telegrams and radio messages according to the ETCS [**?**] performance requirements). In highspeed traffic, a group of 8 balises must be read in about 250 msec. In addition, 1 message per sec. on the radio interface is to be expected.
API requirements	Interfaces to this unit are defined in the API sections [BTM], [EURORADIO], [ODO].In these sections, also a detailed definition of the concepts implemented on those interfaces is documented.

4.1.2 Interface

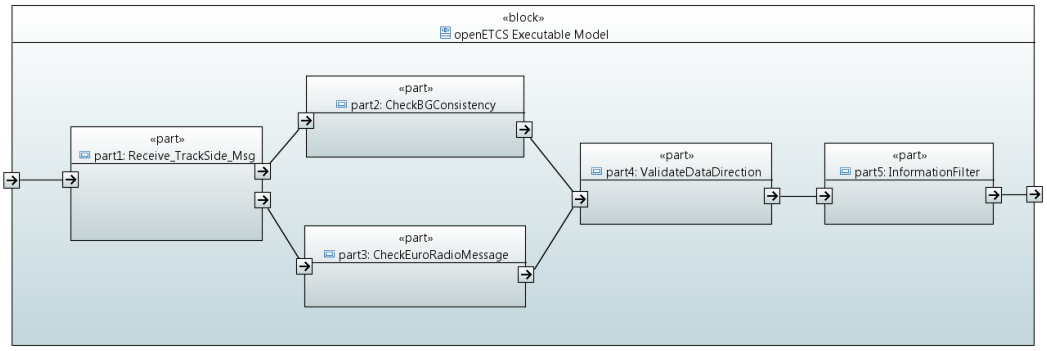


Figure 6. Manage_TrackSideInformation_Integration component SysML diagram.

120 An overview of the interface of component Manage_TrackSideInformation_Integration is shown in Figure 6. The inputs and outputs are described in detail in Section 4.1.2.1 respectively 4.1.2.2. Sub components are described in Section 4.1.3.

4.1.2.1 Inputs

4.1.2.1.1 fullChecks

Input name	fullChecks
Description	Indicates, if all checks on the message should be performed.
Source	This item is only relevant in verification phases. In a real system checks are always activated.
Type	bool
Valid range of values	**true** All checks are performed. **false** Component InformationFilter is deactivated.
Behaviour when value is at boundary	n/a
Behaviour for values out of valid range	n/a
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

125 ##### 4.1.2.1.2 Receive_trackSide_Message

Input name	API_trackSide_Message
Description	Track side message received from the API. The API performs pre-processing of RTM and BTM messages and deliveres a maximum of a single message per cycle. The structure of this message is defined in the API [BTM] and [EURORADIO] sections.
Source	API
Type	API_Msg_Pkg::API_TrackSideInput_T

Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

4.1.2.1.3 ActualOdometry

Input name	ActualOdometry
Description	Provided by the external odometry module of the train. It contains relative location information with inaccuracies.
Source	Odometer
Type	Obu_BasicTypes_Pkg::odometry_T
Valid range of values	[Complete list of valid values]
Behaviour when value is at boundary	[Description of components behaviour when input value is at boundary]
Behaviour for values out of valid range	[Description of components behaviour when input value is out of valid range]
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

4.1.2.1.4 reset

Input name	reset
Description	To delete all data stored in the module (e.g. collected balise telegrams, which do not yet form a complete message), a reset input can be used. If the input is set to true, all data kept in the module is deleted and no input is accepted.
Source	Environment
Type	bool
Valid range of values	**true** All data kept in the module is deleted and no input is accepted. **false** No action. Data at input is accepted.
Behaviour when value is at boundary	[Description of components behaviour when input value is at boundary]
Behaviour for values out of valid range	[Description of components behaviour when input value is out of valid range]

Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

4.1.2.1.5 trainPosition

Input name	trainPosition
Description	Contains the current position of the train.
Source	CalculateTrainPosition
Type	TrainPosition_Types_Pck::trainPosition_T
Valid range of values	
Behaviour when value is at boundary	[Description of components behaviour when input value is at boundary]
Behaviour for values out of valid range	[Description of components behaviour when input value is out of valid range]
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

4.1.2.1.6 modeAndLevel

Input name	modeAndLevel
Description	Provides the current level and mode of the EVC.
Source	ModeAndLevel
Type	BG_Types_Pkg::ModeAndLevelStatus_T
Valid range of values	[Complete list of valid values]
Behaviour when value is at boundary	[Description of components behaviour when input value is at boundary]
Behaviour for values out of valid range	[Description of components behaviour when input value is out of valid range]
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

4.1.2.1.7 tNvContact

Input name	tNvContact

Description	For monitoring the safe radio connection, this national value is needed as an input.
Source	Database
Type	Obu_BasicTypes_Pkg::T_internal_Type
Valid range of values	[Complete list of valid values]
Behaviour when value is at boundary	[Description of components behaviour when input value is at boundary]
Behaviour for values out of valid range	[Description of components behaviour when input value is out of valid range]
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

4.1.2.1.8 lastRelevantEventTimestamp

Input name	lastRelevantEventTimestamp
Description	For monitoring the safe radio connection, it is necessary that the time between two packets is less than the value of T_NVCONTACT. In situations like level-changes or announced radio holes, not the timestamp of the last message is relevant for comparison, but the timestamp of the last relevant event. This can for example be the timestamp of the level change or the timestamp of the moment, when the train was passing the end of the radiohole. For performing this check, the timestamp of the last relevant event is provided to the model as an T_internal_Type-type.
Source	Database
Type	Obu_BasicTypes_Pkg::T_internal_Type
Valid range of values	[Complete list of valid values]
Behaviour when value is at boundary	[Description of components behaviour when input value is at boundary]
Behaviour for values out of valid range	[Description of components behaviour when input value is out of valid range]
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

4.1.2.1.9 connectionStatus

Input name	connectionStatus
Description	Status information about the radio connection. The information is needed to perform the timing check, which depends on the connection state.
Source	ManageRadioCommunication
Type	Radio_Types_Pkg::sessionStatus_Type
Valid range of values	**DISCONNECTED** The OBU is currently not connected to a RBC. **CONNECTING** The OBU is currently connecting to the RBC. Received messages belong to the process of establishing a connection. **CONNECTION_ESTABLISHED** The connection to the RBC is established.
Behaviour when value is at boundary	[Description of components behaviour when input value is at boundary]
Behaviour for values out of valid range	[Description of components behaviour when input value is out of valid range]
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

4.1.2.1.10 inSupervisingRbcId

Input name	inSupervisingRbcId
Description	For the sub component InformationFilter, the information which radio messages are sent by the supervising RBC is needed. To recognize these messages, the identifier of the supervising RBC is needed.
Source	Database
Type	int
Valid range of values	[Complete list of valid values]
Behaviour when value is at boundary	[Description of components behaviour when input value is at boundary]
Behaviour for values out of valid range	[Description of components behaviour when input value is out of valid range]

Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

4.1.2.1.11 inAnnouncedBGs

Input name	inAnnouncedBGs
Description	Provides information about balise groups which will be passed by the train soon. This information is generated by Calculate Train Position based on the linking information received from trackside.
Source	CalculateTrainPosition
Type	TrainPosition_Types_Pck::positionedBGs_T
Valid range of values	[Complete list of valid values]
Behaviour when value is at boundary	[Description of components behaviour when input value is at boundary]
Behaviour for values out of valid range	[Description of components behaviour when input value is out of valid range]
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

4.1.2.1.12 q_nvlocacc

Input name	q_nvlocacc
Description	The national value determines the location accuracy.
Source	Database
Type	Q_NVLOCACC
Valid range of values	[Complete list of valid values]
Behaviour when value is at boundary	[Description of components behaviour when input value is at boundary]
Behaviour for values out of valid range	[Description of components behaviour when input value is out of valid range]
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

4.1.2.2 Outputs

4.1.2.2.1 outputMessage

Output name	outputMessage
Description	Combines both balise and radio messages to one common datatype. This datatype contains all variables and packets, which are possible for the given scenario.
Destination	[Name of the destination component(s)]
Type	Common_Types_Pkg::ReceivedMessage_T
Valid range of values	[Complete list of valid values]
Behaviour when value is at boundary	[Description of components behaviour when output value is at boundary]
Behaviour for values out of valid range	[Description of components behaviour when output value is out of valid range]
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

4.1.2.2.2 ApplyServiceBrake

Output name	ApplyServiceBrake
Description	Indicates if the balise group the train just passed could not be processed correctly. The check results in the request for a service break.
Destination	[Name of the destination component(s)]
Type	bool
Valid range of values	[Complete list of valid values]
Behaviour when value is at boundary	[Description of components behaviour when output value is at boundary]
Behaviour for values out of valid range	[Description of components behaviour when output value is out of valid range]
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

4.1.2.2.3 BadBaliseMessageToDMI

Output name	BadBaliseMessageToDMI

Description	Information to be passed to the DMI to indicate the reception of a "bad balise" to the driver.
Destination	DMI
Type	bool
Valid range of values	**true** ??? **false** ???
Behaviour when value is at boundary	[Description of components behaviour when output value is at boundary]
Behaviour for values out of valid range	[Description of components behaviour when output value is out of valid range]
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

4.1.2.2.4 errorLinkedBG

Output name	errorLinkedBG
Description	[Brief description of the output]
Destination	[Name of the destination component(s)]
Type	[Type of the output]
Valid range of values	**true** An error in a linked balise group was detected. **false** No error in a linked balise group was detected.
Behaviour when value is at boundary	[Description of components behaviour when output value is at boundary]
Behaviour for values out of valid range	[Description of components behaviour when output value is out of valid range]
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

4.1.2.2.5 errorUnlinkedBG

Output name	errorUnlinkedBG

Description	[Brief description of the output]
Destination	[Name of the destination component(s)]
Type	bool
Valid range of values	**true** An error in a unlinked balise group was detected. **false** No error in a unlinked balise group was detected.
Behaviour when value is at boundary	[Description of components behaviour when output value is at boundary]
Behaviour for values out of valid range	[Description of components behaviour when output value is out of valid range]
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

4.1.2.2.6 passedBG

Output name	passedBG
Description	Provides the received balise group message in a special format needed by the component CalculateTrainPosition.
Destination	[Name of the destination component(s)]
Type	BG_Types_Pkg::passedBG_T
Valid range of values	[Complete list of valid values]
Behaviour when value is at boundary	[Description of components behaviour when output value is at boundary]
Behaviour for values out of valid range	[Description of components behaviour when output value is out of valid range]
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

4.1.2.2.7 outPositionParams

Output name	outPositionParams
Description	Provides the parameters for the position report in a special format needed by the component ProvidePositionReport.

Destination	[Name of the destination component(s)]
Type	Common_Types_Pkg::PositionReportParameter_T
Valid range of values	[Complete list of valid values]
Behaviour when value is at boundary	[Description of components behaviour when output value is at boundary]
Behaviour for values out of valid range	[Description of components behaviour when output value is out of valid range]
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

4.1.2.2.8 outRadioManagement

Output name	outRadioManagement
Description	Provides the messages for radio session management in a special format needed by the component ManagementOfRadioCommunication.
Destination	[Name of the destination component(s)]
Type	Common_Types_Pkg::radioManagementMessage_T
Valid range of values	[Complete list of valid values]
Behaviour when value is at boundary	[Description of components behaviour when output value is at boundary]
Behaviour for values out of valid range	[Description of components behaviour when output value is out of valid range]
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

4.1.2.2.9 radioSequenceError

Output name	radioSequenceError
Description	[Brief description of the output]
Destination	[Name of the destination component(s)]
Type	bool

Valid range of values	**true** A sequence error or a timeout has been detected in the radio message. **false** No error in the radio message sequence was detected.
Behaviour when value is at boundary	[Description of components behaviour when output value is at boundary]
Behaviour for values out of valid range	[Description of components behaviour when output value is out of valid range]
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

4.1.2.2.10 radioMessageConsistencyError

Output name	radioMessageConsistencyError
Description	[Brief description of the output]
Destination	[Name of the destination component(s)]
Type	bool
Valid range of values	**true** A consistency error has been detected in the radio message. **false** No consistency error in the radio message was detected.
Behaviour when value is at boundary	[Description of components behaviour when output value is at boundary]
Behaviour for values out of valid range	[Description of components behaviour when output value is out of valid range]
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

4.1.3 Sub Components

4.1.3.1 Receive_TrackSide_Msg

4.1.3.1.1 Component Requirements

Component name	Receive_TrackSide_Msg

Link to SCADE model	https://github.com/openETCS/modeling/tree/master/model/Scade/System/ObuFunctions/ManageLocationRelatedInformation/BaliseGroup/Receive_TrackSide_Msg
SCADE designer	[Name, affiliation]
Description	This function defines the interface of the OBU model to the openETCS generic API for Eurobalise and Euroradio messages. On the interface, either a valid telegram/message is provided or a telegram/message is indicated which could not be received correct when passing the balise or receiving the radio message. The function passes a balise telegram without major changes of the information to the next entity for collecting the balise group information. This entity collects telegrams received via the interface into Balise Group Information. In case of a radio message, the message is converted to an internal format for further processing and passed without changing the information contained. • The decoding of balises is done at the API. Also, packets received via the interface are already transformed into a usable shape. • Only packets used inside the current model are passed via the interface. • Treatment of Packet 5: Linking Information. Linking Information is added to the linking array starting from index 0 without gaps. Used elements are marked as valid. Elements are sorted according to the order given by the telegram sequence. • Telegrams received as invalid are passed to the "Check-Function" to process errors in communication with the track side according to the requirements and in a single place. Telegrams are added to the telegram array starting from index 0 without gaps. Used elements are marked as valid. Elements are stored according to the order given by the telegram sequence. • This function does not process information from the packets. The information is passed to the check without further processing of the values.
Input documents	Subset-026, Chapter 7 and 8: Definition of the Balise Telegram Subset-026, Chapter 4.2.2, 4.2.4, 4.2.9: Interface to the BTM Subset-026, Chapter 3.4.1 - 3.4.3, 3.16.2: Handling of Balise Telegrams Subset-026, Chapter 3.16.2: Check of the balise group Subset-026, Chapter 3.4.2: Determining the orientation Subset-026, Chapter 4.5.2 Active Functions Table Subset-026, Chapter 8.4.4: Rules for Euroradio messages
Safety integrity level	4
Time constraints	n/a

API requirements	n/a

4.1.3.1.2 Interface

For an overview of the interface of this internal component we refer to the SCADE model (c.f. link above) respectively the SCADE generated documentation.

4.1.3.2 CheckBGConsistency

4.1.3.2.1 Component Requirements

Component name	CheckBGConsistency
Link to SCADE model	https://github.com/openETCS/modeling/tree/master/model/Scade/System/ObuFunctions/ManageLocationRelatedInformation/BaliseGroup/CheckBGConsistency
SCADE designer	[Name, affiliation]
Description	This function verifies the completeness and correctness of the received messages from balise groups. A message consists of at least a telegram and a maximum of 8 telegrams. • A message is still complete and correct, if a telegram is missing (or not decoded or incomplete decoded), and this telegram is duplicated within the balise group and the duplicating one is correctly read. • By more than one telegram, the order of the telegrams must be either ascending (nominal) or descending(reverse). • A message is correct, if all message counters (M MCUNT) do not equal 254 (that means: The telegram never fits any message of the group). A message counter can be equal 255 (that means: The telegram fits with all telegrams of the same balise group) and all other values must be the same. The orientation of the BG will also be calculated in this block. The check, if the message has been received in due time and the right at the right expected location, will be performed in "Calculate Train Position". The checks on the validity of the data in the packets and the validity with respect to the direction of motion will be performed in other modules, e.g. "Validate Data Direction".
Input documents	Subset-026, Chapter 7 and 8: Definition of the Balise Telegram Subset-026, Chapter 3.4.1-3, 3.16.2: Handling of Balise Telegrams Subset-026, Chapter 3.16.2: Check of the balise group Subset-026, Chapter 4.5.2: Active Functions Table
Safety integrity level	4
Time constraints	n/a
API requirements	n/a

4.1.3.2.2 Interface

For an overview of the interface of this internal component we refer to the SCADE model (c.f. link above) respectively the SCADE generated documentation.

