OETCS/WP3/D3.5.2

openETCS

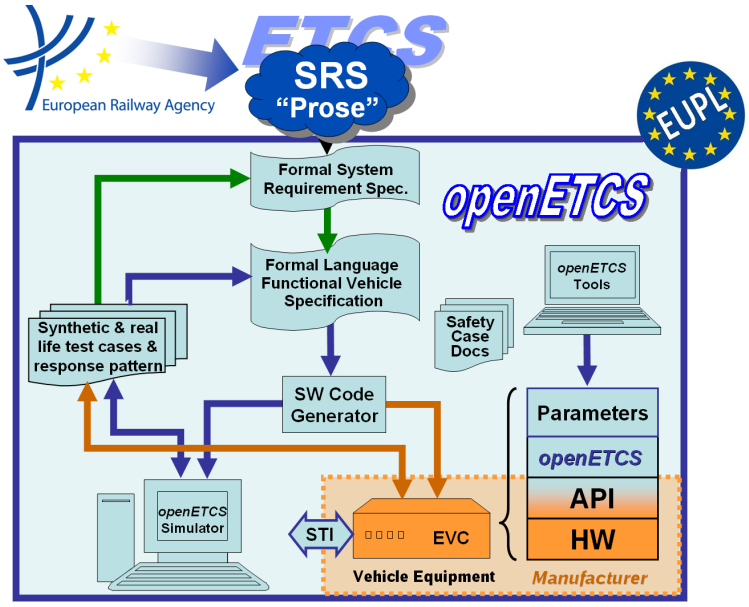
ITEA2 Project
Call 6 11025
2012 – 2015

Work Package 3: “Modeling”

openETCS Design Specification

Baseliyos Jacob, Peter Mahlmann

May 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/D3.5.2
May 2015

openETCS Design Specification

Document approbation

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

Baseliyos Jacob, Peter Mahlmann
DB Netz AG

Architecture and Design Specification

Prepared for openETCS@ITEA2 Project

Abstract: This document gives an introduction to the software and component design of the openETCS OBU model. The functional scope is tailored to cover the functionality required for the openETCS demonstration as an objective of the ITEA2 project. The goal is to develop a formal model and to demonstrate the functionality during a proof of concept on the ETCS Level 2 Utrecht Amsterdam track with real scenarios. It has to be read as a complement to the models in SysML and Scade languages.

Modification History

Version	Section	Modification / Description	Author	Date
0.1	Document	Initial document providing structure	Peter Mahlmann	27.05.2015

Table of Contents

Figures and Tables

Figures

Tables

1 Functional Breakdown

1.1 openETCS API Runtime System and Input to the EVC)

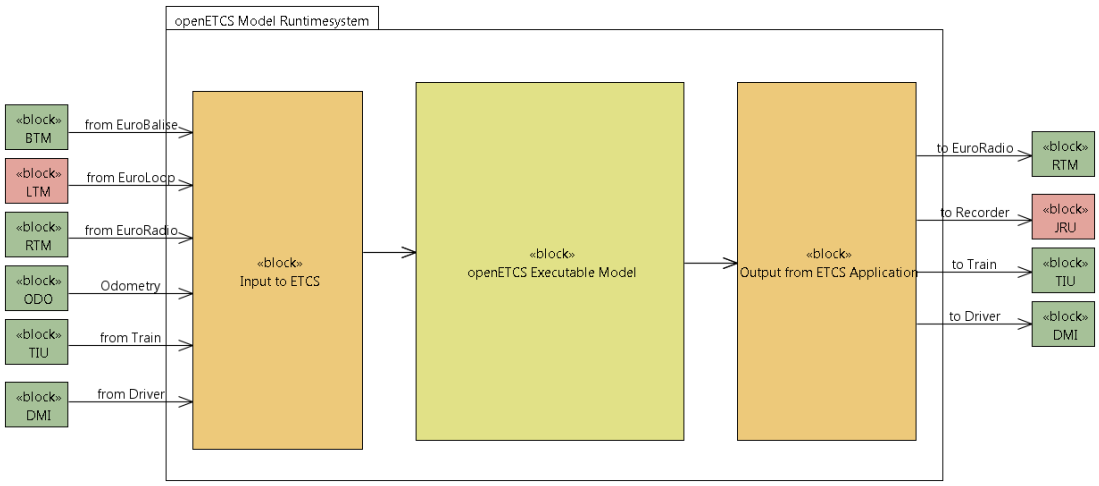


Figure 1. openETCS API Highlevel View

Figure 1 shows the structure of API with respect of the software architecture. Input boxes and output boxes not implemented in this stage are marked as red, other interfaces are marked as green. The System covers functions for processing Inputs from other Units, functions for processing Outputs to other functions and a basic runtime system. Inputs are used to feed the input to the executable model before calling it, outputs are used for collecting information provided by the executable model to be passed to the relevant interfaces after the execution cycle has finished.

1.1.1 Principles for Interfaces (openETCS API)

Information is exchanged *messages* in an asynchronous way. A message is a set of information corresponding to an event of a particular unit, e.g. a balise received from the BTM. The possible kind of messages are described in chapter **??**.

The information is passed to the executable model as parmeters to the snychronous call of a procedure (Interface to the executable model). Since the availability of input messages to the application is not guaranteed the parts of the interfaces are defined with a "present" flag. In addition, fields of input arraysquite often is of variable size. Implementation in the concrete interface in this use-case is the use of a "size" parameter and a "valid"-flag.

1.1.2 openETCS Model Runtime System

The openETCS model runtime system also provides:

Input Functions From other Units In this entity messages from other connected units are received.

Output Functions to other Units The entity writes messages to other connected units.

Conversation Functions for Messages (Bitwalker) The conversion function are triggered by Input and Ouput Functions. The main task is to convert input messages from an bit-packed format into logical ETCS messages (the ETCS language) and Output messages from Logical into a bit-packed format. The logical format of the messages is defined for all used types in the openETCS data dictonary.

Variable size elements in the Messages are converted to fixed length arrays with an used elements indicator. Optional elements are indicated with an valid flag.

The conversion routines are responsible for checking the data received is valid. If faults are detected the information is passed to the openETCS executable model for further reaction.

Model Cycle The version management function is part of the message handling. This implies, conversions from other physical or logical layouts of messages are mapped onto a generic format used in the EVC. Information about the origin version of the message is part of the messages.

The executable model is called in cycles. In the cycle

- First the received input messages are decoded
- The input data is passed to the executable model in a predefined order. **(Details for the interface to be defined)**.
- Output is encoded according to the SRS and passed to the buffers to the units.

