ENSEMBLE

EUROPEAN COMMISSION

HORIZON 2020
H2020-ART-2016-2017/H2020-ART-2017-Two-Stages
GA No. 769115

ENSEMBLE

ENabling **S**af**E** **M**ulti-**B**rand p**L**atooning for **E**urope

Deliverable No.	D3.1	
Deliverable Title	Detailed design of the unbranded Tactical Layer	
Dissemination level	Public	
Written By	Dr. A.J.C. Schmeitz, G.J.A. Verhaeg, M. Fusco, TNO	23-07-2019
Checked by	WP Leader: D.M.C. Willemsen, TNO	23-07-2019
Approved by	Coordinator: Dr. D.M. Hoedemaeker, TNO	24-07-2019

Status	FINAL, approved by EC	05-08-2022

Please refer to this document as:

Schmeitz, A.J.C., Verhaeg, G.J.A., Fusco, M. (2019). *Detailed design of the unbranded Tactical Layer.* D3.1 of H2020 project ENSEMBLE, (www.platooningensemble.eu).

Acknowledgement:

The authors would like to thank the task participants, DAF, Daimler, Iveco, MAN, Scania, Volvo and CLEPA for their contributions leading to the design of the Tactical Layer modules presented in this deliverable.

Disclaimer:

ENSEMBLE is co-funded by the European Commission, DG Research and Innovation, in the HORIZON 2020 Programme. The contents of this publication is the sole responsibility of the project partners involved in the present activity and do not necessarily represent the view of the European Commission and its services nor of any of the other consortium partners.

TABLE OF CONTENTS

Revision history

Version	Date	Author	Summary of changes	Status
0.1	23/07/2019	A.J.C. Schmeitz (TNO)	Initial version	Prepared
0.2	23/07/2019	D.M.C. Willemsen (TNO)	Report review by WP 3 Leader	Draft
0.3	24/07/2019	D.M. Hoedemaeker (TNO)	Report review by Coordinator	Draft
1.0	25/07/2019	A.J.C. Schmeitz (TNO)	Updated based on review comments	Final

FIGURES

TABLES

1. EXECUTIVE SUMMARY

1.1. Context and need of a multi brand platooning project

Context

Platooning technology has made significant advances in the last decade, but to achieve the next step towards deployment of truck platooning, an integral multi-brand approach is required. Aiming for Europe-wide deployment of platooning, 'multi-brand' solutions are paramount. It is the ambition of ENSEMBLE to realise pre-standards for interoperability between trucks, platoons and logistics solution providers, to speed up actual market pick-up of (sub)system development and implementation and to enable harmonisation of legal frameworks in the member states.

Project scope

The main goal of the ENSEMBLE project is to pave the way for the adoption of multi-brand truck platooning in Europe to improve fuel economy, traffic safety and throughput. This will be demonstrated by driving up to seven differently branded trucks in one (or more) platoon(s) under real world traffic conditions across national borders. During the years, the project goals are:

- Year 1: setting the specifications and developing a reference design with acceptance criteria
- Year 2: implementing this reference design on the OEM own trucks as well as perform impact assessments with several criteria
- Year 3: focus on testing the multi-brand platoons on test tracks and international public roads

The technical results will be evaluated against the initial requirements. Also, the impact on fuel consumption, drivers and other road users will be established. In the end, all activities within the project aim to accelerate the deployment of multi-brand truck platooning in Europe.

1.2. Abstract of this deliverable

In the context of the ENSEMBLE project, T 3.1 is an umbrella task in WP 3. The objective of WP 3 is the implementation of the requirements and specifications of WP 2 in demonstrator trucks (i.e. comprising hardware and software). This implementation includes the operational and the tactical layer, as well as the interface to the strategic layer. WP 3 focusses on the implementation of platooning as a support function (in D2.4 referred to as level A). The automation consists of longitudinal automation with optional lateral support. The aim of T 3.1 is the development and prototyping of a reference tactical layer and V2X design, which compromise the common multi-brand functionality, according to the specifications of WP 2.