4.1.3.3 CheckEuroradioMessage

4.1.3.3.1 Component Requirements

Component name	CheckEuroradioMessage
Link to SCADE model	https://github.com/openETCS/modeling/tree/b9c31ce6fdf702b412bbeab3032a8a4dc7c92e5c/model/Scade/System/ObuFunctions/ManageLocationRelatedInformation/BaliseGroup/CheckEuroRadioMessage
SCADE designer	Stefan Karg, DB Netz AG
Description	The component "CheckEuroradioMessage" performs several checks on the received radio message. These checks include checking of the message sequence, completeness of messages. Invalid messages are marked as invalid in the message header.
Input documents	Subset-026, Chapter 3.16 Subset-026, Chapter 8.4.4
Safety integrity level	4
Time constraints	n/a
API requirements	n/a

4.1.3.3.2 Interface

For an overview of the interface of this internal component we refer to the SCADE model (c.f. link above) respectively the SCADE generated documentation.

4.1.3.4 ValidateDataDirection

4.1.3.4.1 Component Requirements

Component name	CheckEuroradioMessage
Link to SCADE model	https://github.com/openETCS/modeling/tree/master/model/Scade/System/ObuFunctions/ManageLocationRelatedInformation/BaliseGroup/ValidateDataDirection
SCADE designer	???

Description	The component filters an input message in order to mark all elements as invalid, which are not designated for the current driving direction of the train. • The operator contains two processing paths for different message types. Radio messages and balise group messages are handeled in a different way. For validating the data direction of a radio message, the check is performed using the balise group referenced in the radio message header as relevant balise group. For balise group message, the LRBG is used. • The metadata of packets, which are recognized as not valid for the current driving direction, is invalidated.

Input documents	Subset-026, Chapter 3.6.3
Safety integrity level	4
Time constraints	n/a
API requirements	n/a

165 #### 4.1.3.4.2 Interface

For an overview of the interface of this internal component we refer to the SCADE model (c.f. link above) respectively the SCADE generated documentation.

4.1.3.5 InformationFilter

4.1.3.5.1 Component Requirements

Component name	CheckEuroradioMessage
Link to SCADE model	https://github.com/openETCS/modeling/tree/master/model/Scade/System/ObuFunctions/ManageLocationRelatedInformation/BaliseGroup/InformationFilter
SCADE designer	Alexander Stante, FhG

Description	The filter receives track information (balise and radio) and filters them depending of the mode, level and source of the message. Only messages that pass the filter are valid and should be considered by other ETCS subsystems. Figure 7 shows the highlevel decomposition of the functionality. The filter is consists of four components: FirstFilter, SecondFilter, ThirdFilter and TransitionBuffer. **FirstFilter** This filter performs filtering of messages based on the current ETCS level. The decisions taken process is described via a big decision table which contains rows for every packet and columns for every ETCS level. This table encodes also if certain additional information is necessary to filter a message like pending ETCS Level transitions. Based on this filter packets of an incoming message is either rejected, accepted or the whole message is put in the TransitionBuffer. Messages are put in the TransitionBuffer if there is an announced level transition and the received message is only valid for the upcoming level. **SecondFilter** The SecondFilter mainly considers messages that are received via Euroradio. Certain messages are directly rejected while other may be stored in the TransitionBuffer. The buffer is used to store messages that are received from non supervising RBCs, but will be reevaluated after a RBC transition. **ThirdFilter** The last filter is functionally very similiar the the FirstFilter, however it filters depending on the mode. It also contains a decision table with rows for every packet but the columns are modes. **TransitionBuffer** The InformationFilter uses two TransitionBuffers. One is used to store up to three messages for the ETCS level transition and the other buffer is used for RBC transitions. The buffer is designed as a ring buffer and message are read in FIFO order.
Input documents	Subset-026, Chapter 4.8
Safety integrity level	4
Time constraints	n/a
API requirements	n/a

4.1.3.5.2 Interface

For an overview of the interface of this internal component we refer to the SCADE model (c.f. link above) respectively the SCADE generated documentation.

4.2 Speed and Distance Monitoring

4.2.1 Component Requirements

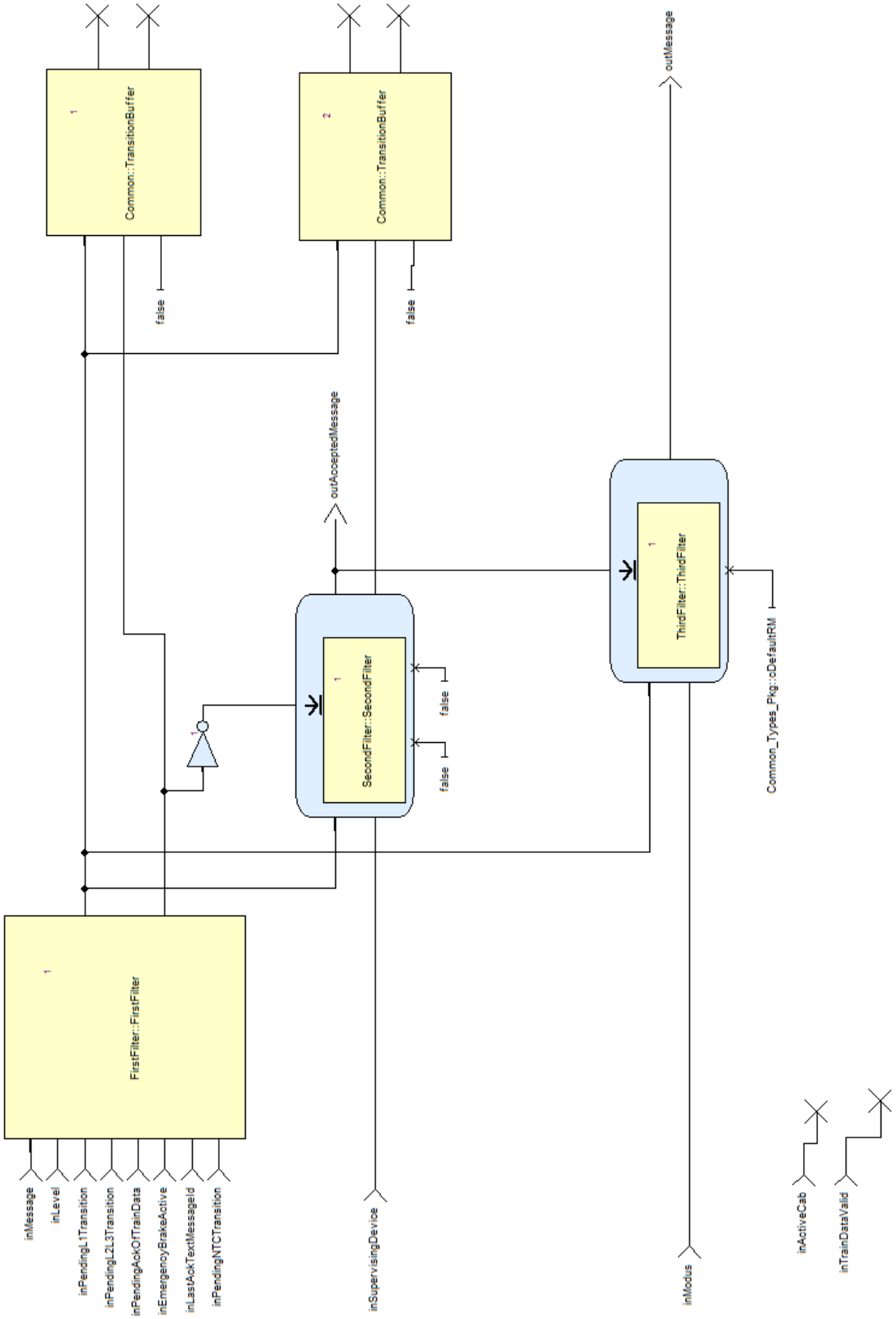


Figure 7. High level overview of the InformationFilter components.

Component name	SpeedSupervision_Integration (“SDM”)
Link to SCADE model	https://github.com/openETCS/modeling/tree/master/model/Scade/System/ObuFunctions/SpeedSupervison
SCADE designer	Benjamin Beichler, University of Rostock; Christian Stahl, TWT; Thorsten Schulz, University of Rostock
Description	The task of SDM is to monitor the speed of the train and the train location and as such to ensure that the speed remains within the given speed and distance limits. This block is based on [1, Chapt. 3.13]. The integration node “SpeedSupervision_Integration” takes as input (1) movement related information such as train speed, train position and acceleration, (2) train related information such as brake information and train length, and (3) track related information such as speed and distance limits and national values. Based on this information a speed profile is calculated. Speed restrictions create target speeds (targets) that have to be followed. For each such target braking curves are generated to supervise at which location of the track the train must apply the brake. In case of no target restrictions the train may accelerate to the supervised maximum speed of the speed profile. These calculations lead to commands being sent to the driver and the brake system. The functionality is modeled using eight operators, as shown in Figure 8, which are explained below. The current status of the analysis of “SDM” and a functional breakdown can be found in a separate document, `SpeedSupervision_analysis.pdf`.
Input documents	Subset-026, Chapter 3.13: Speed and distance monitoring
Safety integrity level	4
Time constraints	n/a
API requirements	n/a

4.2.2 Interface

An overview of the interface of component SpeedSupervision_integration is shown in Figure 8. The inputs and outputs are described in detail in Section 4.2.2.1 respectively 4.2.2.2. Sub components are described in Section 4.2.3.

4.2.2.1 Inputs

4.2.2.1.1 NationalValues

Input name	NationalValues
Description	This input is packet 3 of [1, Chapt. 8], describing the national values.
Source	Track Atlas Data

Type	P3_NationalValues_T
Valid range of values	P3_NationalValues_T is a complex data type
Behaviour when value is at boundary	as specified in SRS
Behaviour for values out of valid range	currently not checked
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	not checked; node must not be called without reasonable National Value

4.2.2.1.2 TrainPosition

Input name	TrainPosition
Description	This input is the current train position.
Source	Manage Track Data
Type	trainPosition_T
Valid range of values	complex data type
Behaviour when value is at boundary	not checked, may overflow
Behaviour for values out of valid range	currently not checked
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	not checked; node must not be called without reasonable position data

4.2.2.1.3 odometry

Input name	odometry
Description	This input is the odometry data.
Source	Odometry
Type	odometry_T
Valid range of values	complex data type used fields are: - acceleration: Obu_BasicTypes_Pkg::A_internal_Type. No valid range defined, neither checked. - motionState: [noMotion | Motion] (enum type)

Behaviour when value is at boundary	possible overflow not evaluated
Behaviour for values out of valid range	not checked
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	not handled, valid data is expected for valid function

4.2.2.1.4 m_level

Input name	m_level
Description	This input is the current level of the train. (will be removed in next release!)
Source	Mode and Level
Type	M_LEVEL
Valid range of values	enum type, valid range is ensured at compile time
Behaviour when value is at boundary	n/a
Behaviour for values out of valid range	n/a
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	n/a

4.2.2.1.5 trainProps

Input name	trainProps
Description	This input is a set of train related properties.
Source	Database
Type	trainProperties_T
Valid range of values	complex type used fields are: d_baliseAntenna_2_frontend.nominal: Obu_BasicTypes_Pkg::L_internal_Type No valid range defined, neither checked.
Behaviour when value is at boundary	n/a

Behaviour for values out of valid range	n/a
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	Value is only evaluated in Level 1. Low values (e.g. invalid-default 0) will lead to early trip, brake and alike. Larger values will lead to late braking, possibly numeric overflow.

185 #### 4.2.2.1.6 MRSP

Input name	MRSP
Description	This input is the most restrictive speed profile.
Source	Track Atlas
Type	MRSP_Profile_t
Valid range of values	complex type
Behaviour when value is at boundary	n/a
Behaviour for values out of valid range	n/a
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	at least one valid entry is expected (the maximum vehicle speed), else a trip shall be commanded.

4.2.2.1.7 MA

Input name	MA
Description	This input is a movement authority.
Source	???
Type	MAs_t
Valid range of values	complex type
Behaviour when value is at boundary	n/a
Behaviour for values out of valid range	n/a
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	on .valid = false, brake/trip should be commanded.

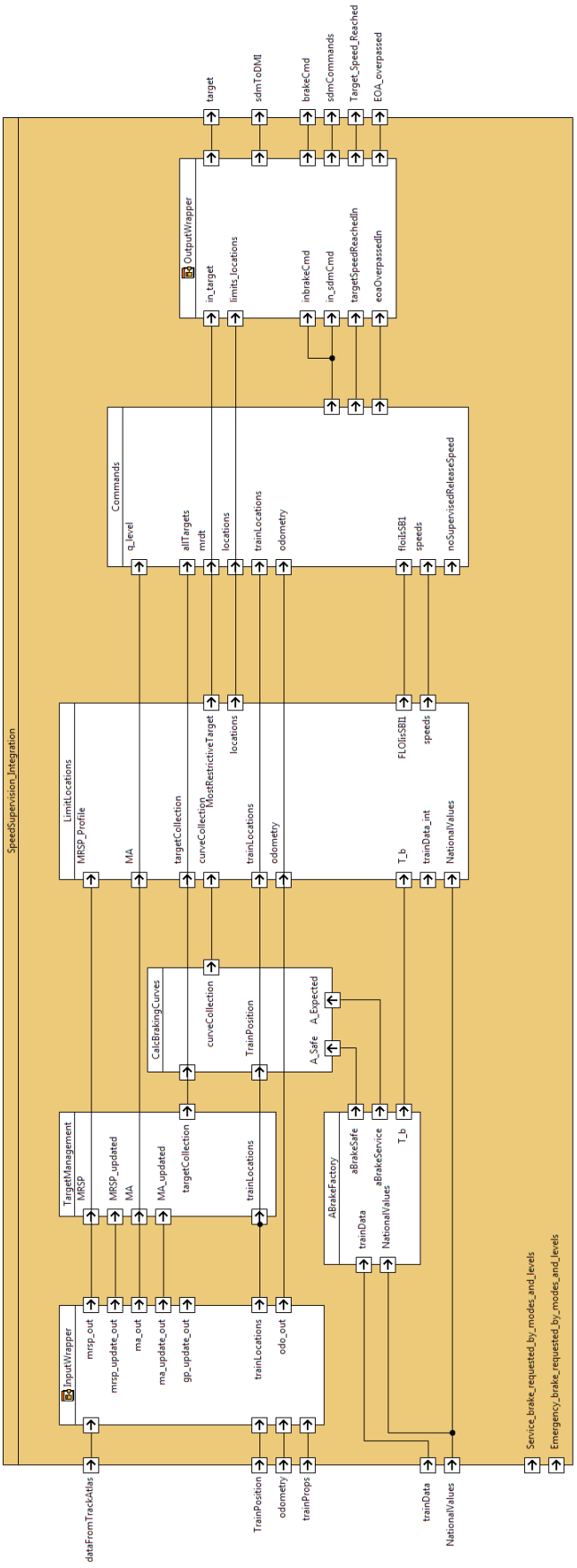


Figure 8. Structure of component SpeedSupervision_Integration.