1.1.3 Input Interfaces of the openETCS API From other Units of the OBU

Interfaces are defined in the Scade project APITypes (package API_Msg_Pkg.xscade).

In the interfaces the following principles for indicating the quality of the information is used:

Indicator	Type	Purpose
present	bool	True indicates the component has been changed compared to the previous call of the routine
valid	bool	True indicates the component is valid to be used.

In the next table we can see the interfaces being used in the openETCS system. Details on the interfaces are defined further down.

Unit	Name	Processing Function
BTM	Balise Telegram	Receive Messages
DMI	Driver Machine Interface	DMI Manager
EURORADIO	Communication Management	Communication Management
EURORADIO	Radio Messages	Receive Messages
ODO	Odometer	All Parts
System TIME	Time system of the OBU	All Parts
TIU	Train Data	All Parts

Information in the following sections gives an more detailed overview of the structure of the interfaces.

1.1.4 Message based interface (BTM, RTM)

Balise Message (Track to Train)

Message Name	Optional Packets	Restrictions in the current scope
Balise Telegram	3: National Values 41: Level Transition Order 42: Session Management 45: Radio Network registration 46: Conditional Level Transition Order 65: Temporary Speed Restriction 66: Revoke Temporary Speed Restriction 72: Packet for sending plain text messages 137: Stop if in Staff Responsible 255: End of Information	Used in Scenario
Balise Telegram	0, 2, 3, 5, 6, 12, 16, 21, 27, 39, 40, 41, 42, 44, 45, 46, 49, 51, 52, 65, 66, 67, 68, 69, 70, 71, 72, 76, 79, 80, 88, 90, 131, 132, 133, 134, 135, 136, 137, 138, 139, 141, 145, 180, 181, 254	Not Used in Scenario

Radio Messages (Track to Train)

Message Name	Optional Packets	Restrictions in the current scope
2: SR Authorisation	63: List of Balises in SR Authority	Message Not Supported
3: Movement Authority	21: Gradient Profile 27: International Static Speed Profile 49: List of balises for SH Area 80: Mode profile plus common optional packets	a
9: Request To Shorten MA	49: List of balises for SH Area 80: Mode profile	
24: General Message	From RBC: 21: Gradient Profile 27: International Static Speed Profile plus common optional packets From RIU: 44, 45, 143, 180, 254	Messages from RIU are not supported

28: SH authorised	3, 44, 49	
33: MA with Shifted Location Reference	21: Gradient Profile 27: International Static Speed Profile 49: List of balises for SH Area 80: Mode profile plus common optional packets	
37: Infill MA	5, 21, 27, 39, 40, 41, 44, 49, 51, 52, 65, 66, 68, 69, 70, 71, 80, 88, 138, 139	Message Not Supported
List of common optional parameters	3, 5, 39, 40, 51, 41, 42, 44, 45, 52, 57, 58, 64, 65, 66, 68, 69, 70, 71, 72, 76, 79, 88, 131, 138, 139, 140, 180	

The runtime system is in charge to transfer the messages from its stream mode first to compressed message format.

1.1.5 Interfaces to the Time System

The interface types are defined in the OBU_Basic_Types_Pkg Package. The system time is defined in the basic software.

The system TIME is provided to the executable model at the begin of the cycle. It is not refreshed during the cycle. The time provided to the application is equal to 0 at power-up of the EVC (it is not a “UTC time” nor a “Local Time”), then must increase at each cycle (unit = 1 msec), until it reaches its maximum value (i.e current EVC limitation = 24 hours)

- TIME (T_internal_Type, 32-bit INT)
  Standardized system time type used for all internal time calculations: in ms. The time is defined as a cyclic counter: When the maximum is exceeded the time starts from 0 again.
- CLOCK (to be implemented)
  The clocking system is provided by the JRU. A GPS based clock is assumed to provide the local time.

1.1.6 Interfaces to the Odometry System

The interface types are defined in the OBU_Basic_Types_Pkg Package. The odometer gives the current information of the positing system of the train. In this section the structure of the interfaces are only highlighted. Details, including the internal definitions for distances, locations speed and time are implemented in the package.

- Odometer (odometry_T)
  - valid (bool)
    valid flag, i.e., the information is provided by the ODO system and can be used.
  - timestamp (T_internal_Type)
    of the system when the odometer information was collected. Please, see also general remarks on the time system.

- Coordinate (odometryLocation_T)
  - nominal (L_internal_Type) [cm]
  - min (L_internal_Type) [cm]
  - max (L_internal_Type) [cm]

  The type used for length values is a 32 bit integer. Min and max value give the interval where the train is to be expected. The bounderies are determined by the inaccuracy of the positioning system. All values are set to 0 when the train starts.
- speed (OdometrySpeeds_T) [km/h]
  - v_safeNominal (speed internal type) [km/h]
    The safe nominal estimation of the speed which will be bounded between 98% and 100% of the upper estimation
  - v_rawNominal (speed internal type) [km/h]
    The raw nominal estimation of the speed which will be bounded between the lower and the upper estimations
  - v_lower (speed internal type) [km/h]
    The lower estimation of the speed
  - v_upper (speed internal type) [km/h]
    The upper estimation of the speed

  The type used for speed values is a 32 bit integer. Min and max value give the interval where the train is to be expected. The bounderies are determined by the inaccuracy of the positioning system. All values are set to 0 when the train starts.
- acceleration (A_internal_Type)[0.01 m/s2],
  Standardized acceleration type for all internal calculations : in
- motionState (Enumeration)
  indicates whether the train is in motion or in no motion
- motionDirection (Enumeration)
  indicates the direction of the train, i.e., CAB-A first, CAB-B first or unknown.

1.1.7 Interfaces to the Train Interfaces (TIU)

The following infomration is based on the implementation of the Alstom API. The interface is organised in packets. The packets of the Alstom implementation are listed in the appendix to this document.

The description of interfaces needed for the current scope will be added according to the use.

1.1.8 Output Interfaces of the openETCS API TO other Units of the OBU

From Function	Name	To Unit	Description
	Radio Output Message	EURORADIO	
	Communication Management	EURORADIO	
	Driver Information	DMI	
	Train Data	TIU	

Packets: to be completed

Radio Messages to be completed

2 Design Description

2.1 F1: Receive information from Trackside

2.2 F2: ETCS Kernel

2.2.1 Manage_TrackSideInformation_Integration

The block “Manage_TrackSideInformation_Integration” is responsible for receiving Eurobalise telegrams and Euroradio messages from the API and perform several consistency checks on the input.

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.