Deliverable 3.1 (D3.1) consists of 1) a reference implementation in software of the tactical layer modules, i.e. the Platoon Coordinator, and 2) a report describing this design in detail (this document). This report, i.e. the current document, is Project Milestone 3 (MS3): Reference design of the tactical layer. The source code of the software of the Platoon Coordinator is made available to the members of the Consortium (including the Commission Services) via the ENSEMBLE SharePoint site. The relation of D3.1 with other T3.1 deliverables is that the V2X reference design is D3.2 and the prototyping of the total reference design, i.e. tactical layer and V2X design, in a rapid control prototyping setup is D3.3.

In this document the design of the tactical layer modules is described, starting from the specifications defined in WP 2. The tactical layer modules consist of a manoeuvre coordinator and a platoon status and property sharing module. The manoeuvre coordinator is responsible for handling the sequences of the interaction protocol to conduct the manoeuvres to join, leave or split the platoon. The platoon status and property sharing module ensures the collection and sharing of information that must be available to all vehicles in the platoon.

Next to the design, the implementation of the design in software is described. The Simulink programming environment of the MathWorks is used to develop the Platoon Coordinator model. This model is set up such that it can interface with and be integrated in the white label truck model, specified in WP 2. This white label truck model can be implemented in a HIL facility, which includes the reference V2X communication device (as hardware unit), for prototyping the reference design. Note that before the model can be embedded in the HIL facility it has to be compiled on a Rapid Control Prototyping device. In this compiling step, the readability of the software is lost. Therefore, this deliverable focusses on describing the (readable) source code, i.e. the Simulink model, of the Platoon Coordinator.

In order to understand and test the Platoon Coordinator model without having to integrate it in a truck environment, a Simulink model for testing is provided. This test model is also described in this document.

In conclusion, this document describes the reference design and software implementation of the tactical layer modules, i.e. the Platoon Coordinator. The design has to be considered as the first version, which is the baseline for implementation and further testing. It is likely that new insights obtained during implementation and testing in WP 3 and/or WP5 will lead to future updates of the reference design. Version management will be used to track these changes. Finally, feedback to WP 2 is continuously provided if required changes affect the requirements and specifications.

2. INTRODUCTION

2.1. Background

In the context of the ENSEMBLE project, T 3.1 is an umbrella task in WP 3. The objective of WP 3 is the implementation of the requirements and specifications of WP 2 in demonstrator trucks (i.e. comprising hardware and software). This implementation includes the operational and the tactical layer, as well as the interface to the strategic layer. WP 3 focusses on the implementation of platooning level A, which means that platooning will be implemented as a driver support function. The automation consists of longitudinal automation with optional lateral support. The aim of T 3.1 is the development and prototyping of a reference tactical layer and V2X design, which compromise the common multi-brand functionality, according to the specifications of WP 2.

Deliverable 3.1 (D3.1) consists of 1) a reference implementation in software of the tactical layer modules, i.e. the Platoon Coordinator, and 2) a report describing this design in detail (this document). This report, i.e. the current document, is Project Milestone 3 (MS3): Reference design of the tactical layer.

2.1.1. Specification of the Reference Design

In WP 2 the specification for the multi-brand truck platooning concept, i.e. white-label truck, is defined. The following documents serve as (direct) inputs to the specification of the reference design and implementation:

- ENSEMBLE Deliverable D2.2 (Vissers, 2018), *V1 Platooning use-cases, scenario definition and Platooning Levels*, Final version 19-12-2018 (pending EC approval).
- ENSEMBLE Deliverable D2.4 (Konstantinopoulou, 2019), *Functional specification for white-label truck*, Final version 15-2-2019 (pending EC approval).
- ENSEMBLE Deliverable D2.8 (Atanassow, 2019a), *Platooning protocol definition and Communication strategy*, Final version 12-12-2018 (pending EC approval).

ENSEMBLE Deliverable D2.2 introduces a system overview (e.g., in terms of operational, tactical, strategic and service layers), platoon levels and use cases. In particular, the use cases serve as important input for identification of (state-changing), required information flows, and interactions between manoeuvring, control and communication. The deliverable constrains the scope of the project to its defined Platooning Level A (platooning as a driver support function with longitudinal automation and optionally lateral support).