4.2.2.1.8 MA_updated

Input name	MA_updated
Description	This flag is true if the movement authority has been updated in this clock cycle and false otherwise.
Source	internal
Type	bool
Valid range of values	$false, true$
Behaviour when value is at boundary	limited range, check at compile time
Behaviour for values out of valid range	limited range, check at compile time
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	n/a

4.2.2.1.9 MRSP_updated

Input name	MRSP_updated
Description	This flag is true if the most restrictive speed profile has been updated in this clock cycle and false otherwise.
Source	internal
Type	bool
Valid range of values	$false, true$
Behaviour when value is at boundary	limited range, check at compile time
Behaviour for values out of valid range	limited range, check at compile time
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	n/a

4.2.2.2 Outputs

4.2.2.2.1 sdmToDMI

Output name	sdmToDMI
Description	This output contains information about different speeds and positions, on the one hand and the current supervision status, on the other hand. This information shall be displayed to the driver.
Destination	DMI
Type	speedSupervisionForDMI_T
Valid range of values	n/a
Behaviour when value is at boundary	n/a
Behaviour for values out of valid range	n/a
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	.valid can be false

4.2.2.2.2 target

Output name	target
Description	This output is the most restrictive displayed target (MRDT).
Destination	[Name of the destination component(s)]
Type	Target_T
Valid range of values	n/a
Behaviour when value is at boundary	n/a
Behaviour for values out of valid range	n/a
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	.valid may be false if no target is supervised or known, other values of this output must be ignored then.

4.2.2.2.3 sdmCommands

Output name	sdmCommands
Description	This output gives some intermediate results of operator SDM_Commands. It is currently used for test purposes only.

Destination	[Name of the destination component(s)]
Type	SDM_Commands_T
Valid range of values	n/a
Behaviour when value is at boundary	n/a
Behaviour for values out of valid range	n/a
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	overall .valid is always set, individual speeds have their corresponding valid flag.

4.2.2.2.4 brakeCmd

Output name	brakeCmd
Description	This output is the brake command, indicating whether performing the service brake or the emergency brake have been commanded.
Destination	[Name of the destination component(s)]
Type	Brake_command_T (enum)
Valid range of values	brake_signal_command_not_defined, apply_brake, release_brake
Behaviour when value is at boundary	n/a
Behaviour for values out of valid range	n/a
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	n/a

4.2.2.2.5 EOA_overpassed

Output name	EOA_overpassed
Description	This output is true if the end of authority has been overpassed and false otherwise.
Destination	[Name of the destination component(s)]
Type	bool
Valid range of values	[false, true]

Behaviour when value is at boundary	n/a
Behaviour for values out of valid range	n/a
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	n/a

4.2.2.2.6 Target_Speed_Reached

Output name	Target_Speed_Reached
Description	This output is true if the current speed is greater than or equal the target speed and false otherwise.
Destination	[Name of the destination component(s)]
Type	bool
Valid range of values	[false/true]
Behaviour when value is at boundary	n/a
Behaviour for values out of valid range	n/a
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	n/a

4.2.3 Sub Components

4.2.3.1 Receive_TrackSide_Msg

4.2.3.1.1 Component Requirements

Component name	SDM_InputWrapper
Link to SCADE model	https://github.com/openETCS/modeling/tree/master/model/Scade/System/ObuFunctions/SpeedSupervison/SpeedSupervision_Integration
SCADE designer	Benjamin Beichler, Thorsten Schulz, University of Rostock

Description	The motivation for this operator is to convert all inputs of SDM that contain information about length, speed, distance, and acceleration defined as integer into `real` to allow automatically the highest precision in the calculations by the meaning of floating point operations. In addition, to ease the modeling, inside block "Speed Supervision" only units meters ($[m]$), seconds($[s]$), meters per second($[\frac{m}{s}]$), and meters per square second($[\frac{m}{s^2}]$) are used. This operator forwards input messages, takes data from complex data types or transforms inputs messages into an internal type thereby converting int to real.
Input documents	Subset-026, Chapter 3.13, (not specific, helper function)
Safety integrity level	4
Time constraints	n/a
API requirements	n/a

4.2.3.1.2 Interface

200 For an overview of the interface of this internal component we refer to the SCADE model (c.f. link above) respectively the SCADE generated documentation.

4.2.3.2 TargetManagement

4.2.3.2.1 Component Requirements

Component name	TargetManagement
Link to SCADE model	https://github.com/openETCS/modeling/tree/master/model/Scade/System/ObuFunctions/SpeedSupervison/TargetManagement
SCADE designer	Benjamin Beichler, Thorsten Schulz University of Rostock

Description	This operator calculates/updates the list of targets to be supervised by SDM. Taking the current movement authority, the most restrictive speed profile and the current maximum safe front end position as an input, the operator outputs a list of MRSP-Targets and a Limit-of-Authority-Target (LoA) or, if an End of Authority is known, the End-of-Authority-Target (EoA) and the Supervised Location (SvL). Since LoA and SvL are mutually exclusive but both result in a trip-target, they use the same flow. **Derivation of Targets from Movement Authority** The *Movement Authority* creates three types of targets: **Limit of Authority (LoA)** if the End of Authority is not yet known to the RBC and is accompanied by a speed limit for the given location. **End Of Authority (EoA)** requires the train to stop and creates a Service Brake Target **Supervised Location (SvL)** is derived from the EoA but results in an emergency brake target and on passing in a trip. The SvL may be offset from the EoA to the Overlap (OL) or the Dangerpoint (DP). **Derivation of Targets from MRSP** According to [1, Chapt. 3.13.8.2], every speed decrease of the MRSP is used to derive a target. Therefore in every cycle in which the MRSP is updated, the operator iterates through the entire MRSP searching for all MRSP targets. For this purpose, every element of the MRSP is compared with its successor. **Update of Targets** In every cycle the operator monitors whether all targets are already passed. To this end, it iterates over the list of targets comparing the current front end position with the target's location.
Input documents	Subset-026, Chapter 3.13.8.2: Determination of the supervised targets
Safety integrity level	4
Time constraints	n/a
API requirements	n/a

4.2.3.2.2 Interface

205 For an overview of the interface of this internal component we refer to the SCADE model (c.f. link above) respectively the SCADE generated documentation.

4.2.3.3 CalcBrakingCurves_Integration

4.2.3.3.1 Component Requirements

Component name	CalcBrakingCurves_Integration

Link to SCADE model	https://github.com/openETCS/modeling/tree/master/model/Scade/System/ObuFunctions/SpeedSupervison/CalcBrakingCurves
SCADE designer	Benjamin Beichler, University of Rostock
Description	For each type of target a certain braking curve has to be calculated. This curve enables proactive monitoring of the train's speed. A reverse lookup on this braking curve indicates, where the train has to start braking given the current speed. The braking curve does not depend on the actual train status. As a consequence the braking curve stays constant over time. As a legitimate simplification the calculation of the braking curve is not extended past the estimated front end position of the train.
Input documents	Subset-026, Chapter 3.13.8.3: Emergency Brake Deceleration curves (EBD) Subset-026, Chapter 3.13.8.4: Service Brake Deceleration curves (SBD) Subset-026, Chapter 3.13.8.5: Guidance curves (GUI)
Safety integrity level	4
Time constraints	n/a
API requirements	n/a

4.2.3.3.2 Interface

210 For an overview of the interface of this internal component we refer to the SCADE model (c.f. link above) respectively the SCADE generated documentation.

4.2.3.4 SDMLimitLocations

4.2.3.4.1 Component Requirements

Component name	SDMLimitLocations
Link to SCADE model	https://github.com/openETCS/modeling/tree/master/model/Scade/System/ObuFunctions/SpeedSupervison/SDM_TargetLimits
SCADE designer	Thorsten Schulz, University of Rostock

Description	This operator calculates the various locations and speeds needed to determine the speed and distance monitoring commands. The current implementation of functionality is stateless and requires a complete recalculation each cycle. This operator gathers all necessary input values and computes some frequently used intermediate values in the operators `surplusTractionDeltas` and v_{bec}. The other input preparation operator is the `TargetSelector` whose main task is to dissect the list of targets to find the Most Restrictive Target. The accompanying braking curves are extracted and promoted to trailing location calculations. Also the special values of the EOA are exposed. The operator creates the requested values for the commands package. These are in particular the preindication locations for EBD and SBD based targets, the release speed monitoring start locations, the locations for target speed monitoring of the I-, W-, P- and FLOI-curve, the related FLOI speed and the location of the permitted speed supervision limit. Included in the output are also certain flags for the validity of linked values.
Input documents	Subset-026, Chapter 3.13.9: Supervision Limits Subset-026, Chapter 5.3.1.2: f_{41} – accuracy of speed known onboard Subset-026, Chapter 3.13.10: Monitoring Commands as reference for required outputs of this module
Safety integrity level	4
Time constraints	n/a
API requirements	n/a

4.2.3.4.2 Interface

215 For an overview of the interface of this internal component we refer to the SCADE model (c.f. link above) respectively the SCADE generated documentation.

4.2.3.5 CalcSpeeds

4.2.3.5.1 Component Requirements

Component name	CalcSpeeds
Link to SCADE model	???
SCADE designer	Benjamin Beichler, University of Rostock
Description	This operator calculates some speeds needed to determine the speed and distance monitoring commands. This operator will be integrated into other operators in the next iteration.
Input documents	Subset-026, Chapter 3.8: Movement authority
Safety integrity level	4

Time constraints	n/a
API requirements	n/a

4.2.3.5.2 Interface

220 For an overview of the interface of this internal component we refer to the SCADE model (c.f. link above) respectively the SCADE generated documentation.

4.2.3.6 ReleaseSpeed_Selection

4.2.3.6.1 Component Requirements

Component name	ReleaseSpeed_Selection
Link to SCADE model	???
SCADE designer	Thorsten Schulz, University of Rostock
Description	This operator outputs the release speed which can be given either by the national values or the movement authority. This operator will be integrated into other operators in the next iteration.
Input documents	Subset-026, Chapter 3.8: Movement authority
Safety integrity level	4
Time constraints	n/a
API requirements	n/a

4.2.3.6.2 Interface

225 For an overview of the interface of this internal component we refer to the SCADE model (c.f. link above) respectively the SCADE generated documentation.

4.2.3.7 SDM_Commands

4.2.3.7.1 Component Requirements

Component name	SDM_Commands
Link to SCADE model	https://github.com/openETCS/modeling/tree/master/model/Scade/System/ObuFunctions/SpeedSupervison/SDM_Commands
SCADE designer	Christian Stahl, TWT; (cur. maintainer Thorsten Schulz, University of Rostock)

Description	This operator models the speed and distance monitoring commands. More precisely, it triggers the service or emergency brake and outputs the current supervision status of the OBU together with information on speeds and locations to the driver. The OBU can be in any of three types of speed and distance monitoring modes: ceiling speed monitoring, release speed monitoring and target speed monitoring. We use a state machine to model the switching between the three modes: each state models a mode and a transition between to states is enabled if the condition two switch between the two corresponding modes is evaluated to true. In each mode, the OBU can be in up to five different supervision stati. The behavior of changing from one status to another is also modeled as a state machine. As a result, the model is a hierarchical state machine.
Input documents	Subset-026, Chapter 3.13.10: Speed and distance monitoring commands
Safety integrity level	4
Time constraints	n/a
API requirements	n/a

4.2.3.7.2 Interface

230 For an overview of the interface of this internal component we refer to the SCADE model (c.f. link above) respectively the SCADE generated documentation.

4.2.3.8 SDM_OutputWrapper

4.2.3.8.1 Component Requirements

Component name	SDM_OutputWrapper
Link to SCADE model	https://github.com/openETCS/modeling/tree/master/model/Scade/System/ObuFunctions/SpeedSupervison/SpeedSupervision_Integration
SCADE designer	Benjamin Beichler, Thorsten Schulz, University of Rostock
Description	This operator is the counterpart to operator SDM_OutputWrapper—that is, it converts all internal outputs of SDM that contain information about length, speed, distance, and acceleration defined as real into int, such that all other blocks can stick to their types and also performs the calculation into units used by the environment. This operator forwards input messages and transforms inputs messages into an internal type thereby converting real to int.
Input documents	Subset-026, Chapter 3.13.10: Speed and distance monitoring commands
Safety integrity level	4

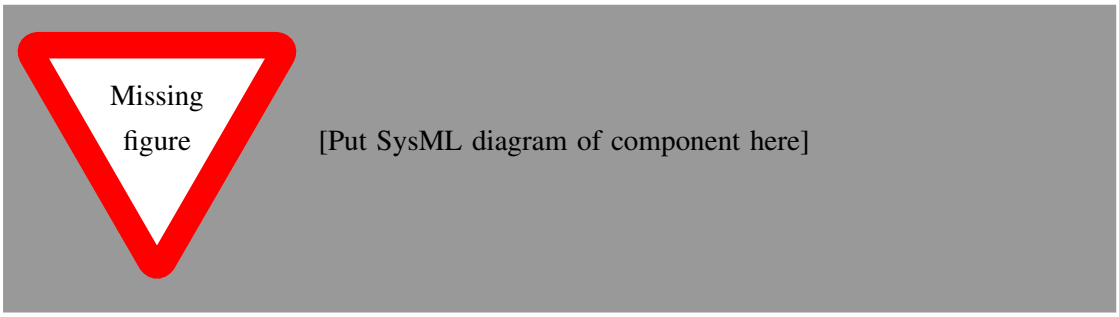


Figure 9. Manage_ETCS_Procedures component SysML diagram

Time constraints	n/a
API requirements	n/a

4.2.3.8.2 Interface

235 For an overview of the interface of this internal component we refer to the SCADE model (c.f. link above) respectively the SCADE generated documentation.

4.3 Manage_ETCS_Procedures

4.3.1 Component Requirements

Component name	Manage_ETCS_Procedures
Link to SCADE model	https://github.com/openETCS/modeling/tree/master/model/Scade/System/ObuFunctions/Procedures
SCADE designer	Baseliyos Jacob, DB Netz AG
Description	This function describes the Start of Mission procedure of the train until the current status will change to another mode, level or other procedure.
Input documents	Subset-026, Chapter 5.4
Safety integrity level	4
Time constraints	n/a
API requirements	n/a

4.3.2 Interface

240 An overview of the interface of component Manage_ETCS_Procedures is shown in Figure 9. The inputs and outputs are described in detail in Section 4.3.2.1 respectively 4.3.2.2. Sub components are described in Section 4.3.3.