2.2.1.1 Inputs

For providing the output, the module needs different input data flows. An overview is provided in table 1

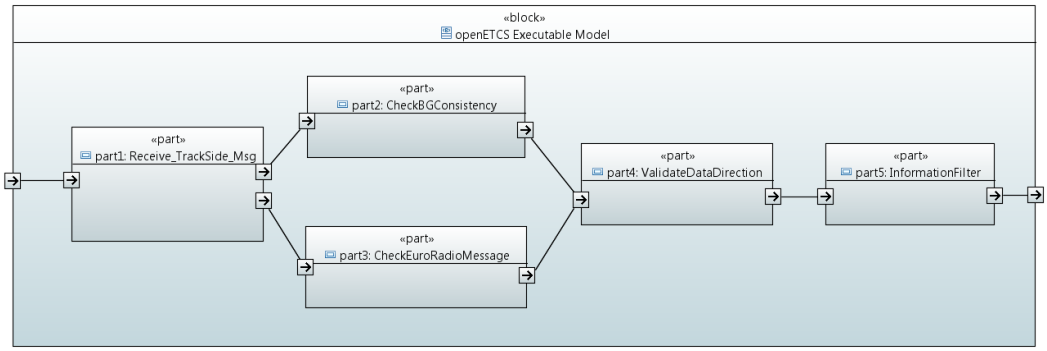


Figure 2. Structure of the Manage_TrackSideInformation_Integration module with submodules.

Index	Input name	Input type	Source
0	fullChecks	bool	Configuration
1	API_trackSide_Message	API_Msg_Pkg::API_TrackSideInput_T	API
2	ActualOdometry	Obu_BasicTypes_Pkg::odometry_T	Odometer
3	reset	bool	Environment
4	trainPosition	TrainPosition_Types_Pck::trainPosition_T	Calculate Train Position
5	modeAndLevel	BG_Types_Pkg::ModeAndLevelStatus_T	Mode and Level
6	tNvContact	Obu_BasicTypes_Pkg::T_internal_Type	Database
7	lastRelevantEventTimestamp	Obu_BasicTypes_Pkg::T_internal_Type	Database
8	connectionStatus	Radio_Types_Pkg::sessionStatus_Type	Manage Radio Communication
9	inSupervisingRbcId	int	Database
10	inAnnouncedBGs	TrainPosition_Types_Pck::positionedBGs_T	Calculate Train Position
11	q_nvlocacc	Q_NVLOCACC	Database

Table 1. Overview over input

Input 0: fullChecks

The boolean indicates, if all checks on the message should be performed. The possible values are given in table 2.

Value	Interpretation
true	All checks are performed.
false	The module Information Filter is deactivated.

Table 2. Possible values for the input fullChecks

Input 1: API_trackSide_Message

The API_trackSide_Message is the message received from the API. The API performs preprocessing of RTM and BTM messages and deliveres a maximum of a single message per cycle to the SCADE model.

Input 2: ActualOdometry

The input ActualOdometry is provided by the external odometry module of the train. It contains location information with inaccuracies.

Input 3: reset

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.

Value	Interpretation
true	All data kept in the module is deleted and no input is accepted.
false	No action. Data at input is accepted.

Table 3. Possible values for the input reset.

Input 4: trainPosition

The input trainPosition is generated by the “Calculate Train Position” module and contains the current position of the train.

Input 5: modeAndLevel

The input is generated by the “Mode and level management” module. It provides the current level and mode of the EVC.

Input 6: tNvContact

For monitoring the safe radio connection, the national value T_NVCONTACT is needed as an input.

Input 7: lastRelevantEventTimestamp

For monitoring the safe radio connection, it’s necessary, that the time between two packets is less than the value of T_NVCONTACT.

In situations like level-changes or announced radioholes, not the timestamp of the last message is relevant for comparison, but the timestamp of the last relevant event. This can be e.g. the timestamp of the level change or the timestamp of 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.

Input 8: connectionStatus

The input connectionStatus will give information about the radio connection. This input is delivered by the session management module, not from the API. The information is needed to perform the timing check, which is depending on the connection state.

Value	Interpretation
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 RBC is established.

Table 4. Possible values for the input connectionStatus.

Input 9: inSupervisingRbcId

For the submodule “Information Filter”, the information is needed, which radio messages are sent by the supervising RBC. To recognize these messages, the identifier of the supervising RBC is needed.

Input 10: inAnnouncedBGs

This input 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.

Input 11: q_nvlocacc

The national value determines the location accuracy and is delivered by the database.

2.2.1.2 Outputs

The output of the module provides the received and processed Euroradio and Eurobalise messages. The module combines messages both from Eurobalises and from Euroradio to one common dataflow.

An overview over the output dataflows is provided in table 5.

Index	Output name	Output type
0	outputMessage	Common_Types_Pkg::ReceivedMessage_T
1	ApplyServiceBrake	bool
2	BadBAliseMessageToDMI	bool
3	errorLinkedBG	bool
4	errorUnlinkedBG	bool
5	passedBG	BG_Types_Pkg::passedBG_T
6	outPositionParams	Common_Types_Pkg::PositionReportParameter_T
7	outRadioManagement	Common_Types_Pkg::radioManagementMessage_T
8	radioSequenceError	bool
9	radioMessageConsistencyError	bool

Table 5. Dataflow at output

Output 0: outputMessage

The element outputMessage consists of the type ReceivedMessage_T combines both balise and radio messages to one common datatype. This datatype contains all variables and packets, which are possible for the given scenario.

Name	Datatype	Description
valid	bool	true, if no consistency errors were detected.
source	Common_Types_Pkg::MsgSource_T	Defines, if this is a Euroradio or Eurobalise message.
packetMetadata	Common_Types_Pkg::Metadata_T	contains the metadata of the packets
radioMetadata	Common_Types_Pkg::RadioMetadata_T	contains the metadata of the radio specific header variables
BG_Common_Header	BG_Types_Pkg::BG_Header_T	Header of Eurobalise message
Radio_Common_Header	Radio_Types_Pkg::Radio_TrackTrain_Header_T	Header of Euroradio message
packets	Common_Types_Pkg::Packets_T	Structure of packets in messages

Table 6. Structure of ReceivedMessage_T

The Eurobalise-common-header BG_Header_T consists of the fields visible in the SCADE-declaration. The structure corresponds to the structure defined in the SRS chapter 8.4.2.1. Some fields were removed since they are not needed anymore for further processing after building messages from separate telegrams.