ENSEMBLE Deliverable D2.4 provides the definition of the requirements and specifications of the white-label multi-brand truck platooning concept to be implemented, tested and demonstrated with up to trucks of 6 different European OEMs. The white-label truck concept takes into consideration Platoon level A which will form the basis of the intended demonstration at the end of the project on public road. D2.4 concentrates on the operational and tactical layer, but also identifies required interactions with the Strategic and Services Layers.

ENSEMBLE Deliverable D2.8 presents the communication strategy (following ETSI TC ITS ENs, mostly) and ENSEMBLE-specific platooning protocol definition in terms of messages and message sequences. In particular, the document provides an ASN.1 description of the platooning messages (Protocol Data Units, or PDUs). Deliverable D2.8 was submitted in December 2018.

After the publication of D2.2, D2.4 and D2.8, new insights required changes to the original ENSEMBLE protocol definition reported in the deliverables. These changes can be found on SharePoint in the T2.3 folders. Regarding this deliverable D3.1, the ASN.1 definitions and sequence diagrams of the ‘M13 version’ of 5-7-2019 (Atanassow, 2019b) are used.

2.1.2. Approach

In several WP 3 conference calls and physical meetings, also some with WP 2 and WP 5, the initial specifications and requirements have been discussed and reviewed. This has led to 1) updated specifications and 2) ideas for the reference design, which is described in this document. The minutes and presentations of these meetings can be found on the ENSEMBLE SharePoint site. Moreover, a change request table has been setup and made available on the project SharePoint.

2.1.3. Relation to other deliverables

The relation of D3.1 with other T3.1 deliverables is that the V2X reference design is D3.2 and the prototyping of the total reference design, i.e. tactical layer and V2X design, in a rapid control prototyping setup is D3.3.

2.2. Aim

This report describes the reference design of the tactical layer modules, i.e. Platoon Coordinator, in order to explain the design and guide the reader through the developed software.

2.3. Structure of this report

In Chapter 3, the design of the tactical layer modules, i.e. Platoon Coordinator, is described. The implementation of the design in software is the subject of Chapter 4. In Chapter 5, it is explained how the Platoon Coordinator can be tested and evaluated using a simulation model. A brief summary of the report and conclusions are provided in Chapter 6. Note, a glossary of used terms is provided in Appendix A.

3. DESIGN OF THE TACTICAL LAYER MODULES

In this chapter the design of the tactical layer is described. The design is based on the requirements and specifications, which are defined in WP2. In the description of the design, references are made to these requirements and specifications. The chapter first describes the overall architecture of the white label truck to indicate where the tactical layer modules fit in. After that the components of the tactical layer modules are explained in more detail.

3.1. Architecture

In D2.4 (Konstantinopoulou, 2019) the functional modules of the white label truck have been defined. In Figure 1 these modules are shown.