4.3.2.1 Inputs

4.3.2.1.1 statusDMI_from_DMI

Input name	statusDMI_from_DMI
Description	input interface of DMI Controller status
Source	manageDMI
Type	DMI_Types_Pkg::DMI_EVC_status_T
Valid range of values	To be completed
Behaviour when value is at boundary	n/a
Behaviour for values out of valid range	function will not be triggered
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	function will not be triggered

4.3.2.1.2 Status_MA_FS_SR_OS_LS_SH_from_MA_L2_Management

Input name	Status_MA_FS_SR_OS_LS_SH_from_MA_L2_Management
Description	Status of MA, Mode and Level from Level and Mode Management
Source	ManageLevelsAndModes
Type	bool
Valid range of values	**true** Movement Authority for Level 2 FS is valid **false** Movement Authority for Level 2 FS is not valid
Behaviour when value is at boundary	n/a
Behaviour for values out of valid range	n/a
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	n/a

4.3.2.1.3 systemtime

Input name	systemtime
Description	Standardized system time used for all internal calculations
Source	Obu_BasicTypes
Type	Obu_BasicTypes_Pkg::T_internal_Type
Valid range of values	[0, maximum positive int value of target platform]
Behaviour when value is at boundary	system time is assumed to be valid
Behaviour for values out of valid range	system time is assumed to be valid
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	system time is assumed to be valid

4.3.2.1.4 StatusModeandLevel_from_Level_and_Mode_Management

Input name	StatusModeandLevel_from_Level_and_Mode_Management
Description	Status of Mode and Level
Source	ManageLevelsAndModes
Type	Level_And_Mode_Types_Pkg::T_Mode_Level
Valid range of values	To be completed
Behaviour when value is at boundary	To be completed
Behaviour for values out of valid range	To be completed
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	To be completed

4.3.2.1.5 mobileSwStatus_p_from_MoRC

Input name	mobileSwStatus_p_from_MoRC

Description	Information about SW status from Management of Radio Communication function
Source	MoRC
Type	MoRC_Pck::mobileSWStatus_Type
Valid range of values	To be completed
Behaviour when value is at boundary	To be completed
Behaviour for values out of valid range	To be completed
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	To be completed

4.3.2.1.6 statusRBCSessioneEstabilished_status_from_MoRC

Input name	statusRBCSessioneEstabilished_status_from_MoRC
Description	Information about RBC Session status from the Management of Radio Communication function
Source	MoRC
Type	Radio_Types_Pkg::sessionStatus_Type
Valid range of values	To be completed
Behaviour when value is at boundary	To be completed
Behaviour for values out of valid range	To be completed
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	To be completed

4.3.2.1.7 cabStatus_from_TIU

Input name	cabStatus_from_TIU
Description	Information about cab desk status from Train Interface Unit function
Source	manageTIU
Type	TIU_Types_Pkg::TIU_trainStatus_T
Valid range of values	To be completed
Behaviour when value is at boundary	To be completed
Behaviour for values out of valid range	To be completed
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	To be completed

4.3.2.1.8 statusValid_Position_from_Position_Calculation

Input name	statusValid_Position_from_Position_Calculation
Description	Information about validity status of the train position calculation
Source	TrainPosition
Type	bool
Valid range of values	**true** Calculated train position is valid **false** Calculated train position is not valid
Behaviour when value is at boundary	n/a
Behaviour for values out of valid range	n/a
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	n/a

4.3.2.1.9 status_DMIlevel_from_DMI

Input name	status_DMIlevel_from_DMI
Description	Information about the status of DMI menu and level request from DMIcController function
Source	manageDMI
Type	DMI_Messages_DMI_to_EVC_Pkg::DMI_Driver_Request_T
Valid range of values	To be completed
Behaviour when value is at boundary	To be completed
Behaviour for values out of valid range	To be completed
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	To be completed

4.3.2.1.10 LevelValid_from_Level_and_Mode_Management

Input name	LevelValid_from_Level_and_Mode_Management
Description	Information about the validty status of the StatusModeandLevel_from_Level_and_Mode_Management input
Source	ManageLevelsAndModes
Type	bool
Valid range of values	**true** Level and Mode information are valid **false** Level and Mode information are not valid
Behaviour when value is at boundary	n/a
Behaviour for values out of valid range	n/a
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	n/a

4.3.2.2 Outputs

255 ##### 4.3.2.2.1 DMI_Entry_Request_to_DMI

Output name	DMI_Entry_Request_to_DMI
Description	Information about input request to the driver
Destination	manageDMI
Type	DMI_Messages_EVC_to_DMI_Pkg::DMI_Entry_Request_T
Valid range of values	To be completed
Behaviour when value is at boundary	To be completed
Behaviour for values out of valid range	To be completed
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	To be completed

4.3.2.2.2 request_Start_Mobile_Terminal_and_RBC_Registration_to_MoRC

Output name	request_Start_Mobile_Terminal_and_RBC_Registration_to_MoRC
Description	This output is a trigger to start the mobile terminal and RBC session registration within the Management of Radio Communication function
Destination	MoRC
Type	Common_Types_Pkg::radioManagementMessage_T
Valid range of values	To be completed
Behaviour when value is at boundary	To be completed

Behaviour for values out of valid range	To be completed
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	To be completed

4.3.2.2.3 powerUp_to_MoRC

Output name	powerUp_to_MoRC
Description	This output is the trigger to activate the Management of Radio Communication function
Destination	MoRC
Type	bool
Valid range of values	**true** MoRC will be activated **false** no action
Behaviour when value is at boundary	n/a
Behaviour for values out of valid range	n/a
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	n/a

4.3.2.2.4 statusstartofmissionongoing_to_MoRC

Output name	statusstartofmissionongoing_to_MoRC
Description	This output gives the information about the start of mission status procedure to the Management of Radio Communication function
Destination	MoRC
Type	bool
Valid range of values	**true** Start of mission procedure is currently ongoing **false** Start of mission procedure is currently not ongoing

Behaviour when value is at boundary	n/a
Behaviour for values out of valid range	n/a
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	n/a

4.3.2.2.5 powerOff_to_MoRC

Output name	powerOff_to_MoRC
Description	This output is the trigger to de-activate the Management of Radio Communication function
Destination	MoRC
Type	bool
Valid range of values	**true** MoRC will be deactivated **false** no action
Behaviour when value is at boundary	n/a
Behaviour for values out of valid range	n/a
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	n/a

4.3.2.2.6 start_ack_to_TIU

Output name	start_ack_to_TIU
Description	This output indicates that the start of mission procedure is completed
Destination	manageTIU
Type	bool
Valid range of values	**true** Start of mission procedure is completed

Behaviour when value is at boundary	n/a
Behaviour for values out of valid range	n/a
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	n/a

4.3.3 Sub Components

4.3.3.1 Awakness_of_Train

4.3.3.1.1 Component Requirements

Component name	Awakness_of_Train
Link to SCADE model	https://github.com/openETCS/modeling/blob/master/model/Scade/System/ObuFunctions/Procedures/ManageProcedure_Pkg.xscade
SCADE designer	Baseliyos Jacob, DB Netz AG
Description	This component describes the Start of Mission procedure of the train until the status of the awakening is completed. From this point on the train will be able to switch to further modes, levels and procedures.
Input documents	Subset-026, Chapter 5, § 5.4
Safety integrity level	4
Time constraints	n/a
API requirements	n/a

4.3.3.1.2 Interface

265 For an overview of the interface of this internal component we refer to the SCADE model (c.f. link above) respectively the SCADE generated documentation.

4.3.3.2 NP

4.3.3.2.1 Component Requirements

Component name	NP
Link to SCADE model	https://github.com/openETCS/modeling/blob/master/model/Scade/System/ObuFunctions/Procedures/ManageProcedure_Pkg.xscade
SCADE designer	Baseliyos Jacob, DB Netz AG

Description	This component implements the No Power status of the train before the driver opens the cab desk.
Input documents	Subset-026, Chapter 5, § 5.4
Safety integrity level	4
Time constraints	n/a
API requirements	n/a

4.3.3.2.2 Interface

270 For an overview of the interface of this internal component we refer to the SCADE model (c.f. link above) respectively the SCADE generated documentation.

4.3.3.3 SoM_L2_3_FS_SR_OS_LS_SH

4.3.3.3.1 Component Requirements

Component name	SoM_L2_3_FS_SR_OS_LS_SH
Link to SCADE model	https://github.com/openETCS/modeling/blob/master/model/Scade/System/ObuFunctions/Procedures/ManageProcedure_Pkg.xscade
SCADE designer	Baseliyos Jacob, DB Netz AG
Description	This component switch to Level 2 or 3 and Mode FS, SR, OS, LS and SH after completion of the awakening of the train.
Input documents	Subset-026, Chapter 5, § 5.4
Safety integrity level	4
Time constraints	n/a
API requirements	n/a

4.3.3.3.2 Interface

275 For an overview of the interface of this internal component we refer to the SCADE model (c.f. link above) respectively the SCADE generated documentation.

4.3.3.4 SoM_NTC_SN

4.3.3.4.1 Component Requirements

Component name	SoM_NTC_SN
Link to SCADE model	https://github.com/openETCS/modeling/blob/master/model/Scade/System/ObuFunctions/Procedures/ManageProcedure_Pkg.xscade

SCADE designer	Baseliyos Jacob, DB Netz AG
Description	This component switch to Level NTC and Mode SN after completion of the awakening of the train.
Input documents	Subset-026, Chapter 5, § 5.4
Safety integrity level	4
Time constraints	n/a
API requirements	n/a

4.3.3.4.2 Interface

280 For an overview of the interface of this internal component we refer to the SCADE model (c.f. link above) respectively the SCADE generated documentation.

4.4 Manage_Track_Data

4.4.1 Component Requirements

Component name	Manage_Track_Data
Link to SCADE model	???
SCADE designer	???
Description	???
Input documents	Subset-026, Chapter ???
Safety integrity level	4
Time constraints	[If applicable description of time constraints, otherwise n/a]
API requirements	[If applicable description of API requirements, otherwise n/a]

4.4.2 Interface

285 An overview of the interface of component Manage_Track_Data is shown in Figure 10. The inputs and outputs are described in detail in Section 4.4.2.1 respectively 4.4.2.2. Sub components are described in Section 4.5.3.

4.4.2.1 Inputs

4.4.2.1.1 [Input 1 name]

Input name	[Name of the input]
Description	[Brief description of the input]
Source	[Name of the source component]

Type	[Type of the input]
Valid range of values	[Complete list of valid values]
Behaviour when value is at boundary	[Description of components behaviour when input value is at boundary]
Behaviour for values out of valid range	[Description of components behaviour when input value is out of valid range]
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

290 #### 4.4.2.1.2 [Input 2 name]

Input name	[Name of the input]
Description	[Brief description of the input]
Source	[Name of the source component]
Type	[Type of the input]
Valid range of values	[Complete list of valid values]
Behaviour when value is at boundary	[Description of components behaviour when input value is at boundary]
Behaviour for values out of valid range	[Description of components behaviour when input value is out of valid range]
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

4.4.2.2 Outputs

4.4.2.2.1 [Output 1 name]

Output name	[Name of the output]
Description	[Brief description of the output]
Destination	[Name of the destination component(s)]
Type	[Type of the output]
Valid range of values	[Complete list of valid values]
Behaviour when value is at boundary	[Description of components behaviour when output value is at boundary]

Behaviour for values out of valid range	[Description of components behaviour when output value is out of valid range]
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

4.4.2.2.2 [Output 2 name]

Output name	[Name of the output]
Description	[Brief description of the output]
Destination	[Name of the destination component(s)]
Type	[Type of the output]
Valid range of values	[Complete list of valid values]
Behaviour when value is at boundary	[Description of components behaviour when output value is at boundary]
Behaviour for values out of valid range	[Description of components behaviour when output value is out of valid range]
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

4.4.3 Sub Components

295 #### 4.4.3.1 Calculate_Train_Position

4.4.3.1.1 Component Requirements

Component name	calculateTrainPosition
Link to SCADE model	https://github.com/openETCS/modeling/tree/master/model/Scade/System/ObuFunctions/ManageLocationRelatedInformation/TrainPosition/CalculateTrainPosition
SCADE designer	Uwe Steinke / Siemens AG

Description	The main purpose of the function is to calculate the locations of linked and unlinked balise groups (BGs) and the current train position while the train is running along the track. In detail, the calculateTrainPosition function provides a couple of essential subfunctions for the onboard unit. These are mainly • creating and maintaining an obu internal coordinate system for all types of location based data • storing all linked and unlinked balise groups resulting from over passing or from announcements (linking information) from the track • calculating and maintaining the locations of all stored balise groups during the train trip, based on odometry and linking information • permanently calculating the current train position based on odometry and passed balise group information • providing the last recently passed linked balise group as the LRBG • providing additional position attribute information • deleting stored balise groups, when appropriate • detecting linking consistency errors • determining, if linking is used on board The calculation algorithms for locations and positions are implemented as specified in https://github.com/openETCS/SRS-Analysis/blob/master/System%20Analysis/WorkingRepository/Group4/SUBSET_26_3-6/DetermineTrainLocationProcedures.pdf
Input documents	Subset-026, Chapter 3.6
Safety integrity level	4
Time constraints	n/a
API requirements	See interface description below

4.4.3.1.2 Interface

An overview of the interface of component calculateTrainPosition is shown in Figure 14. The inputs and outputs are described in detail in Section 4.5.3.2 respectively 4.5.3.3.

300 ### 4.4.3.2 Inputs

4.4.3.2.1 currentOdometry

Input name	currentOdometry

Description	currentOdometry is the actual odometry information as known by the whole EVC model and provided by the models external interface.
Source	External model interface input
Type	Obu_BasicTypes_Pkg::odometry_T
Valid range of values	Obu_BasicTypes_Pkg::odometry_T is a complex data type.
Valid range of values	Obu_BasicTypes_Pkg::odometry_T is a complex data type. Values are given for each element. Format is: Type Name: range/ list of values • bool valid: [true | false]. Must be permanently set to "true". • timestamp: (0 - 2147483647). Current time in ms, must be monotonically increasing. • odo: Obu_BasicTypes_Pkg::OdometryLocations_T: current odometry log values with uncertainties; must behave according to https://github.com/openETCS/SRS-Analysis/blob/master/System%20Analysis/WorkingRepository/Group4/SUBSET_26_3-6/DetermineTrainLocationProcedures.pdf [[3.1]]. Members of OdometryLocations_T are: – o_nominal: L_internal_Type: nominal value in cm. – o_min: L_internal_Type: min. distance = o_min2 - o_min1 – o_max: L_internal_Type: max distance = o_max2 - o_max1 • speed: Obu_BasicTypes_Pkg::OdometrySpeeds_T: not used by calculateTrainPosition • acceleration: Obu_BasicTypes_Pkg::A_internal_Type: not used by calculateTrainPosition • motionState: [noMotion | Motion] • motionDirection: Obu_BasicTypes_Pkg::odoMotionDirection_T [unknownDirection | cabAFirst | cabBFirst] *calculateTrainPosition requires consistent value sets of currentOdometry. calculateTrainPosition itself does not check.*
Behaviour when value is at boundary	n/a
Behaviour for values out of valid range	Enumerated values out of range prohibit code generation. In all other cases, calculateTrainPosition does not have the knowledge for out-of-range checks.