The Euroradio-common-header Radio_TrackTrain_Header_T consists of the fields visible in the SCADE declaration. The structure corresponds to the structure defined in the SRS chapter 8.4.4.6.1. The structure contains all variables required by possible NID_MESSAGE values for the given scenario. Which values are valid is defined in the field radioMetadata.

Output 1: ApplyServiceBreak

The flag indicates the balise group the train just passed could not be processed correctly. The check results in the request for a service break.

Output 2: BadBaliseMessageToDMI

Information to be passed to the DMI to indicate the reception of a “bad balise” to the driver.

Output 3: errorLinkedBG

Value	Interpretation
true	A error in a linked balise group was detected.
false	No error in a linked balise group was detected.

Table 7. Possible values for the input errorLinkedBG

Output 4: errorUnlinkedBG

Value	Interpretation
true	A error in an unlinked balise group was detected.
false	No error in an unlinked balise group was detected.

Table 8. Possible values for the input errorUnlinkedBG

Output 5: passedBG

The output passedBG provides the received balise group message in a special format needed by the module “Calculate train position”.

Output 6: outPositionParams

The output outPositionParams provides the parameters for the position report in a special format needed by the module “Provide Position Report”.

Output 7: outRadioManagement

The output outRadioManagement provides the messages for radio session management in a special format needed by the module “Management of Radio Communication”.

Output 8: radioSequenceError

Value	Interpretation
true	A sequence error or a timeout has been detected in the radio message.
false	No error in the radio message sequence was detected.

Table 9. Possible values for the input radioSequenceError

Output 9: radioMessageConsistencyError

Value	Interpretation
true	A consistency error has been detected in the radio message.
false	No consistency error in the radio message was detected.

Table 10. Possible values for the input radioMessageConsistencyError

2.2.1.3 Receive_TrackSide_Msg in Manage_TrackSideInformation_Integration

Reference to the SRS (or other requirements)

- [1, Chapt. 7 and 8]: Definition of the Balise Telegram
- [2, Chapt. 4.2.2, 4.2.4, 4.2.9]: Interface to the BTM
- [1, Chapt. 3.4.1 - 3.4.3, 3.16.2]: Handling of Balise Telegrams
- [1, Chapt. 3.16.2]: Check of the balise group
- [1, Chapt. 3.4.2]: Determining the orientation
- [1, Chapt. 4.5.2]: Active Functions Table
- [1, Chapt. 8.4.4]: Rules for Euroradio messages

Short description of the functionality

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.

Interface

Functional Design Description

Design Constraints and Choices

1. The decoding of balises is done at the API. Also, packets received via the interface are already transformed into a usable shape.
2. Only packets used inside the current model are passed via the interface.
3. 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.
4. 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.
5. This function does not process information from the packets. The information is passed to the check without further processing of the values.

Reference to the Scade Model

The SCADE model can be found on GitHub under the following path: https://github.com/openETCS/modeling/tree/master/model/Scade/System/ObuFunctions/ManageLocationRelatedInfor
BaliseGroup/Receive_TrackSide_Msg

2.2.1.4 CheckBGConsistency in Manage_TrackSideInformation_Integration

Reference to the SRS or other Requirements (or other requirements)

- [1, Chapt. 7 and 8]: Definition of the Balise Telegram
- [1, Chapt. 3.4.1 - 3.4.3, 3.16.2]: Handling of Balise Telegrams
- [1, Chapt. 3.16.2]: Check of the balise group
- [1, Chapt. 4.5.2]: Active Functions Table

Short description of the functionality

This function has the task to verify 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.

Interface

An input of the operator is a list of received telegrams. After consistency check of the telegrams' list, the function generates the balise-group-message. This function is active in certain modes and the output and reactions are dependent on if the linking information is used.

Functional Design Description

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" .

Reference to the Scade Model

The SCADE model can be found on github under the following path: `https://github.com/openETCS/modeling/tree/master/model/Scade/System/ObuFunctions/ManageLocationRelatedInfor BaliseGroup/CheckBGConsistency`

2.2.1.5 CheckEuroradioMessage in Manage_TrackSideInformation_Integration

Reference to the SRS or other Requirements (or other requirements)

- [1, Chapt. 8.4.4]: Rules for Euroradio messages
- [1, Chapt. 3.16]: Data consistency

Short description of the functionality

The operator "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.

Interface

The operator expects a radio message, information about the timestamp of the last relevant event, the national value for timeout (`T_NVCONTACT`) and the status of the radio connection as input. The operator provides as output the radio message marked as valid or invalid in the message header and updated metadata in the message. Errors are indicated through boolean outputs.

Functional Design Description

The operator performs the following checks:

- Content checks
  - The whole message must be complete and contains all necessary fields. [1, 3.16.1.1]
  - The message must respect the ETCS language. [1, 3.16.1.1] The operator checks, if all necessary variables and packets for the corresponding message are part of the message and if no packets and variables are included in the message, which are not allowed.
- Timing checks
  - Check if the timestamp of a message is greater than the timestamp of the message received before. [1, 3.16.3.3.3]
  - If a message contains the timestamp "Unknown", check if this message is part of the initiation of the communication session. [1, 3.16.3.3.4]
  - Perform the check with the current message n: $T\_TRAIN_n <= T\_TRAIN_{n-1}+T\_NVCONTACT$ [1, 3.16.1.1]. This ensures, that the message was received in due time.

For inconsistent messages, the following actions are performed by the operator:

- If a message is not consistent, it shall be rejected. [1, 3.16.3.1.1.1] For this purpose, the message is marked as invalid, so it can be rejected by the operators using the output of the CheckEuroradioMessage-operator.
- The RBC shall be informed, when a message was rejected. [1, 3.16.3.1.1.2] Therefore the necessary information for creating an error report is provided as boolean flags.
- This operator will not trigger the reaction for an interrupted radio connection to the RBC. The reaction sepcified by `M_NVCONTACT` will be triggered by the RBC session management module.

The check by the Euroradio-protocol [1, 3.16.3.1.1] will not be performed by the operator, but on a lower level (RTM or openETCS-API).

The operator is not responsible for checking the integrity of the message in the bitstream format. The bitwalker, which is converting the bitstream into the internally used datatypes, is performing the checks, which are only possible on the raw data. This includes a CRC check on the whole message and a value range check for each variable. If a consistency error is detected by the bitwalker, it is signalled to the model. If the bitwalker marks a packet as valid, all variables are expected to contain a valid value. The bitwalker is not part of the ETCS kernel.

Safe connection supervision is not in the scope of this operator. This functionality will be implemented by the “Management of Radio communication” module.