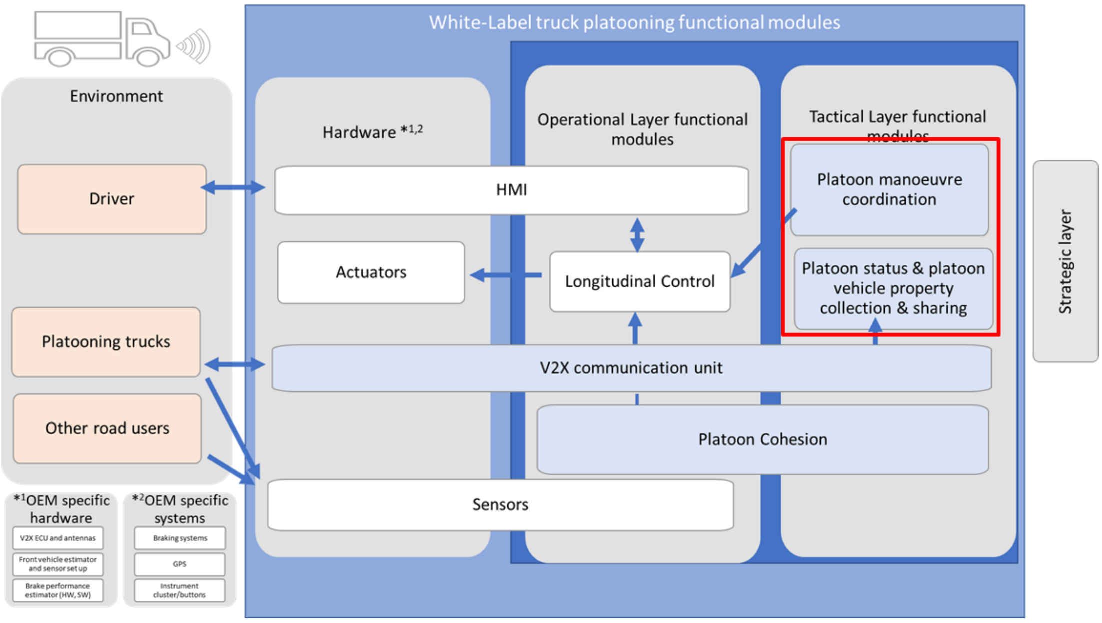


Figure 1: Platooning modules of the white-label truck from D2.4 (Konstantinopoulou, 2019).

The Tactical Layer functional modules are shown on the right. As can be seen in Figure 1, the Tactical Layer consists of two Tactical Layer modules, which are the Platoon manoeuvre coordinator and the Platoon status & platoon vehicle property collection & sharing modules, and two layer overlapping modules, which are the V2X communication and Platoon Cohesion modules. The V2X communication (unit) module contains the communication unit (hardware and embedded software) and the software relating to the ENSEMBLE-specific platooning protocol definition in terms of messages and message sequences. The operational ITS-G5 communication of the reference design is described in detail in D3.2 (de Jongh, 2019). In this deliverable communication is only addressed in the context of applying the ENSEMBLE-specific platooning protocol definition for the Tactical Layer modules.

The Platoon Cohesion module, as defined in D2.4, is active in both the operational and tactical layers. In the Operational Layer, cohesion problems are identified and dealt with, whereas the Tactical Layer ensures that relevant information for addressing cohesion problems is shared within the platoon. In D2.4 this information sharing is part of the Platoon vehicle property collection & sharing and will be dealt with in a similar way in this document, meaning that this cohesion functionality is part of the Platoon status & platoon vehicle property collection & sharing module. For the sake of clarity and simplicity, these two Tactical Layer modules are grouped into the Platoon Coordinator. In this way, the grouped software modules are distinguished from the Tactical Layer.

The Tactical Layer modules interact with the Operational Layer modules and the V2X communication. The V2X module is specified in detail in ENSEMBLE, as part of Task 2.3, because V2X communication needs to be standardized for multi-brand platooning. The Operational Layer modules are OEM specific and only high level requirements have been specified in D2.4. However, in order to have a detailed design for the Tactical Layer modules, the interfacing with the Operational Layer modules must be defined in detail as well. In Figure 2 the proposed interfacing of the Platoon Coordinator is shown.