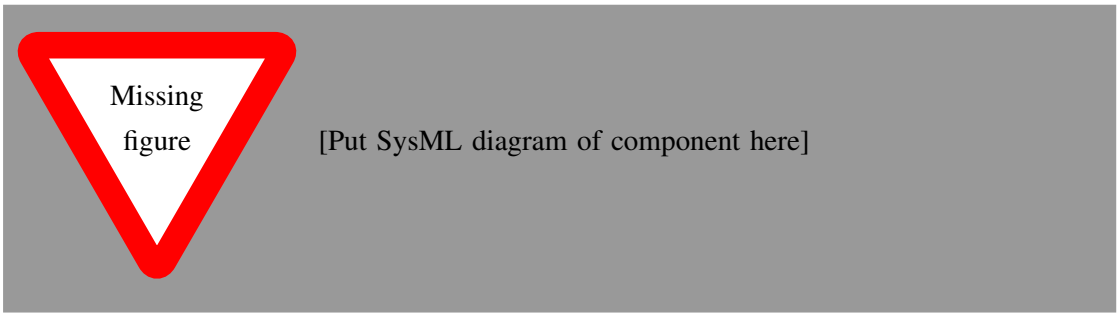


Figure 10. Manage_Track_Data component SysML diagram

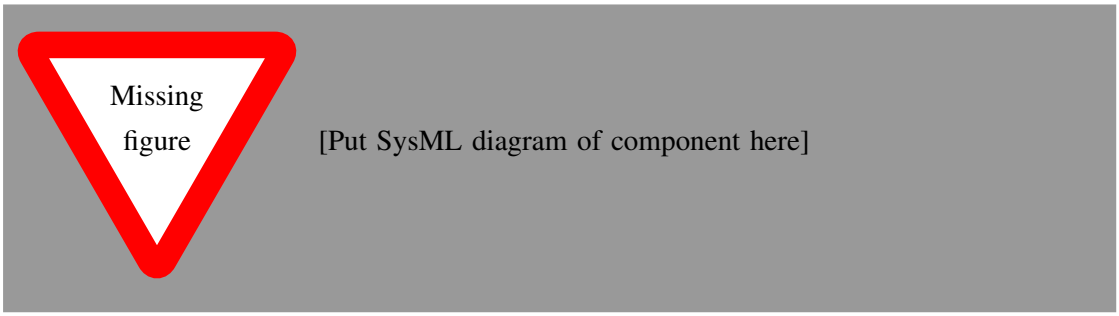


Figure 11. Component SysML diagram

4.4.3.2.2 msgFromTrack

Input name	msgFromTrack
Description	With msgFromTrack calculateTrainPosition receives datagrams from balise groups and RBC.
Source	Manage_TrackSideInformation_Integration_Pkg::Manage_TrackSideInformation_Integra
Type	Common_Types_Pkg::ReceivedMessage_T

Valid range of values	Common_Types_Pkg::ReceivedMessage_T is a complex data type. Values are given for each element. Format is: Type Name: range/ list of values • bool valid: [true | false]. "true" flags a datagram as received and to be evaluated by calculateTrainPosition. Must be set for exactly 1 clock for each received datagram and stay unset otherwise • source: Common_Types_Pkg::MsgSource_T: Designates the source of the datagram: (msrc_undefined | msrc_Euroradio | msrc_Eurobalise | msrc_RadioInfillUnit | msrc_OBU) • radioMetaData: Common_Types_Pkg::radioMetaData_T: not used by calculateTrainPosition • BG_Common_Header: BG_Types_Pkg::BG_Header_T: Header information received from balise groups, refer to Manage_TrackSideInformation_Integration_Pkg::Manage_TrackSideInformation_Integrat • Radio_Common_Header: Radio_Types_Pkg::Radio_TrackTrain_Header_T: Header information received from RBC via radio, refer to Manage_TrackSideInformation_Integration_Pkg::Manage_TrackSideInformation_Integrat • packets: Common_Types_Types_Pkg::CompressedPackets_T: datagram packets, refer to Manage_TrackSideInformation_Integration_Pkg::Manage_TrackSideInformation_Integrat calculatesTrainPosition extracts packet 5 (linking information), if available. • sendingRBC: Common_Types_Types_Pkg::RBC_Id_T: designates the origin RBC and the mobile modem channel used onboard, if received via radio. Refer to Manage_TrackSideInformation_Integration_Pkg::Manage_TrackSideInformation_Integrat for more detailed information. *calculateTrainPosition expects the received information to be consistent and validated before applied to. It does not check, if the information is appropriate due to current EVC mode, level, train or balise orientation. Received balise group or linking information already known by calculateTrainPosition overrides former data.*
Behaviour when value is at boundary	n/a
Behaviour for values out of valid range	Enumerated values out of range prohibit code generation. In all other cases, calculateTrainPosition does not have the knowledge for out-of-range checks.

4.4.3.2.3 trainProperties

Input name	trainProperties
Description	Supplies calculateTrainPosition with train specific properties required for position calculation.
Source	EVC_Support_Pkg::maintainTrainProperties/
Type	TrainPosition_Types_Pck::trainProperties_T
Valid range of values	TrainPosition_Types_Pck::trainProperties_T is a complex data type. Values are given for each element. Format is: Type Name: range/ list of values • nid_engine:: NID_ENGINE as defined by subset 026-7. • nid_operational: NID_OPERATIONAL as defined by subset 026-7. • l_train: L_TRAIN as defined by subset 026-7. • d_baliseAntenna_2_frontend: Obu_BasicTypes_Pkg::LocWithInAcc_T: Distance from the trains balise antenna to the trains front end, in cm with uncertainties. • d_frontend_2_rearend: Obu_BasicTypes_Pkg::LocWithInAcc_T: Distance from the trains Distance from the trains front end to rear end, in cm with uncertainties. • locationAccuracy_DefaultValue: Obu_BasicTypes_Pkg::LocWithInAcc_T: Default location accuracy of balise groups (subset 026, 3.6.4.3.2), in cm with uncertainties. • centerDetectionAcc_DefaultValue: Obu_BasicTypes_Pkg::LocWithInAcc_T: Default accuracy of balise groups detection of the BTM, in cm with uncertainties. Will be applied, if centerDetectionInaccuracy from BTM is not available, especially for announced and not yet passed BGs. *calculateTrainPosition expects this information to be consistent and validated before applied to.*
Behaviour when value is at boundary	n/a
Behaviour for values out of valid range	Enumerated values out of range prohibit code generation. In all other cases, calculateTrainPosition does not have the knowledge for out-of-range checks.

4.4.3.2.4 passedBG

Input name	passedBG
Description	Deprecated alternative input to msgFromTrack. Must not be used any more and is subject to be removed in subsequent releases.

4.4.3.2.5 reset

Input name	reset
Description	Resets and keeps calculateTrainPosition at its initial state and deletes all internally stored data.
Source	To whom it may concern/
Type	bool
Valid range of values	[false | true]
Behaviour when value is at boundary	n/a
Behaviour for values out of valid range	Enumerated values out of range prohibit code generation.

4.4.3.3 Outputs

4.4.3.3.1 trainPosition

Output name	trainPosition
Description	Provides the current train position and LRBG with its attributes. All distance and location computations of the OBU must be based on this information.
Destination	Any drain component which needs the current train position or LRBG
Type	TrainPosition_Types_Pck::trainPosition_T

Valid range of values	TrainPosition_Types_Pck::trainPosition_T is a complex data type. Values are given for each element. Format is: Type Name: range/ list of values • valid: bool: [true | false]. Always true, except for exceptional circumstances. • timestamp: Obu_BasicTypes_Pkg::T_internal_Type: latest time in ms. • trainPositionIsUnknown: bool: true, if the train position is evaluated as "unknonwn" (refer to subset-026, 3.6.3.1.3.1). • noCoordinateSystemHasBeenAssigned: bool: refer to subset 026, 3.4.2, 3.6.3.1.4 • trainPosition: Obu_BasicTypes_Pkg::LocWithInAcc_T: The calculated train position with uncertainties • estimatedFrontEndPosition: Obu_BasicTypes_Pkg::Location_T: Train front end position in cm. • minSafeFrontEndPosition: Obu_BasicTypes_Pkg::Location_T: Train front end position in cm. • maxSafeFrontEndPostion: Obu_BasicTypes_Pkg::Location_T: Train front end position in cm. • LRBG: TrainPosition_Types_Pck::positionedBG_T: the current LRBG. • prvLRBG: TrainPosition_Types_Pck::positionedBG_T: the balise group passed previously to LRBG. For type definition, see below. • nominalOrReverseToLRBG: Q_DLRBG: Orientation of the train in relation to the direction of the LRBG, see subset 026-7. • trainOrientationToLRBG: Q_DIRLRBG: Orientation of the train in relation to the direction of the LRBG, see subset 026-7. • trainRunningDirectionToLRBG: Q_DIRTRAIN: Direction of train movement in relation to the LRBG orientation, see subset 026-7. • linkingIsUsedOnboard: bool: Designates, if at least one announced linked BG is ahead *calculateTrainPosition provides the train position to whom it concerns and recalculates it with every clock cycle*
Behaviour when value is at boundary	n/a
Behaviour for values out of valid range	n/a

Behaviour when value is errorneous, absent or unwanted	n/a

4.4.3.3.2 BGs

Output name	BGs
Description	A list of all linked and unlinked balise groups - known to calculateTrainPosition - in the order they are arranged on the track.
Destination	Any subsequent component which needs the current collection of balises groups
Type	array of TrainPosition_Types_Pck::positionedBG_T
Valid range of values	TrainPosition_Types_Pck::positionedBG_T is a complex data type. Values are given for each array element. Format is: Type Name: range/ list of values • valid: bool: [true | false]. "true" for every existing balise group. • nid_c: NID_C: refer to subset 026-7. • nid_bg: NID_BG: refer to subset 026-7. • q_link: Q_LINK: refer to subset 026-7. • location: Obu_BasicTypes_Pkg::LocWithInAcc_T: The best known location (with inaccuracies) calculated from linking and from passing information. • seqNoOnTrack: int: Sequence number, specifies the order of the BG passed or expected to be passed • infoFromLinking: TrainPosition_Types_Pck::infoFromLinking_T: Describes a linked BG as announced from the linking BG. Mainly, this information is taken from the linking packet. • infoFromPassing: BG_Types_Pkg::passedBG_T: If the balise group has been passed already, this is the relevant information received from the BG. *calculateTrainPosition provides the list of balise groups to whom it concerns.*
Behaviour when value is at boundary	n/a
Behaviour for values out of valid range	n/a

Behaviour when value is errorneous, absent or unwanted	n/a

4.4.3.3.3 errors

Output name	errors
Description	Provides a collection of error flags, raised by calculateTrainPosition.
Destination	Error handlers and components which need to know of common and linking consistency errors.
Type	TrainPosition_Types_Pck::positionErrors_T
Valid range of values	TrainPosition_Types_Pck::positionErrors_T is a complex data type. Values are given for each array element. Format is: Type Name: range/ list of values • outOfMemSpace: bool: Memory overrun: a passed or announced BG could not be stored. • passedBG_foundNotWhereExpected: bool: The currently passed linked BG location does not match its expectation window. • positionCalculation_inconsistent: A consistency problem arose during position calculation. • linkedBGMissed: bool: The expectation window for an announced BG was passed without detecting the BG. • BGpassedInUnexpectedDirection: bool: The BG was passed in a different orientation than announced via linking. • BG_LinkingConsistencyError: bool: Linking consistency error (ref. subset 026, 3.16.2.3). • twoConsecutiveLinkedBGs_missed: bool: 2 consecutive linked balise groups announced by linking are not detected and the end of the expectation window of the second balise group has been passed (subset 026, 3.16.2.7.1). • doubleRepositioningError: bool: Double repositioning error (3.16.2.7.2). • bg: TrainPosition_Types_Pck::positionedBG_T: The corresponding balise group in the case of an error.
Behaviour when value is at boundary	n/a

Behaviour for values out of valid range	n/a
Behaviour when value is errorneous, absent or unwanted	n/a

4.4.3.4 Provide_Position_Report

4.4.3.4.1 Component Requirements

Component name	Provide_Position_Report
Link to SCADE model	???
SCADE designer	Christian Stahl, TWT GmbH
Description	The component builds a position report for the RBC, i.e., message 132, and provides it as an output. There are two triggers for sending message 132: (1) at least one of the triggers of the position report parameters (packet 58) holds or (2) one of the events enabling the sending of the report occurs. As the core position report (i.e., packet 0 or 1) is added to other packets, the component also provides in every clock cycle this core position report. At most one of the two packets is valid. Figure 15 depicts the architecture of the component. In the following, we describe the functionality of each subcomponent: **ReceiveReportParameters** The component reads the position report parameters (i.e., packet 58) from the message bus. When a report is received, the BG information provided is used to update the location of respective BG. This BG is being stored in the list of the last 8 BGs. **PosReport_Supervision** The component supervises trigger (i.e., position report parameter) and events that trigger the sending of a position report. If the output is true, then a report has to be sent. **ErrorManager** The component takes all nine possible error messages as an input and aggregates them to a vector. **Build_Packets0_1** The component builds packets 0 and 1; at most one of them is valid. **Build_PosReport** The component builds nine position report messages—there can be up to nine errors, and for each error an individual report has to be sent. **AddBGToFIFO** The component adds the current reported BG to the list of BGs for which a report has been sent. Adding of this BG is performed according to the FIFO method.

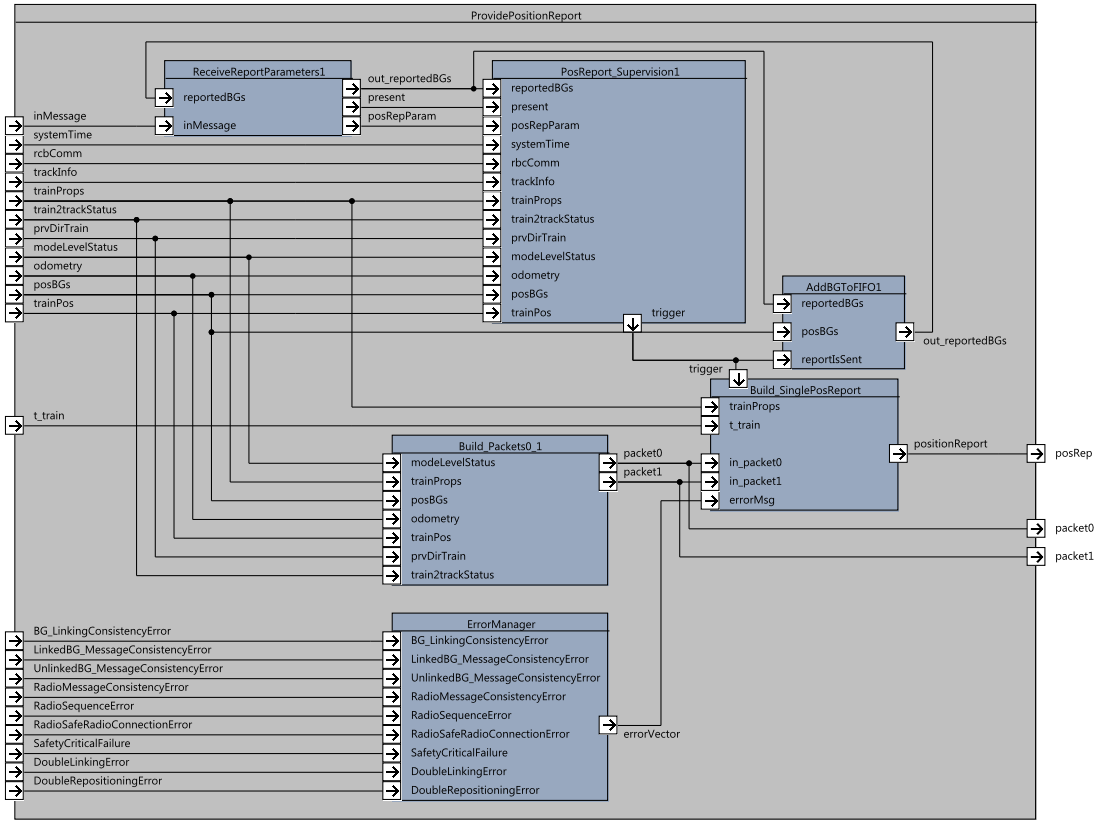


Figure 12. Provide Position Report component SysML diagram

Input documents	Subset-026, Chapter 3.6.5
Safety integrity level	4
Time constraints	[If applicable description of time constraints, otherwise n/a]
API requirements	[If applicable description of API requirements, otherwise n/a]

4.4.3.4.2 Interface

Figure 15 shows the architecture of component Provide Position Report. Most of the inputs and outputs have been explained in detail in Section 4.4.2.1 and Section 4.4.2.2

4.5 trainData

4.5.1 Component Requirements

Component name	trainData
Link to SCADE model	https://github.com/openETCS/modeling/blob/master/model/Scade/System/ObuFunctions/manageData/trainData/trainData.etp
SCADE designer	Bernd Hekele (DB)

Description	Implementation of the train data with the corresponding interfaces to track, driver and RBC. Data first is received from the train (TIU). Second step: train data is sent to the driver (DMI). The part relevant for driver interface is confirmed by the driver and sent back to the evc. Data received via this interface is merged with the data received via TIU. Message Flow: Message 129 (Validated Train Data) Message 8 (Acknowledment of Train Data) is processed as apart of the validation procedure with the RBC. Message 146 (Acknolwedement) Messages from RBC: Message 8 (Acknowledment) is processed as apart of the validation procedure with the RBC.
Input documents	Subset-026, Chapter 3.18.3
Safety integrity level	4
Time constraints	n/a
API requirements	Train Data needs systemtime for stamping messages, access to input from the track messages and access to the output of RBC messages

4.5.2 Interface

An overview of the interface of component trainData is shown in Figure 13. The inputs and outputs are described in detail in Section 4.5.2.1 respectively 4.5.2.2. Sub components are described in Section **??**.