Reference to the Scade Model

The SCADE model can be found on github under the following path: `https://github.com/openETCS/modeling/tree/master/model/Scade/System/ObuFunctions/ManageLocationRelatedInfor BaliseGroup/CheckEuroRadioMessage`

2.2.1.6 ValidateDataDirection in Manage_TrackSideInformation_Integration

Reference to the SRS or other Requirements (or other requirements)

- The functionality is mainly described in [1, Chapter 3.6.3].

Short description of the functionality

The operator will filter an input message in order to mark all elements as invalid, which are not designated for the current driving direction of the train.

Interface

The operator expects a message and information about the LRBG, passed balises and the current train position. As output, the message with packets valid for the current direction of driving is provided.

Functional Design Description

- 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.

Reference to the Scade Model

The SCADE model can be found on github under the following path: `https://github.com/openETCS/modeling/tree/master/model/Scade/System/ObuFunctions/ManageLocationRelatedInfor BaliseGroup/ValidateDataDirection`

2.2.1.7 InformationFilter

Reference to the SRS and other requirements

- The functionality of the InformationFilter is described in [1, Chapter 4.8].

Short description of the functionality

The function InformationFilter filters incoming information received from Eurobalise, Euroradio, and Euroloop. The information is received via messages and filtering is done depending on the criteria described in [1, Chapter 4.8]. Messages are only allowed to pass the filter if specified critera are met like for example the correct mode of the train (e.g. Full Supervision, Shunting, etc.) or the ETCS level. Some messages have to be stored in a TransitionBuffer to be later reevaluated by the filter again.

Interface

The interface of the information filter contains mainly the incoming message and inputs to check for the conditions described in the SRS. The complete interface is shown in table 11.

Name	Direction	Description
`inMessage`	`IN`	Received message that is valid for the train direction
`inLevel`	`IN`	The current ETCS Level
`inMode`	`IN`	The current train mode
`inSupervisingDevice`	`IN`	The device id which communicates with the current supervising RBC
`inPendingL1Transition`	`IN`	Information if an ETCS Level 1 transition is pending
`inPendingL1L2Transition`	`IN`	Information if an ETCS Level 2/3 transition is pending
`inPendingNTCTransition`	`IN`	Information if a NTC transition is pending
`inPendingAckOfTrainData`	`IN`	Information if the acknowledgement of train data is pending
`inEmergencyBrakeActive`	`IN`	Information if the emergency brake is active
`inLastAckTextMessageId`	`IN`	The id of the last acknowledged message ID
`inActiveCab`	`IN`	Information if the cab is active
`inTrainDataValid`	`IN`	Information if the train data is valid
`outMessage`	`OUT`	The filtered input message

Table 11. Overview of the InformationFilter interface

Functional Design Description

The filter receives track information (balise and radio) and filter them depending of the mode, level and further information. Only messages that pass the filter are valid and should be considered

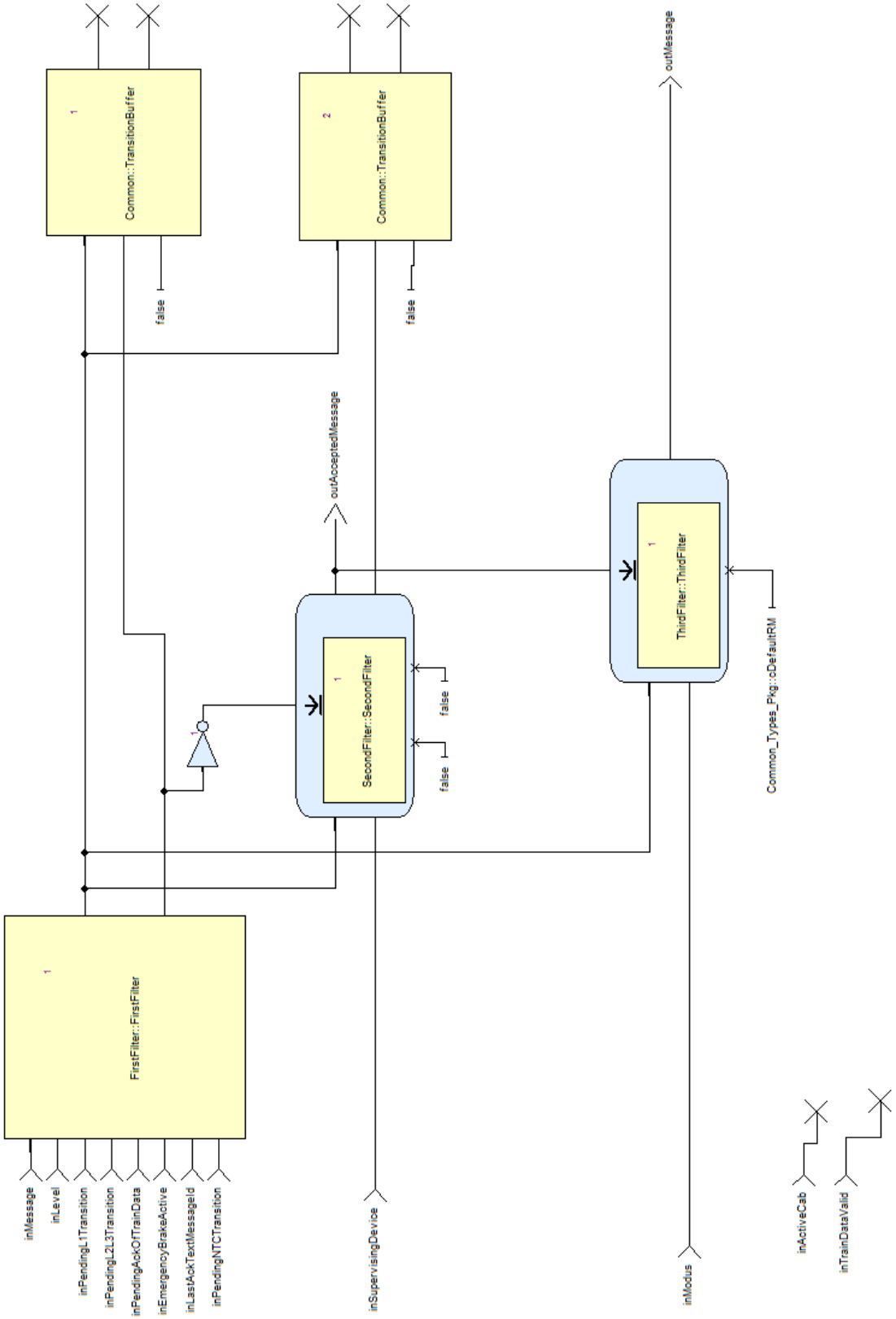


Figure 3. High level overview of the InformationFilter components.

by other ETCS subsystems. The figure 3 show the highlevel decomposition of the functionality. The filter functionality can be decomposed into a FirstFilter, SecondFilter, ThirdFilter an 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.