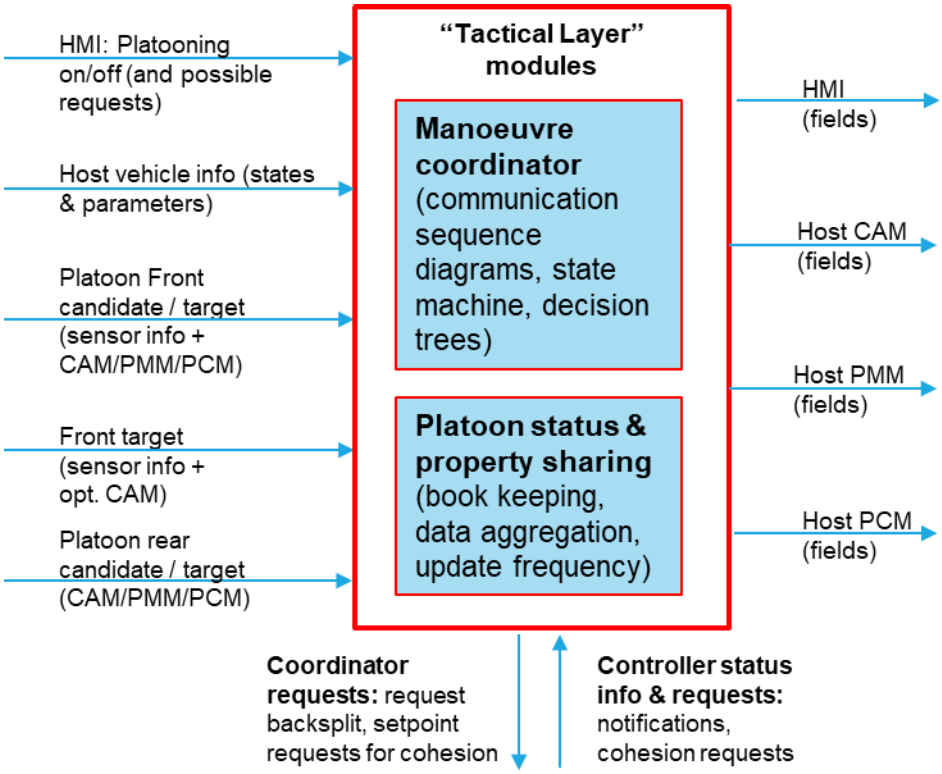


Figure 2: Interfacing of the platoon coordinator, consisting of Tactical Layer modules.

On the left side of the block, the inputs to the Platoon Coordinator are shown. These originate from the host vehicle's world model, consisting of the fused perception of objects/targets using vehicle sensors and V2X information, from the HMI and from host vehicle information obtained via vehicle sensors. On the right side of the block the outputs of the coordinator are shown, which are the fields to set for V2X communication and the HMI of the host vehicle. The interfaces on the bottom of the block show the interaction with the Longitudinal Control, consisting of coordinator and controller requests and controller status information.

Before the inputs form the world model, i.e. the targets, can be provided to the Platoon Coordinator, sensor fusion and V2X data association is required. The world model component gathers target information using different vehicle sensors, e.g. radar, camera and/or lidar. The obtained information is fused to get information of unique targets (other vehicles, motorcycles, etc.). Additionally, the world model obtains information of certain targets via V2X communication. This V2X target information must be associated with the corresponding targets. Typically the received GPS position and vehicle length of the targets are used for binding the V2X information to vehicle observed targets. The next step in the pre-processing process is to select the targets directly in front of and behind the host vehicle, driving in the same lane. In the Platoon Coordinator interface, the following 3 targets are distinguished:

- <u>Platoon front (candidate) target</u>: this is the target directly in front of the host vehicle, which is either joinable for platooning (i.e. isJoinable flag is TRUE in the CAM message or PMM messages are received, e.g. a join request) or with which the host vehicle is platooning (i.e. exchange of PCM is established).
- <u>Front target</u>: this is the target vehicle directly in front of the host vehicle, which is either:
    - identical to the Platoon front (candidate) target, or:
    - not sending V2X messages or is sending V2X messages that are not associated with platooning, or is sending CAM messages with an isJoinable flag that is FALSE.
- <u>Platoon rear (candidate) target</u>: this is the target directly in the rear of the host vehicle of which platooning messages (i.e. PCM, PMM) are received. As the platooning trucks generally do not have vehicle sensors observing the rear, this target is only known to the vehicle via V2X communication.

The interfacing of the Platoon Coordinator with the white label truck is schematically shown in Figure 3. Note that the reference design of Task 3.1 consists of the communication (receiver, transmitter modules in Figure 3) and the Platoon Coordinator (i.e. Tactical Layer). The other modules are OEM specific, but must comply with the ENSEMBLE requirements.
In the design of the Platoon Coordinator module the concept is applied that there is only minimum necessary interaction with the Operational Control, allowing operational and tactical decision making to be done in separate layers.