4.5.2.1 Inputs

4.5.2.1.1 reset

Input name	reset
Description	triggers the reset of the train data and the train data status data
Source	persistant data status management

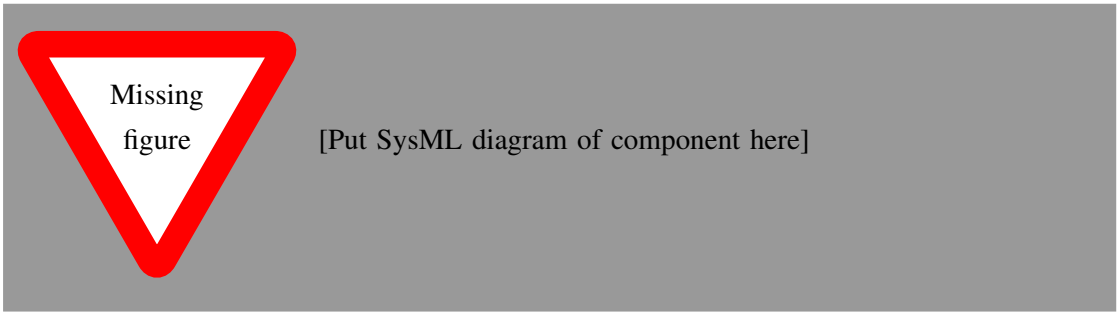


Figure 13. Manage_Track_Data component SysML diagram

Type	bool
Valid range of values	**true** ??? **false** ???
Behaviour when value is at boundary	n/a
Behaviour for values out of valid range	n/a
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	n/a

4.5.2.1.2 trainDatafromTIU

Input name	trainDatafromTIU
Description	train data received via TIU. Change of data is indicated with the valid flag. This data is expected to be received in the first place. In the current implementation it is not supported to change data after a mission has been started.
Source	Train Interface Unit (TIU)
Type	TIU_Types_Pkg::trainData_T
Valid range of values	Input with valid information is indicated with the valid flag.
Behaviour when value is at boundary	[Description of components behaviour when input value is at boundary]
Behaviour for values out of valid range	[Description of components behaviour when input value is out of valid range]
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

4.5.2.2 Outputs

4.5.2.2.1 [Output 1 name]

Output name	[Name of the output]
Description	[Brief description of the output]
Destination	[Name of the destination component(s)]

Type	[Type of the output]
Valid range of values	[Complete list of valid values]
Behaviour when value is at boundary	[Description of components behaviour when output value is at boundary]
Behaviour for values out of valid range	[Description of components behaviour when output value is out of valid range]
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

4.5.2.2.2 [Output 2 name]

Output name	[Name of the output]
Description	[Brief description of the output]
Destination	[Name of the destination component(s)]
Type	[Type of the output]
Valid range of values	[Complete list of valid values]
Behaviour when value is at boundary	[Description of components behaviour when output value is at boundary]
Behaviour for values out of valid range	[Description of components behaviour when output value is out of valid range]
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

4.5.3 Sub Components

4.5.3.1 Calculate_Train_Position

4.5.3.1.1 Component Requirements

Component name	calculateTrainPosition
Link to SCADE model	https://github.com/openETCS/modeling/tree/master/model/Scade/System/ObuFunctions/ManageLocationRelatedInformation/TrainPosition/CalculateTrainPosition
SCADE designer	Uwe Steinke / Siemens AG

Description	The main purpose of the function is to calculate the locations of linked and unlinked balise groups (BGs) and the current train position while the train is running along the track. In detail, the calculateTrainPosition function provides a couple of essential subfunctions for the onboard unit. These are mainly • creating and maintaining an obu internal coordinate system for all types of location based data • storing all linked and unlinked balise groups resulting from over passing or from announcements (linking information) from the track • calculating and maintaining the locations of all stored balise groups during the train trip, based on odometry and linking information • permanently calculating the current train position based on odometry and passed balise group information • providing the last recently passed linked balise group as the LRBG • providing additional position attribute information • deleting stored balise groups, when appropriate • detecting linking consistency errors • determining, if linking is used on board The calculation algorithms for locations and positions are implemented as specified in https://github.com/openETCS/SRS-Analysis/blob/master/System%20Analysis/WorkingRepository/Group4/SUBSET_26_3-6/DetermineTrainLocationProcedures.pdf
Input documents	Subset-026, Chapter 3.6
Safety integrity level	4
Time constraints	n/a
API requirements	See interface description below

4.5.3.1.2 Interface

An overview of the interface of component calculateTrainPosition is shown in Figure 14. The inputs and outputs are described in detail in Section 4.5.3.2 respectively 4.5.3.3.

4.5.3.2 Inputs

4.5.3.2.1 currentOdometry

Input name	currentOdometry

Description	currentOdometry is the actual odometry information as known by the whole EVC model and provided by the models external interface.
Source	External model interface input
Type	Obu_BasicTypes_Pkg::odometry_T
Valid range of values	Obu_BasicTypes_Pkg::odometry_T is a complex data type.
Valid range of values	Obu_BasicTypes_Pkg::odometry_T is a complex data type. Values are given for each element. Format is: Type Name: range/ list of values • bool valid: [true | false]. Must be permanently set to "true". • timestamp: (0 - 2147483647). Current time in ms, must be monotonically increasing. • odo: Obu_BasicTypes_Pkg::OdometryLocations_T: current odometry log values with uncertainties; must behave according to https://github.com/openETCS/SRS-Analysis/blob/master/System%20Analysis/WorkingRepository/Group4/SUBSET_26_3-6/DetermineTrainLocationProcedures.pdf [[3.1]]. Members of OdometryLocations_T are: – o_nominal: L_internal_Type: nominal value in cm. – o_min: L_internal_Type: min. distance = o_min2 - o_min1 – o_max: L_internal_Type: max distance = o_max2 - o_max1 • speed: Obu_BasicTypes_Pkg::OdometrySpeeds_T: not used by calculateTrainPosition • acceleration: Obu_BasicTypes_Pkg::A_internal_Type: not used by calculateTrainPosition • motionState: [noMotion | Motion] • motionDirection: Obu_BasicTypes_Pkg::odoMotionDirection_T [unknownDirection | cabAFirst | cabBFirst] *calculateTrainPosition requires consistent value sets of currentOdometry. calculateTrainPosition itself does not check.*
Behaviour when value is at boundary	n/a
Behaviour for values out of valid range	Enumerated values out of range prohibit code generation. In all other cases, calculateTrainPosition does not have the knowledge for out-of-range checks.

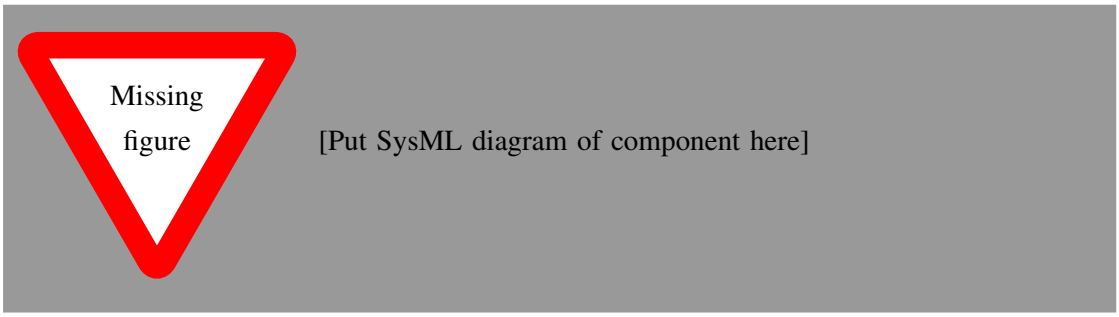


Figure 14. Component SysML diagram

4.5.3.2.2 msgFromTrack

Input name	msgFromTrack
Description	With msgFromTrack calculateTrainPosition receives datagrams from balise groups and RBC.
Source	Manage_TrackSideInformation_Integration_Pkg::Manage_TrackSideInformation_Integra
Type	Common_Types_Pkg::ReceivedMessage_T

Valid range of values	Common_Types_Pkg::ReceivedMessage_T is a complex data type. Values are given for each element. Format is: Type Name: range/ list of values • bool valid: [true | false]. "true" flags a datagram as received and to be evaluated by calculateTrainPosition. Must be set for exactly 1 clock for each received datagram and stay unset otherwise • source: Common_Types_Pkg::MsgSource_T: Designates the source of the datagram: (msrc_undefined | msrc_Euroradio | msrc_Eurobalise | msrc_RadioInfillUnit | msrc_OBU) • radioMetaData: Common_Types_Pkg::radioMetaData_T: not used by calculateTrainPosition • BG_Common_Header: BG_Types_Pkg::BG_Header_T: Header information received from balise groups, refer to Manage_TrackSideInformation_Integration_Pkg::Manage_TrackSideInformation_Integrat • Radio_Common_Header: Radio_Types_Pkg::Radio_TrackTrain_Header_T: Header information received from RBC via radio, refer to Manage_TrackSideInformation_Integration_Pkg::Manage_TrackSideInformation_Integrat • packets: Common_Types_Types_Pkg::CompressedPackets_T: datagram packets, refer to Manage_TrackSideInformation_Integration_Pkg::Manage_TrackSideInformation_Integrat calculatesTrainPosition extracts packet 5 (linking information), if available. • sendingRBC: Common_Types_Types_Pkg::RBC_Id_T: designates the origin RBC and the mobile modem channel used onboard, if received via radio. Refer to Manage_TrackSideInformation_Integration_Pkg::Manage_TrackSideInformation_Integrat for more detailed information. *calculateTrainPosition expects the received information to be consistent and validated before applied to. It does not check, if the information is appropriate due to current EVC mode, level, train or balise orientation. Received balise group or linking information already known by calculateTrainPosition overrides former data.*
Behaviour when value is at boundary	n/a
Behaviour for values out of valid range	Enumerated values out of range prohibit code generation. In all other cases, calculateTrainPosition does not have the knowledge for out-of-range checks.

4.5.3.2.3 trainProperties

Input name	trainProperties
Description	Supplies calculateTrainPosition with train specific properties required for position calculation.
Source	EVC_Support_Pkg::maintainTrainProperties/
Type	TrainPosition_Types_Pck::trainProperties_T
Valid range of values	TrainPosition_Types_Pck::trainProperties_T is a complex data type. Values are given for each element. Format is: Type Name: range/ list of values • nid_engine:: NID_ENGINE as defined by subset 026-7. • nid_operational: NID_OPERATIONAL as defined by subset 026-7. • l_train: L_TRAIN as defined by subset 026-7. • d_baliseAntenna_2_frontend: Obu_BasicTypes_Pkg::LocWithInAcc_T: Distance from the trains balise antenna to the trains front end, in cm with uncertainties. • d_frontend_2_rearend: Obu_BasicTypes_Pkg::LocWithInAcc_T: Distance from the trains Distance from the trains front end to rear end, in cm with uncertainties. • locationAccuracy_DefaultValue: Obu_BasicTypes_Pkg::LocWithInAcc_T: Default location accuracy of balise groups (subset 026, 3.6.4.3.2), in cm with uncertainties. • centerDetectionAcc_DefaultValue: Obu_BasicTypes_Pkg::LocWithInAcc_T: Default accuracy of balise groups detection of the BTM, in cm with uncertainties. Will be applied, if centerDetectionInaccuracy from BTM is not available, especially for announced and not yet passed BGs. *calculateTrainPosition expects this information to be consistent and validated before applied to.*
Behaviour when value is at boundary	n/a
Behaviour for values out of valid range	Enumerated values out of range prohibit code generation. In all other cases, calculateTrainPosition does not have the knowledge for out-of-range checks.

4.5.3.2.4 passedBG

Input name	passedBG
Description	Deprecated alternative input to msgFromTrack. Must not be used any more and is subject to be removed in subsequent releases.

4.5.3.2.5 reset

Input name	reset
Description	Resets and keeps calculateTrainPosition at its initial state and deletes all internally stored data.
Source	To whom it may concern/
Type	bool
Valid range of values	[false | true]
Behaviour when value is at boundary	n/a
Behaviour for values out of valid range	Enumerated values out of range prohibit code generation.