A detailed list of packages and their handling depending of ETCS level or mode can be seen in table 4 and 5.

Reference to the Scade Model

The SCADE model can be found on github under the following path: `https://github.com/openETCS/modeling/tree/master/model/Scade/System/ObuFunctions/ManageLocationRelatedInfor BaliseGroup/InformationFilter`

2.3 F3: Measure Train Movement

2.4 F4: Manage Radio Communication

2.5 F5: Manage JRU

2.6 F6: DMI Controller

Package/Variables	Information	From RBC	Onboard operating level				
			0	NTC	1	2	3
Packet 3	National Values	No	A	A	A	A	A
		Yes	R [2]	R [2]	R [2]	A	A
Packet 5	Linking	No	R [1]	R [1]	A	R [1]	R [1]
		Yes	R [2]	R [2]	R [2]	A [3]	A [3]
V_Main Packet 12	Signalling Related Speed Restriction	No	R [1]	R [1]	A	R [1]	R [1]
		Yes					
Packet 12, 15 Packet 80 Packet 49	Movement Authority + (optional) Mode Profile + (optional) List of Balises for SH area	No	R [1]	R [1]	A [4]	R [1]	R [1]
		Yes	R [2]	R [2]	R [2]	A [3] [4] [5]	A [3] [4] [5]
Packet 16	Repositioning Information	No	R	R	A	R	R
		Yes					
Packet 21	Gradient Profile	No	R [1]	R [1]	A	R [1]	R [1]
		Yes	R [2]	R [2]	R [2]	A [3]	A [3]
Packet 27	International SSP	No	R [1]	R [1]	A	R [1]	R [1]
		Yes	R [2]	R [2]	R [2]	A [3]	A [3]
Packet 51	Axle Load speed profile	No	R [1]	R [1]	A	R [1]	R [1]
		Yes	R [2]	R [2]	R [2]	A [3]	A [3]
Packet 41	Level Transition Order	No	A	A	A	A	A
		Yes	A	A	A	A	A
Packet 46	Conditional Level Transition Order	No	A [11]	A [11]	A [11]	A [11]	A [11]
		Yes					
Packet 42	Session Management	No	A	A	A	A	A
		Yes	A	A	A	A	A
Packet 45	Radio Network registration	No	A	A	A	A	A
		Yes	A	A	A	A	A
Packet 57	MA Request Parameters	No					
		Yes	A	A	A	A	A
Packet 58	Position Report parameters	No					
		Yes	A	A	A	A	A
Package 63 + Message Radio 2 + (optional) Packet 49	SR Authorisation + (optional) List of Balises in SR mode	No					
		Yes	R	R	R	A [3]	A [3]
Packet 137	Stop if in SR mode	No	R	R	A	A	A
		Yes					
D_SR in Packet 13	SR distance information from loop	No	R	R	A	R	R
		Yes					

Packet 65	Temporary Speed Restriction	No	A	R [1] [2]	A	A [8]	A [8]
		Yes	R [2]	R [2]	R [2]	A [3]	A [3]
Packet 66	Temporary Speed Restriction Revocation	No	A	R [1] [2]	A	A	A
		Yes	R [2]	R [2]	R [2]	A [3]	A [3]
Package 64	Inhibition of revocable TSRs from balises in L2/3	No					
		Yes	R [2]	R [2]	R [2]	A	A
Packet 141	Default Gradient for TSR	No	A	R [1] [2]	A	A	A
		Yes					
Packet 70	Route Suitability Data	No	R [1]	R [1]	A	R [1]	R [1]
		Yes	R [2]	R [2]	R [2]	A [3]	A [3]
Packet 71	Adhesion Factor	No	R[1]	R[1]	A	R	R
		Yes	R[2]	R[2]	R[2]	A	A
Packet 72	Plain Text Information	No	A	R [1] [2]	A	A	A
		Yes	R [2]	R [2]	R [2]	A [12]	A [12]
Packet 76	Fixed Text Information	No	A	R [1] [2]	A	A	A
		Yes	R [2]	R [2]	R [2]	A [12]	A [12]
Packet 79	Geographical Position	No	A	R [1] [2]	A	A	A
		Yes	R [2]	R [2]	R [2]	A	A
Packet 131	RBC Transition Order	No	R	R	R	A	A
		Yes	R	R	R	A [3]	A [3]
Packet 132	Danger for SH information	No	A [13]	A [13]	A	A	A
		Yes					
Package 135	Stop Shunting on desk opening	No	A	A	A	A	A
		Yes					
Packet 133	Radio Infill Area information	No	R	R	A	R [1]	R [1]
		Yes					
Package 42	Session Management with neighbouring RIU	No	R	R	A	R	R
		Yes					
Packet 134	EOLM information	No	A	A	A	A	A
		Yes					
Messager 45	Assignment of Co-ordinate system	No					
		Yes	A [10]	A [10]	A [10]	A [10]	A [10]
Packet 136	Infill Location Reference	No	R	R	A	R [1]	R [1]
		Yes					
Packet 39, Packet 68	Track Conditions excluding big metal masses	No	R [1]	R [1]	A	R [1]	R [1]
		Yes	R [2]	R [2]	R [2]	A [3]	A [3]
Packet 67	Track condition big metal masses	No	A	A	A	A	A
		Yes					

Header Balise	Location Identity (NID_C + NID_BG transmitted in the balise telegram)	No	A	A	A	A	A
		Yes					
Radio Message 6	Recognition of exit from TRIP mode	No					
		Yes	R	R	R	A	A
Message Radio 8	Acknowledgement of Train Data	No					
		Yes	A	A	A	A	A
Message 9 Packet 80 Packet 49	Co-operative shortening of MA + (optional) Mode Profile + (optional) List of Balises for SH area	No					
		Yes	R	R	R	A [3] [4] [5]	A [3] [4] [5]
Message Radio 16	Unconditional Emergency Stop	No					
		Yes	R [2]	R [2]	R [2]	A	A
Message Radio 15	Conditional Emergency Stop	No					
		Yes	R [2]	R [2]	R [2]	A	A
Message Radio 18	Revocation of Emergency Stop (Conditional or Unconditional)	No					
		Yes	R	R	R	A	A
Message Radio 27	SH refused	No					
		Yes	R	R	R	A [3]	A [3]
Message Radio 28 + (optional) Packet 49	SH authorised + (optional) List of Balises for SH area	No					
		Yes	R	R	R	A [3]	A [3]
??	Trackside constituent System Version	No	A	A	A	A	A
		Yes	A	A	A	A	A
Packet 2	System Version order	No	A	A	A	A	A
		Yes					
Message Radio 34	Track Ahead Free Request	No					
		Yes	R	R	R	A [3]	A [3]
Packet 140 Track to train, Packet 40 Train to track	Train Running Number	No					
		Yes	R	R	R	A	A
Message Radio 38	Initiation of session	No					
		Yes	R	R	R	A	A
Message 39	Acknowledgement of session termination	No	A	A	A	A	A
		Yes	A	A	A	A	A