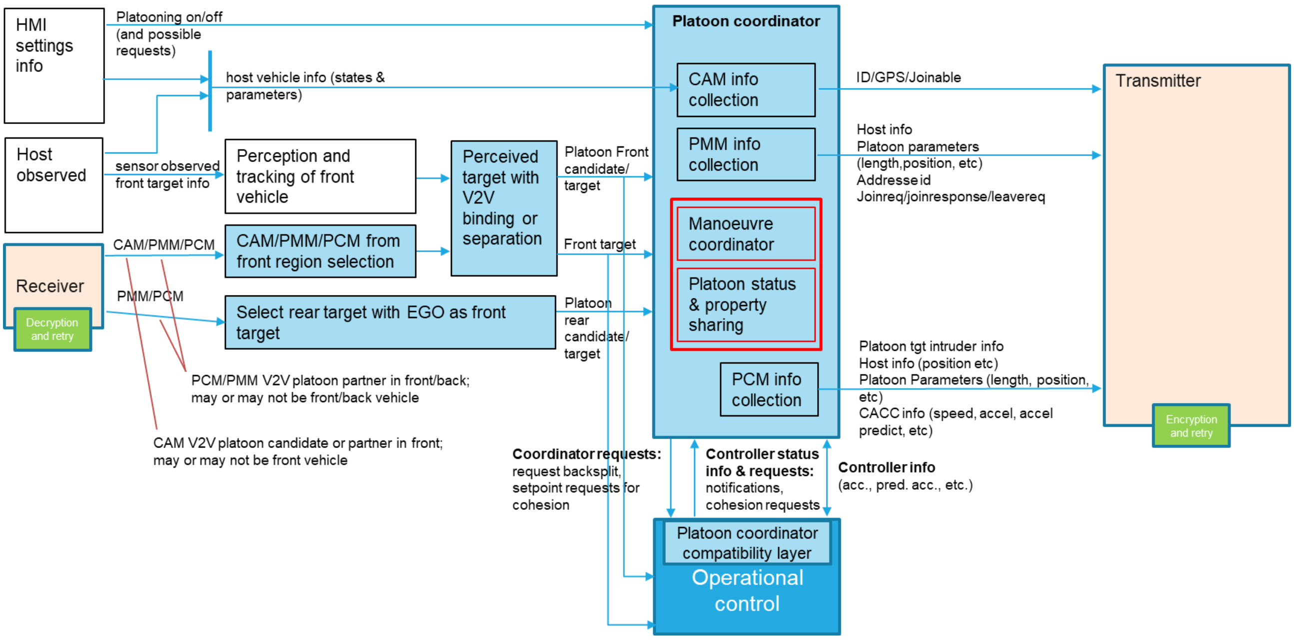


Figure 3: Embedding of the Platoon Coordinator in the white label truck architecture.

The target inputs consist of all information available about the target. Typically this information contains:

- ID of the target (assigned by the world model)
- Pose, velocity and acceleration of the target
- All V2X messages received from the target: CAM, PMM, PCM
- Some additional status information about the target data, which is mainly used for the operational layer.

For the Platoon Coordinator only the target ID, the V2X messages and two status variables are required. The two status variables that have been defined are DATA_MODE and DATA_MODE_V2V. These DATA_MODE variables allow a 'quick' classification of what data is available.

DATA_MODE indicates which sensors are available from which the information is obtained and DATA_MODE_V2V indicates which type of messages are 'continuously' received. The meaning of DATA_MODE and DATA_MODE_V2V is further explained in Tables 1 and 2, respectively.

DATA_MODE	Meaning
0	No sensor and no V2V
1	V2V
2	Radar
3	Radar & V2V
4	Camera
5	Camera & V2V
6	Camera & Radar
7	Camera & Radar & V2V

Table 1: DATA_MODE indicating the availability of sensor and V2V information.

DATA_MODE_V2V	Meaning
0	No CAM and no PCM
1	CAM
2	PCM
3	CAM & PCM

Table 2: DATA_MODE_V2V indicating the availability of CAM and PCM messages.

Some notes:

- DATA_MODE formatting is done such that:
    - even numbers have only vehicle sensors;
    - uneven numbers have V2V.
- Target information is also used in the operational layer of the white label truck implementation and therefore contains more information than is strictly necessary for the Platoon Coordinator.
- Status information about the contents of the target information, like DATA_MODE and DATA_MODE_V2V could also be obtained by analysing the content of the target information inside the Platoon Coordinator, but it was chosen to do this outside in the world model module.

The Platoon Coordinator architecture is further detailed in Figure 4.