4.5.3.3 Outputs

4.5.3.3.1 trainPosition

Output name	trainPosition
Description	Provides the current train position and LRBG with its attributes. All distance and location computations of the OBU must be based on this information.
Destination	Any drain component which needs the current train position or LRBG
Type	TrainPosition_Types_Pck::trainPosition_T

Valid range of values	TrainPosition_Types_Pck::trainPosition_T is a complex data type. Values are given for each element. Format is: Type Name: range/ list of values • valid: bool: [true | false]. Always true, except for exceptional circumstances. • timestamp: Obu_BasicTypes_Pkg::T_internal_Type: latest time in ms. • trainPositionIsUnknown: bool: true, if the train position is evaluated as "unknonwn" (refer to subset-026, 3.6.3.1.3.1). • noCoordinateSystemHasBeenAssigned: bool: refer to subset 026, 3.4.2, 3.6.3.1.4 • trainPosition: Obu_BasicTypes_Pkg::LocWithInAcc_T: The calculated train position with uncertainties • estimatedFrontEndPosition: Obu_BasicTypes_Pkg::Location_T: Train front end position in cm. • minSafeFrontEndPosition: Obu_BasicTypes_Pkg::Location_T: Train front end position in cm. • maxSafeFrontEndPostion: Obu_BasicTypes_Pkg::Location_T: Train front end position in cm. • LRBG: TrainPosition_Types_Pck::positionedBG_T: the current LRBG. • prvLRBG: TrainPosition_Types_Pck::positionedBG_T: the balise group passed previously to LRBG. For type definition, see below. • nominalOrReverseToLRBG: Q_DLRBG: Orientation of the train in relation to the direction of the LRBG, see subset 026-7. • trainOrientationToLRBG: Q_DIRLRBG: Orientation of the train in relation to the direction of the LRBG, see subset 026-7. • trainRunningDirectionToLRBG: Q_DIRTRAIN: Direction of train movement in relation to the LRBG orientation, see subset 026-7. • linkingIsUsedOnboard: bool: Designates, if at least one announced linked BG is ahead *calculateTrainPosition provides the train position to whom it concerns and recalculates it with every clock cycle*
Behaviour when value is at boundary	n/a
Behaviour for values out of valid range	n/a

Behaviour when value is errorneous, absent or unwanted	n/a

4.5.3.3.2 BGs

Output name	BGs
Description	A list of all linked and unlinked balise groups - known to calculateTrainPosition - in the order they are arranged on the track.
Destination	Any subsequent component which needs the current collection of balises groups
Type	array of TrainPosition_Types_Pck::positionedBG_T
Valid range of values	TrainPosition_Types_Pck::positionedBG_T is a complex data type. Values are given for each array element. Format is: Type Name: range/ list of values • valid: bool: [true | false]. "true" for every existing balise group. • nid_c: NID_C: refer to subset 026-7. • nid_bg: NID_BG: refer to subset 026-7. • q_link: Q_LINK: refer to subset 026-7. • location: Obu_BasicTypes_Pkg::LocWithInAcc_T: The best known location (with inaccuracies) calculated from linking and from passing information. • seqNoOnTrack: int: Sequence number, specifies the order of the BG passed or expected to be passed • infoFromLinking: TrainPosition_Types_Pck::infoFromLinking_T: Describes a linked BG as announced from the linking BG. Mainly, this information is taken from the linking packet. • infoFromPassing: BG_Types_Pkg::passedBG_T: If the balise group has been passed already, this is the relevant information received from the BG. *calculateTrainPosition provides the list of balise groups to whom it concerns.*
Behaviour when value is at boundary	n/a
Behaviour for values out of valid range	n/a

Behaviour when value is errorneous, absent or unwanted	n/a

4.5.3.3.3 errors

Output name	errors
Description	Provides a collection of error flags, raised by calculateTrainPosition.
Destination	Error handlers and components which need to know of common and linking consistency errors.
Type	TrainPosition_Types_Pck::positionErrors_T
Valid range of values	TrainPosition_Types_Pck::positionErrors_T is a complex data type. Values are given for each array element. Format is: Type Name: range/ list of values • outOfMemSpace: bool: Memory overrun: a passed or announced BG could not be stored. • passedBG_foundNotWhereExpected: bool: The currently passed linked BG location does not match its expectation window. • positionCalculation_inconsistent: A consistency problem arose during position calculation. • linkedBGMissed: bool: The expectation window for an announced BG was passed without detecting the BG. • BGpassedInUnexpectedDirection: bool: The BG was passed in a different orientation than announced via linking. • BG_LinkingConsistencyError: bool: Linking consistency error (ref. subset 026, 3.16.2.3). • twoConsecutiveLinkedBGs_missed: bool: 2 consecutive linked balise groups announced by linking are not detected and the end of the expectation window of the second balise group has been passed (subset 026, 3.16.2.7.1). • doubleRepositioningError: bool: Double repositioning error (3.16.2.7.2). • bg: TrainPosition_Types_Pck::positionedBG_T: The corresponding balise group in the case of an error.
Behaviour when value is at boundary	n/a

Behaviour for values out of valid range	n/a
Behaviour when value is errorneous, absent or unwanted	n/a

4.5.3.4 Provide_Position_Report

4.5.3.4.1 Component Requirements

Component name	Provide_Position_Report
Link to SCADE model	???
SCADE designer	Christian Stahl, TWT GmbH
Description	The component builds a position report for the RBC, i.e., message 132, and provides it as an output. There are two triggers for sending message 132: (1) at least one of the triggers of the position report parameters (packet 58) holds or (2) one of the events enabling the sending of the report occurs. As the core position report (i.e., packet 0 or 1) is added to other packets, the component also provides in every clock cycle this core position report. At most one of the two packets is valid. Figure 15 depicts the architecture of the component. In the following, we describe the functionality of each subcomponent: **ReceiveReportParameters** The component reads the position report parameters (i.e., packet 58) from the message bus. When a report is received, the BG information provided is used to update the location of respective BG. This BG is being stored in the list of the last 8 BGs. **PosReport_Supervision** The component supervises trigger (i.e., position report parameter) and events that trigger the sending of a position report. If the output is true, then a report has to be sent. **ErrorManager** The component takes all nine possible error messages as an input and aggregates them to a vector. **Build_Packets0_1** The component builds packets 0 and 1; at most one of them is valid. **Build_PosReport** The component builds nine position report messages—there can be up to nine errors, and for each error an individual report has to be sent. **AddBGToFIFO** The component adds the current reported BG to the list of BGs for which a report has been sent. Adding of this BG is performed according to the FIFO method.

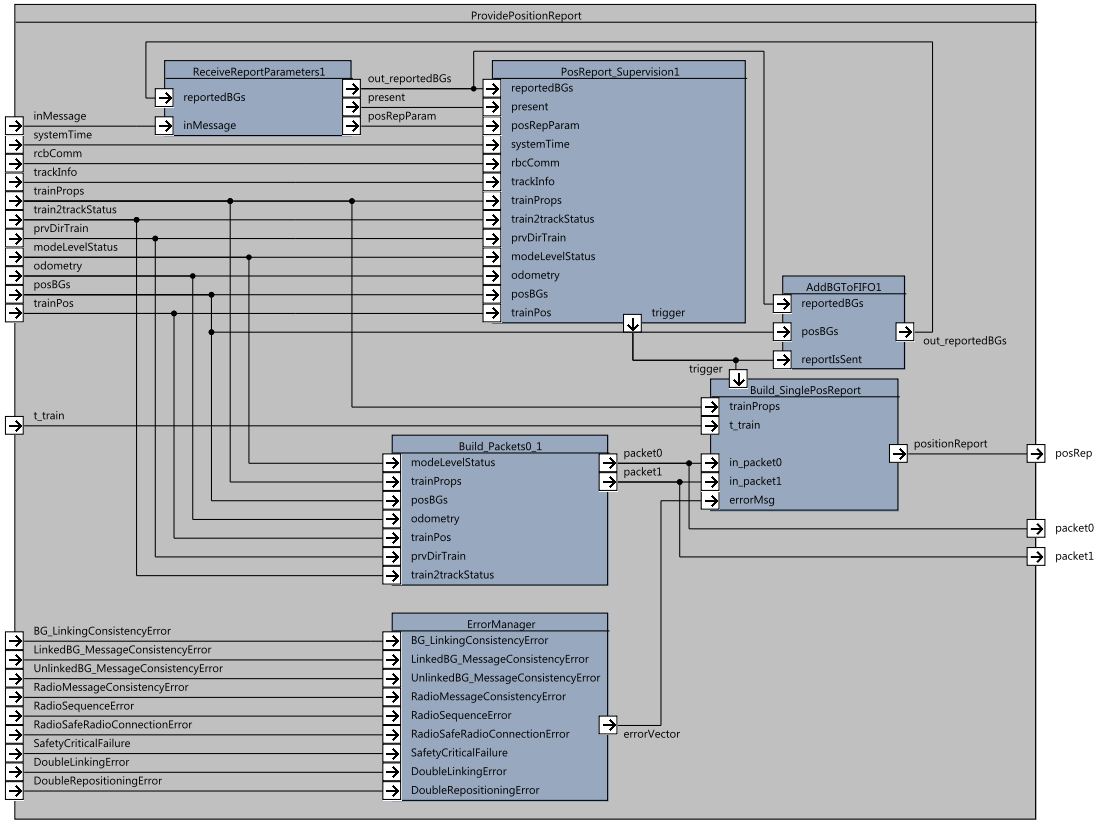


Figure 15. Provide Position Report component SysML diagram

Input documents	Subset-026, Chapter 3.6.5
Safety integrity level	4
Time constraints	[If applicable description of time constraints, otherwise n/a]
API requirements	[If applicable description of API requirements, otherwise n/a]

4.5.3.4.2 Interface

Figure 15 shows the architecture of component Provide Position Report. Most of the inputs and outputs have been explained in detail in Section 4.4.2.1 and Section 4.4.2.2

4.6 Mode_and_Level

4.6.1 Component Requirements

Component name	Mode_and_Level
Link to SCADE model	???
SCADE designer	???
Description	???

Input documents	Subset-026, Chapter 4 Subset-026, Chapter 5
Safety integrity level	4
Time constraints	[If applicable description of time constraints, otherwise n/a]
API requirements	[If applicable description of API requirements, otherwise n/a]

4.6.2 Interface

An overview of the interface of component Mode_and_Level is shown in Figure 16. The inputs and outputs are described in detail in Section 4.6.2.1 respectively 4.6.2.2. Sub components are described in Section 4.6.3.

4.6.2.1 Inputs

4.6.2.1.1 [Input 1 name]

Input name	[Name of the input]
Description	[Brief description of the input]
Source	[Name of the source component]
Type	[Type of the input]
Valid range of values	[Complete list of valid values]
Behaviour when value is at boundary	[Description of components behaviour when input value is at boundary]
Behaviour for values out of valid range	[Description of components behaviour when input value is out of valid range]
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

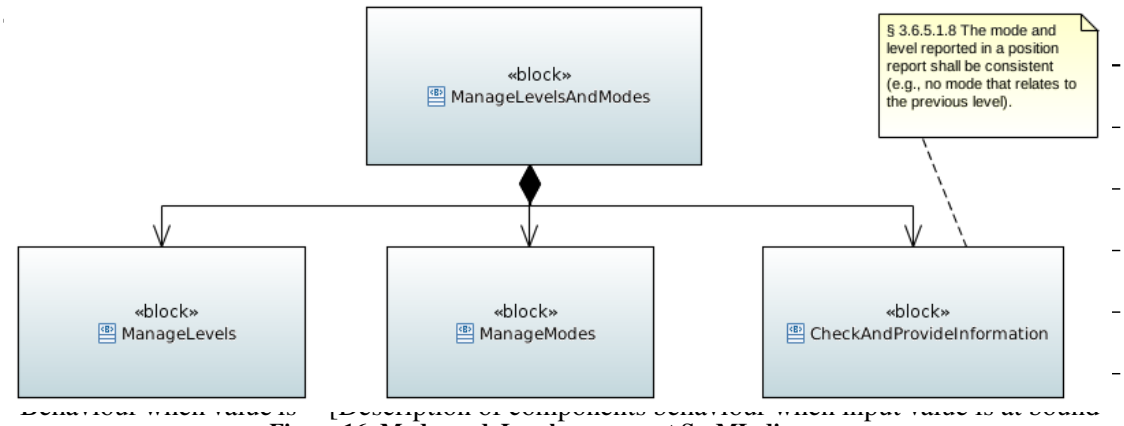


Figure 16: Mode_and_Level component SysML diagram

Behaviour for values out of valid range	[Description of components behaviour when input value is out of valid range]
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

4.6.2.2 Outputs

4.6.2.2.1 [Output 1 name]

Output name	[Name of the output]
Description	[Brief description of the output]
Destination	[Name of the destination component(s)]
Type	[Type of the output]
Valid range of values	[Complete list of valid values]
Behaviour when value is at boundary	[Description of components behaviour when output value is at boundary]
Behaviour for values out of valid range	[Description of components behaviour when output value is out of valid range]
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

4.6.2.2.2 [Output 2 name]

Output name	[Name of the output]
Description	[Brief description of the output]
Destination	[Name of the destination component(s)]
Type	[Type of the output]
Valid range of values	[Complete list of valid values]
Behaviour when value is at boundary	[Description of components behaviour when output value is at boundary]
Behaviour for values out of valid range	[Description of components behaviour when output value is out of valid range]
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

4.6.3 Sub Components

4.6.3.1 Level_Management

4.6.3.1.1 Component Requirements

Component name	Level_Management
Link to SCADE model	https://github.com/openETCS/modeling/tree/master/openETCSArchitectureAndDesign/WorkGroups/Group3/SCADE/LevelManagement/
SCADE designer	Marielle Petit-Doche, Systerel
Description	The level management subsystem receives level transition order tables and selects the order with the highest probability. It stores the information about the selected transition order and transits to the requested level once the train passes the location of the level transition. If required, the driver is asked to acknowledge the transition, in case of no acknowledge or if conditions for the level transition are not fulfilled, the train gets tripped. On the most abstract level the design consists of the *manage_priorities* function which takes the level transition order priority tables as inputs and computes the highest priority transition. This transition order is the fed to the *computeLevelTransitions* operator. This operator consists of three main parts. The *ComputeTransitionConditions* operator that emits the fulfilled conditions to change from a given level to a new level, the *LevelStateMachine* that stores the current level and takes the computed change conditions as input for possible level transitions and finally the *driverAck* operator which contains a state machine that stores the information whether the system is currently waiting for a driver acknowledge and emits the train trip information if necessary.
Input documents	Subset-026, Chapter 5.10
Safety integrity level	4
Time constraints	[If applicable description of time constraints, otherwise n/a]
API requirements	[If applicable description of API requirements, otherwise n/a]

4.6.3.1.2 Interface

For an overview of the interface of this internal component we refer to the SCADE model (c.f. link above) respectively the SCADE generated documentation.

4.6.3.2 Mode_Management

4.6.3.2.1 Component Requirements

Component name	Mode_Management
Link to SCADE model	https://github.com/openETCS/modeling/tree/master/model/Scade/System/ObuFunctions/ManageLevelsAndModes/Modes
SCADE designer	Marielle Petit-Doche, Systerel
Description	This function is in charge of the computation of new mode to apply according to conditions from inputs (track information, driver interactions, train data,...) and other functions. Three subfunctions are defined: **Inputs** proceeds to inputs check and preparation. **ComputeModesCondition** performs all specific procedure linked to mode management and defined in [1] sections 5.4, 5.5, 5.6, 5.7, 5.8, 5.9, 5.11, 5.12, 5.13, 5.19 and specifies the conditions to define a mode transition according condition table of section 4.6.3 of [1] **SwitchModes** performs the mode selection according the conditions and priorities defined in transition table section 4.6.2 of [1] **Outputs** prepares packet of outputs.
Input documents	Subset-026, Chapter 4.4, 4.6, 5.4, 5.5, 5.6, 5.7, 5.8, 5.9, 5.11, 5.12, 5.13, 5.19
Safety integrity level	4
Time constraints	[If applicable description of time constraints, otherwise n/a]
API requirements	[If applicable description of API requirements, otherwise n/a]

4.6.3.2.2 Interface

For an overview of the interface of this internal component we refer to the SCADE model (c.f. link above) respectively the SCADE generated documentation.

4.6.3.3 Check_and_Provide_Mode_and_Level

4.6.3.3.1 Component Requirements

Component name	Check_and_Provide_Mode_and_Level
Link to SCADE model	https://github.com/openETCS/modeling/tree/master/model/Scade/System/ObuFunctions/ManageLevelsAndModes/Modes
SCADE designer	Marielle Petit-Doche, Systerel

Description	Checks compatibility between mode and level and provides outputs.
Input documents	Subset-026, Chapter 3.6.5
Safety integrity level	4
Time constraints	[If applicable description of time constraints, otherwise n/a]
API requirements	[If applicable description of API requirements, otherwise n/a]

4.6.3.3.2 Interface

For an overview of the interface of this internal component we refer to the SCADE model (c.f. link above) respectively the SCADE generated documentation.

4.7 Manage_Radio_Communication

4.7.1 Component Requirements

Component name	Mode_and_Level
Link to SCADE model	???
SCADE designer	Uwe Steinke, Siemens AG
Description	???
Input documents	Subset-026, Chapter 4 Subset-026, Chapter 5
Safety integrity level	4
Time constraints	[If applicable description of time constraints, otherwise n/a]
API requirements	[If applicable description of API requirements, otherwise n/a]

4.7.2 Interface

An overview of the interface of component Manage_Radio_Communication is shown in Figure 17. The inputs and outputs are described in detail in Section 4.7.2.1 respectively 4.7.2.2. Sub components are described in Section 4.7.3.