Message 40	Train Rejected	No					
		Yes	R	R	R	A	A
Message 41	Train Accepted	No					
		Yes	R	R	R	A	A
Message Radio 43	SoM Position Report Confirmed by RBC	No					
		Yes	R	R	R	A	A
Packet 138	Reversing Area Information	No	R [1]	R [1]	A	R [1]	R [1]
		Yes	R [2]	R [2]	R [2]	A [3]	A [3]
Packet 139	Reversing Supervision Information	No	R [1]	R [1]	A	R [1]	R [1]
		Yes	R [2]	R [2]	R [2]	A [3]	A [3]
Packet 254	Default Balise/Loop/RIU Information	No	A	A	A	A	A
		Yes					
Packet 90	Track Ahead Free up to level 2/3 transition location	No	A [9]	A [9]	A [9]	R	R
		Yes					
Package 52	Permitted Braking Distance Information	No	R [1]	R [1]	A	R [1]	R [1]
		Yes	R [2]	R [2]	R [2]	A [3]	A [3]
Package 88	Level Crossing information	No	R [1] [2]	R [1] [2]	A	A	A
		Yes	R [2]	R [2]	R [2]	A [3]	A [3]
Package 6(0)	Virtual Balise Cover order	No	A	A	A	A	A
		Yes					
Package 44	Data to be used by applications outside ERTMS/ETCS	No	A	A	A	A	A
		Yes	A	A	A	A	A
	Information from National System X through STM interface	Onboard operating level					
		0	NTC X	NTC Y	1	2	3
??	STM max speed	A [7]	R	R [6]	A [7]	A [7]	A [7]
??	STM system speed/distance	A [7]	R	R	A [7]	A [7]	A [7]

Figure 4. Lists of packages and their handling depending on train modes

40	Packet 39, 68	Track conditions sound horn, non stopping areas, tunnel stopping areas	NR	A[2][4]	R	R	A	A	A	A	A	R	R	A	A[1]	NR	NR	A	R
41	**Packet 67**	Track condition big metal masses	NR	A[2][4]	A	A	A	A	A	A	A	A	A	A	A[1]	NR	NR	A	R
42	Header Balise	Location Identity (NID_C + NID_BG)	NR	A[2]	A	A	A	A	A	A	A	A	A	A	A	NR	NR	A	A
43	Message Radio 6	Recognition of exit from TRIP mode	NR	R	R	R	R	R	R	R	R	R	R	R	A	NR	NR	R	R
44	Message Radio 8	Acknowledgement of Train Data	NR	A[2]	R	R	A	A	A	A	R	A	R	A	A	NR	NR	A	A
45	Message 9	Co-operative shortening of MA	NR	R	R	R	A	A	R	A	R	R	R	R	R	NR	NR	R	R
46	**Packet 80**	+ (optional) Mode Profile																	
47	**Packet 49**	+ (optional) List of Balises for SH area																	
48	Message Radio 16	Unconditional Emergency Stop	NR	A[2]	R	R	A	A	A	A	R	A	R	A	R	NR	NR	A	R
49	Message Radio 15	Conditional Emergency Stop	NR	R	R	R	A	A	R	A	R	A	R	A	R	NR	NR	A	R
50	Message Radio 18	Revocation of Emergency Stop	NR	R	R	R	A	A	R	A	R	R	R	R	A[1]	NR	NR	R	R
51		(Conditional or Unconditional)																	
52	Message Radio 27	SH refused	NR	A[2]	R	R	A	A	A	A	R	R	R	R	A[1]	NR	NR	R	R
53	Message Radio 28 + (optional) **Packet 49**	SH authorised + (optional) List of Balises for SH area	NR	A[2]	R	R	A	A	A	A	R	R	R	R	A[1]	NR	NR	R	R
54	??	Trackside constituent System Version	NR	A	A	A	A	A	A	A	A	A	A	A	A	NR	NR	A	A
55	**Packet 2**	System Version order	NR	A	A	A	A	A	A	A	A	A	A	A	A	NR	NR	A	A
56	Message Radio 34	Track Ahead Free Request	NR	A[2]	R	R	R	A	A	A	R	R	R	R	A[1]	NR	NR	R	R
57	**Packet 140 Track to train, Packet 40 Train to track**	Train Running Number	NR	A[2]	R	R	A	A	A	A	R	A	R	A	A	NR	NR	R	A
58	Message Radio 38	Initiation of session	NR	A	R	R	A	A	A	A	A	A	R	A	A	NR	NR	R	A
59	Message 39	Acknowledgement of session termination	NR	A	A	A	A	A	A	A	A	A	A	A	A	NR	NR	A	A
60	Message 40	Train Rejected	NR	A[2]	R	R	R	R	R	R	R	R	R	R	R	NR	NR	R	R
61	Message 41	Train Accepted	NR	A[2]	R	R	R	R	R	R	R	R	R	R	R	NR	NR	R	R
62	Message Radio 43	SoM Position Report Confirmed by RBC	NR	A[2]	R	R	R	R	R	R	R	R	R	R	R	NR	NR	R	R
63	**Packet 138**	Reversing Area information	NR	A[2][4]	R	R	A	A	A	A	R	R	A	R	A[1]	NR	NR	A	A
64	**Packet 139**	Reversing Supervision Information	NR	A[2][4]	R	R	A	A	A	A	R	R	A	R	A[1]	NR	NR	A	A
65	**Packet 254**	Default Balise/Loop/RIU Information	NR	A[2]	A	A	A	A	A	A	A	A	A	A	A	NR	NR	A	A
66	**Packet 90**	Track Ahead Free up to level 2/3 transition location	NR	A[2]	R	R	A	A	A	A	R	R	A	A	A	NR	NR	A	R
67	**Package 52**	Permitted Braking Distance Information	NR	A[2][4]	R	R	A	A	A	A	R	R	A	R	A[1]	NR	NR	A	R
68	**Package 88**	Level Crossing information	NR	A[2][4]	R	R	A	A	A	A	R	R	A	R	A[1]	NR	NR	A	R
69	**Package 60)**	Virtual Balise Cover order	NR	A	A	A	A	A	A	A	A	A	A	A	A	NR	NR	A	A
70	**Package 44**	Data to be used by applications outside ERTMS/ETCS	NR	A	A	A	A	A	A	A	A	A	A	A	A	NR	NR	A	A