4.7.2.1 Inputs

4.7.2.1.1 [Input 1 name]

Input name	[Name of the input]
Description	[Brief description of the input]
Source	[Name of the source component]
Type	[Type of the input]

Valid range of values	[Complete list of valid values]
Behaviour when value is at boundary	[Description of components behaviour when input value is at boundary]
Behaviour for values out of valid range	[Description of components behaviour when input value is out of valid range]
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

4.7.2.1.2 [Input 2 name]

Input name	[Name of the input]
Description	[Brief description of the input]
Source	[Name of the source component]
Type	[Type of the input]
Valid range of values	[Complete list of valid values]
Behaviour when value is at boundary	[Description of components behaviour when input value is at boundary]
Behaviour for values out of valid range	[Description of components behaviour when input value is out of valid range]
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

4.7.2.2 Outputs

4.7.2.2.1 [Output 1 name]

Output name	[Name of the output]
Description	[Brief description of the output]
Destination	[Name of the destination component(s)]
Type	[Type of the output]
Valid range of values	[Complete list of valid values]
Behaviour when value is at boundary	[Description of components behaviour when output value is at boundary]

Behaviour for values out of valid range	[Description of components behaviour when output value is out of valid range]
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

4.7.2.2.2 [Output 2 name]

Output name	[Name of the output]
Description	[Brief description of the output]
Destination	[Name of the destination component(s)]
Type	[Type of the output]
Valid range of values	[Complete list of valid values]
Behaviour when value is at boundary	[Description of components behaviour when output value is at boundary]
Behaviour for values out of valid range	[Description of components behaviour when output value is out of valid range]
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

4.7.3 Sub Components

4.7.3.1 Management_of_Radio_Communication

390 #### 4.7.3.1.1 Component Requirements

Component name	Management_of_Radio_Communication
Link to SCADE model	https://github.com/openETCS/modeling/tree/master/model/Scade/System/ObuFunctions/Radio/MoRC
SCADE designer	Marielle Petit-Doche, Systerel

Description	The management of radio communication *MoRC* implements the onboard management part of a single communication session with the track, i.e. a single RBC. It controls the establishing, maintaining and termination process of a radio communication session and steers the underlying communication safety layer and the mobile device. Those and the data transfer itself are not part of the function. The kernel function of the *MoRC* component is *managementOfRadioCommunication* (figure ???). The implementation is kept close to the prose of Subset-026, chap. 3.5. Since chap. 3.5 rarely refers to terms, variable types, packets and messages of the ETCS language as specified in Subset-026, chap. 7 and 8, *managementOfRadioCommunication* does neither. To be capable of being integrated with other OBU software components, *MoRC* had to be wrapped with a transformer between the ETCS and the "chap. 3.5" language. This is the purpose of the main function of *MoRC*, *MoRC_Main*. The function *managementOfRadioCommunication* implements the session states establishing, maintaining and termination as described in Subset-026, chap. 3.5. A SCADE state machine reflects this state model (figure ???) accurately. Within each of the states, the activities needed as long as the state is active, are performed. When there is no communication session (state *NoSession*) currently, the state machine waits for events that initiate a session (subfunction *initiate_a_Session*). When the appropriate conditions are fulfilled, the state machine moves to the *Establishing* state. Here in, it runs through the sequence required fore establishing a session (subfunction *establish_a_Session*. Dependent on the results, the state machine changes over to the *Maintaining* or *Terminating* state. While in *Maintaining*, the communication connection is monitored. When an event triggering the session termination occurs, the state machine switches to the state *Terminating* with the subfunction *terminating_a_CommunicationSession* and performs the session termination sequence. In parallel to the main state machine, *managementOfRadioCommunication* monitors all the time whether the session has to be terminated (subfunction *initiateTerminatingASession*) or if the session has the be terminated and subsequently established (subfunction *terminateAndEstablishSession*). *registeringToTheRadioNetwork* is responsible for connection to the radio network. *safeRadioConnectionIndication* controls the radio connection indication for the driver.
Input documents	Subset-026, Chapter 3.5
Safety integrity level	4
Time constraints	[If applicable description of time constraints, otherwise n/a]
API requirements	[If applicable description of API requirements, otherwise n/a]

4.7.3.1.2 Interface

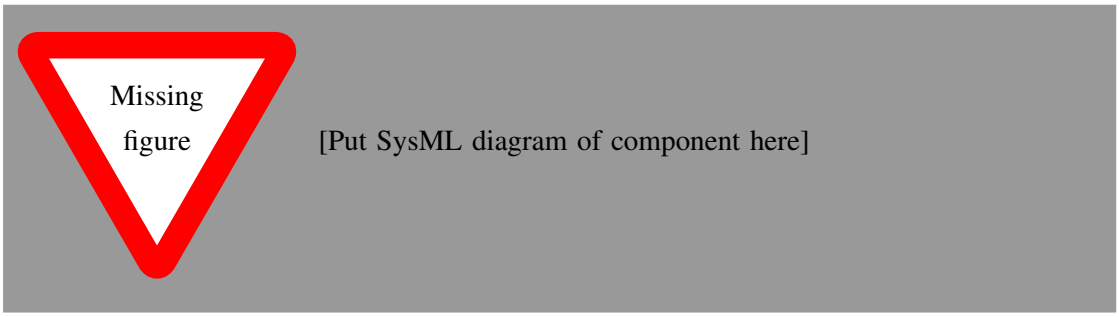


Figure 17. Manage_Radio_Communication component SysML diagram

For an overview of the interface of this internal component we refer to the SCADE model (c.f. link above) respectively the SCADE generated documentation.

5 F3: Measure Train Movement

6 F4: Manage Radio Communication

7 F5: Manage JRU

8 F6: DMI Controller

8.1 DMI

8.1.1 Component Requirements

Component name	DMI
Link to SCADE model	https://github.com/openETCS/modeling/tree/master/model/Scade/System/DMI_Control
SCADE designer	Valerio D'Angelo, DB Netz AG
Description	The DMI controller interacts with the DMI display and is responsible for alls procedures between the DMI display and Driver. Furthermore, the DMI controller will interact with the DMI Management to compute the received information (e.g. driver number request, ...) and send, if necessary, data or reports to the DMI Management (acknowledge, text messages...). The DMI Controller is a passive module, this means that all the processing are performed EVC-side, therefore the DMI Controller simply responds to the requests of the EVC or Driver and performs some checks according with the information received from EVC.
Input documents	ERA_ERTMS_015560
Safety integrity level	4
Time constraints	[If applicable description of time constraints, otherwise n/a]
API requirements	[If applicable description of API requirements, otherwise n/a]

8.1.2 Interface

An overview of the interface of component DMI is shown in Figure 18. The inputs and outputs are described in detail in Section 8.1.2.1 respectively 8.1.2.2.

8.1.2.1 Inputs

8.1.2.1.1 DMI_entry_request

Input name	DMI_entry_request
Description	Request to input data (e.g. driver id, Train running number etc.)
Source	DMI Management
Type	[Type of the input]

Valid range of values	[Complete list of valid values]
Behaviour when value is at boundary	[Description of components behaviour when input value is at boundary]
Behaviour for values out of valid range	[Description of components behaviour when input value is out of valid range]
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

405 #### 8.1.2.1.2 DMI_identifier_request

Input name	DMI_identifier_request
Description	Request of the DMI informations
Source	DMI Management
Type	[Type of the input]
Valid range of values	[Complete list of valid values]
Behaviour when value is at boundary	[Description of components behaviour when input value is at boundary]
Behaviour for values out of valid range	[Description of components behaviour when input value is out of valid range]
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

8.1.2.1.3 DMI_menu_request

Input name	DMI_menu_request
Description	Request to enable or disable buttons
Source	DMI Management
Type	[Type of the input]
Valid range of values	[Complete list of valid values]
Behaviour when value is at boundary	[Description of components behaviour when input value is at boundary]
Behaviour for values out of valid range	[Description of components behaviour when input value is out of valid range]

Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

8.1.2.1.4 DMI_dynamic

Input name	DMI_dynamic
Description	Contains informations about current speed, current mode etc.
Source	DMI Management
Type	[Type of the input]
Valid range of values	[Complete list of valid values]
Behaviour when value is at boundary	[Description of components behaviour when input value is at boundary]
Behaviour for values out of valid range	[Description of components behaviour when input value is out of valid range]
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

8.1.2.1.5 DMI_text_message

Input name	DMI_text_message
Description	Contains predefined or plain text messages
Source	DMI Management
Type	[Type of the input]
Valid range of values	[Complete list of valid values]
Behaviour when value is at boundary	[Description of components behaviour when input value is at boundary]
Behaviour for values out of valid range	[Description of components behaviour when input value is out of valid range]
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

8.1.2.1.6 DMI_icons

Input name	DMI_icons

Description	Request to display one or more icons in any area
Source	DMI Management
Type	[Type of the input]
Valid range of values	[Complete list of valid values]
Behaviour when value is at boundary	[Description of components behaviour when input value is at boundary]
Behaviour for values out of valid range	[Description of components behaviour when input value is out of valid range]
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

410 #### 8.1.2.1.7 DMI_driver_identifier

Input name	DMI_driver_identifier
Description	Contains the default or entered driver identifier
Source	DMI Management
Type	[Type of the input]
Valid range of values	[Complete list of valid values]
Behaviour when value is at boundary	[Description of components behaviour when input value is at boundary]
Behaviour for values out of valid range	[Description of components behaviour when input value is out of valid range]
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

8.1.2.1.8 DMI_train_running_number

Input name	DMI_train_running_number
Description	Contains the default or entered train running number
Source	DMI Management
Type	[Type of the input]
Valid range of values	[Complete list of valid values]

Behaviour when value is at boundary	[Description of components behaviour when input value is at boundary]
Behaviour for values out of valid range	[Description of components behaviour when input value is out of valid range]
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

8.1.2.1.9 DMI_train_data

Input name	DMI_train_data
Description	Contains the default or entered train data
Source	DMI Management
Type	[Type of the input]
Valid range of values	[Complete list of valid values]
Behaviour when value is at boundary	[Description of components behaviour when input value is at boundary]
Behaviour for values out of valid range	[Description of components behaviour when input value is out of valid range]
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

8.1.2.1.10 TIU_trainStatus

Input name	TIU_trainStatus
Description	Open/close Desk signal
Source	TIU
Type	[Type of the input]
Valid range of values	[Complete list of valid values]
Behaviour when value is at boundary	[Description of components behaviour when input value is at boundary]
Behaviour for values out of valid range	[Description of components behaviour when input value is out of valid range]
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

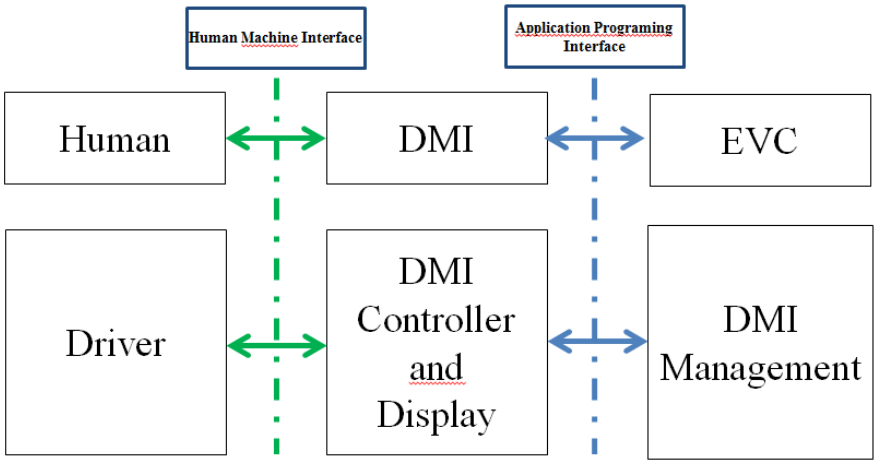


Figure 18. DMI component SysML diagram

8.1.2.2 Outputs

415 ##### 8.1.2.2.1 DMI_identifier

Output name	DMI_identifier
Description	Information about DMI (e.g. version, cabin identifier etc.)
Destination	DMI Management
Type	[Type of the output]
Valid range of values	[Complete list of valid values]
Behaviour when value is at boundary	[Description of components behaviour when output value is at boundary]
Behaviour for values out of valid range	[Description of components behaviour when output value is out of valid range]
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

8.1.2.2.2 DMI_driver_request

Output name	DMI_driver_request
Description	Driver request or acknowledgement
Destination	DMI Management
Type	[Type of the output]
Valid range of values	[Complete list of valid values]
Behaviour when value is at boundary	[Description of components behaviour when output value is at boundary]
Behaviour for values out of valid range	[Description of components behaviour when output value is out of valid range]
Behaviour when value is	[Description of components behaviour when value is erroneous,

Type	[Type of the output]
Valid range of values	[Complete list of valid values]
Behaviour when value is at boundary	[Description of components behaviour when output value is at boundary]
Behaviour for values out of valid range	[Description of components behaviour when output value is out of valid range]
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

8.1.2.2.4 DMI_status_report

Output name	DMI_status_report
Description	The actual status of DMI (keep alive)
Destination	DMI Management
Type	[Type of the output]
Valid range of values	[Complete list of valid values]
Behaviour when value is at boundary	[Description of components behaviour when output value is at boundary]
Behaviour for values out of valid range	[Description of components behaviour when output value is out of valid range]
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

8.1.2.2.5 DMI_text_message_ack

Output name	DMI_text_message_ack
Description	Text message acknowledgement
Destination	DMI Management
Type	[Type of the output]
Valid range of values	[Complete list of valid values]
Behaviour when value is at boundary	[Description of components behaviour when output value is at boundary]
Behaviour for values out of valid range	[Description of components behaviour when output value is out of valid range]

Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

8.1.2.2.6 DMI_icons_ack

Output name	DMI_icons_ack
Description	Icon acknowledgement
Destination	DMI Management
Type	[Type of the output]
Valid range of values	[Complete list of valid values]
Behaviour when value is at boundary	[Description of components behaviour when output value is at boundary]
Behaviour for values out of valid range	[Description of components behaviour when output value is out of valid range]
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

8.1.2.2.7 DMI_driver_identifier

Output name	DMI_driver_identifier
Description	Contains the default or entered driver identifier
Destination	DMI Management
Type	[Type of the output]
Valid range of values	[Complete list of valid values]
Behaviour when value is at boundary	[Description of components behaviour when output value is at boundary]
Behaviour for values out of valid range	[Description of components behaviour when output value is out of valid range]
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

8.1.2.2.8 DMI_train_running_number

Output name	DMI_train_running_number
Description	Contains the default or entered train running number
Destination	DMI Management
Type	[Type of the output]
Valid range of values	[Complete list of valid values]
Behaviour when value is at boundary	[Description of components behaviour when output value is at boundary]
Behaviour for values out of valid range	[Description of components behaviour when output value is out of valid range]
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

8.1.2.2.9 DMI_train_data

Output name	DMI_train_data
Description	Contains the default or entered train data
Destination	DMI Management
Type	[Type of the output]
Valid range of values	[Complete list of valid values]
Behaviour when value is at boundary	[Description of components behaviour when output value is at boundary]
Behaviour for values out of valid range	[Description of components behaviour when output value is out of valid range]
Behaviour when value is erroneous, absent or unwanted (i.e. spurious)	[Description of components behaviour when value is erroneous, absent or unwanted (i.e. spurious)]

References

[1] ERA. *System Requirements Specification, SUBSET-026*, v3.3.0 edition, March 2012.