	Packet, Message	Information	Modes																	
			NP	SB	PS	SH	FS	LS	SR	OS	SL	NL	UN	TR	PT	SF	IS	SN		RV
1	Packet 3	National Values	NR	A[2]	A	A	A	A	A	A	A	A	A	A	A[1]	NR	NR	A		A
2	Packet 5	Linking	NR	AD[4]	R	R	A	A	A	A	R	A	A	R	A[1]	NR	NR	A		R
3	V_Main in Packet 12	Signalling Related Speed Restriction	NR	AD[4]	R	R	A	A	A	A	R	R	A	R	A[1]	NR	NR	A		R
4	Packet 12, 15	Movement Authority	NR	AD[4]	R	R	A	A	A	A	R	R	A	R	A[1]	NR	NR	A		R
5	(optional) Packet 80	+ (optional) Mode Profile																		
6	(optional) Packet 49	+ (optional) List of Balises for SH area																		
7	Packet 16	Repositioning Information	NR	R	R	R	A	A	R	A	R	R	R	R	R	NR	NR	R		R
8	Packet 21	Gradient Profile	NR	AD[4]	R	R	A	A	A	A	R	R	A	R	A[1]	NR	NR	A		R
9	Packet 27	International SSP	NR	AD[4]	R	R	A	A	A	A	R	R	A	R	A[1]	NR	NR	A		R
10	Packet 51	Axle load speed profile	NR	AD[4]	R	R	A	A	A	A	R	R	A	R	A[1]	NR	NR	A		R
11	??	STM max speed	NR	A[2]	R	R	A	A	A	A	R	R	A	A	A[1]	NR	NR	A		R
12	??	STM system speed/distance	NR	A[2]	R	R	A	A	A	A	R	R	A	A	A[1]	NR	NR	R		R
13	Packet 41	Level Transition Order and Conditional Level Transition Order	NR	A[2]	A[7]	A[7]	A	A	A	A	A	A	A	A	A[1][5]	NR	NR	A		R
14	Packet 42	Session Management	NR	A	A[3]	A[3]	A	A	A	A	A	A	A	A	A[1]	NR	NR	A		A
15	Packet 49	Radio Network registration	NR	A[2]	A	A	A	A	A	A	A	A	A	A	A[1]	NR	NR	A		A
16	Packet 57	MA Request Parameters	NR	A[2]	R	R	A	A	A	A	R	R	A	R	A[1]	NR	NR	A		R
17	Packet 58	Position Report parameters	NR	A[2]	R	R	A	A	A	A	R	A	A	R	A[1]	NR	NR	A		A
18	Package Radio 63 + Message Radio 2 + (optional) Packet 49	SR Authorisation+ (optional) List of Balises in SR mode	NR	AD[4]	R	R	R	R	A	R	R	R	R	R	A[1]	NR	NR	R		R
20	Packet 137	Stop if in SR mode	NR	R	R	R	R	R	A	R	R	R	R	R	R	NR	NR	R		R
21	D_SR in Packet 13	SR distance information from loop	NR	R	R	R	R	R	A[6]	R	R	R	R	R	R	NR	NR	R		R
22	Packet 65	Temporary Speed Restriction	NR	AD[4]	R	R	A	A	A	A	R	R	A	A	A[1]	NR	NR	A		R
23	Packet 66	Temporary Speed Restriction Revocation	NR	AD[4]	R	R	A	A	A	A	R	R	A	A	A[1]	NR	NR	A		R
24	Package 64	Inhibition of revocable TSRs from balises in L2/3	NR	A[2]	R	R	A	A	A	A	R	R	A	A	A[1]	NR	NR	A		R
25	Packet 141	Default Gradient for TSR	NR	AD[4]	R	R	A	A	A	A	R	R	A	A	A[1]	NR	NR	A		R
26	Packet 70	Route Suitability Data	NR	AD[4]	R	R	A	A	A	A	R	R	A	R	A[1]	NR	NR	A		R
27	Packet 71	Adhesion Factor	NR	AD[4]	R	R	A	A	A	A	R	R	A	R	A[1]	NR	NR	A		R
28	Packet 72	Plain Text Information	NR	A[2]	R	R	A	A	A	A	R	A	A	A	A[1]	NR	NR	A		A
29	Packet 76	Fixed Text Information	NR	A[2]	R	R	A	A	A	A	R	A	A	A	A[1]	NR	NR	A		A
30	Packet 78	Geographical Position	NR	A[2]	R	R	A	A	A	A	R	A	A	A	A[1]	NR	NR	A		R
31	Packet 131	RBC Transition Order	NR	AD[4]	A[8]	A[8]	A	A	A	A	A	A	R	A	A[1]	NR	NR	R		R
32	Packet 132	Danger for SH information	NR	R	R	A	R	R	R	R	R	R	R	R	R	NR	NR	R		R
33	Package 135	Stop Shunting on desk opening	NR	R	A	R	R	R	R	R	R	R	R	R	R	NR	NR	R		R
34	Package 133	Radio Infill Area information	NR	R	R	R	A	A	A	A	R	R	R	R	R	NR	NR	R		R
35	Package 42	Session Management with (non-)equipped RBC	NR	R	R	R	A	A	A	A	R	R	R	R	R	NR	NR	R		R
36	Packet 134	EOLM information	NR	R	R	A	A	A	A	A	A	A	A	A	R	NR	NR	A		A
37	Message Radio 45	Assignment of Co-ordinate system	NR	A[2]	R	R	R	R	A	R	R	A	A	R	A[1]	NR	NR	A		R
38	Package 136	Infill Location Reference	NR	R	R	R	A	A	R	R	R	R	R	R	R	NR	NR	R		R
39	Packet 39, 68	Track Conditions excluding sound horn, non stopping areas, tunnel stopping areas and big metal masses	NR	AD[4]	R	R	A	A	A	A	R	A	A	A	A[1]	NR	NR	A		R

Figure 5. Lists of packages and their handling depending on train modes

References

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

[2] ERA. *FFFIS for Eurobalise, SUBSET-036*, v3.0.0 edition, February 2012.