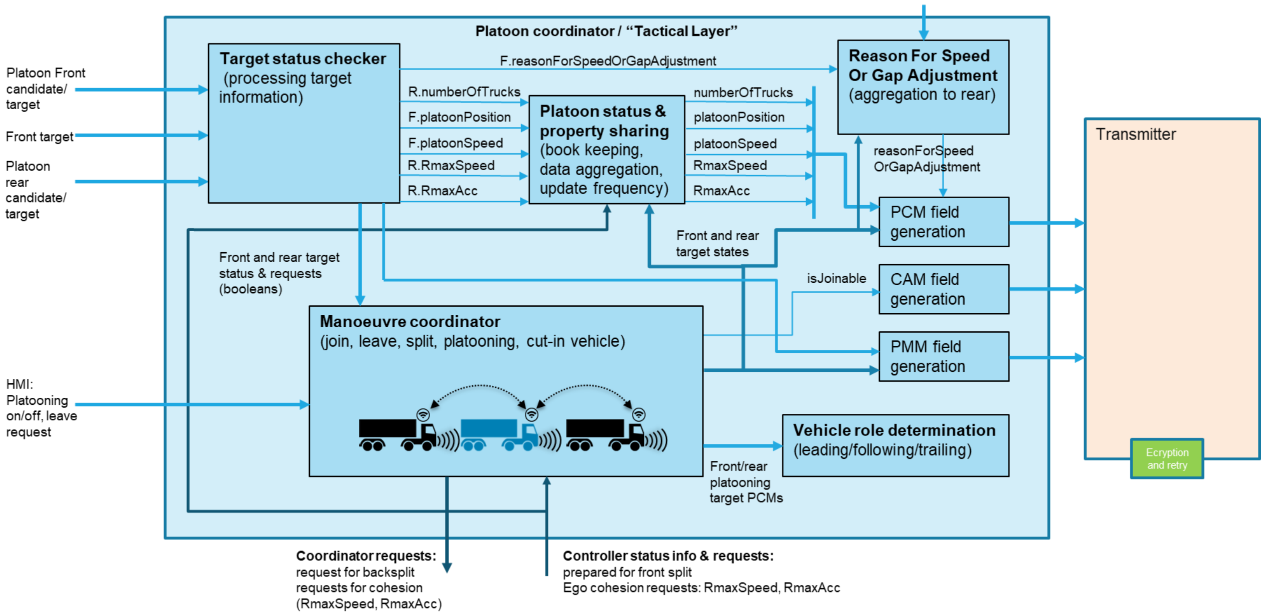


Figure 4: Architecture of the Platoon Coordinator / "Tactical Layer" software module.

The following components are present:

- Manoeuvre coordinator: ensures that the sequence of actions, as defined in the interaction protocol to handle the manoeuvres, is conducted. The block contains finite state machines.
- Platoon status & property sharing: collects and shares platoon status and vehicle property information with the whole platoon - D2.4 specifications: Tactical_Layer_001 to Tactical_Layer_005.
- Reason for speed or gap adjustment: collects and shares the reasons for speed and gap adjustment in the platoon - D2.4 specification: HMI_004.
- Vehicle role determination: determines the role of the host vehicle - D2.4 specification: HMI_008.
- Message field generation blocks: set the correct fields in the CAM, PCM and PMM messages to be sent by the host vehicle.
- Target status checker: processes the target information and extracts fields; this is required to obtain the correct information for the other components.

3.2. Manoeuvre coordinator

As mentioned above, the manoeuvre coordinator is responsible for handling the sequences of the interaction protocol to conduct the manoeuvres to join, leave or split the platoon. The interaction of the trucks is specified in D2.4 and the use cases are described in D2.2. The following use cases are relevant for the manoeuvre coordinator:

- Platooning
- Join from behind
- Leave by leading, follower and trailing truck

- Split by trailing vehicle

In order to handle these manoeuvres, the Manoeuvre coordinator is split in two state machines, one handling the interaction with the rear target and one with the front target, as is depicted in Figure 5. In this way, the interaction logic can be kept relatively simple and also fits the interaction protocol that consists of some elementary sequences for a front split, back split and join. For example:

- Leave by a follower vehicle is a combination of front and back split triggered by the HMI;
- Split is also a combination of front and back split but triggered by the system, such that in the end there are two platoons;
- Leave as leader vehicle is a back split triggered by the HMI;
- Leave as trailing vehicle is a front split triggered by the HMI.

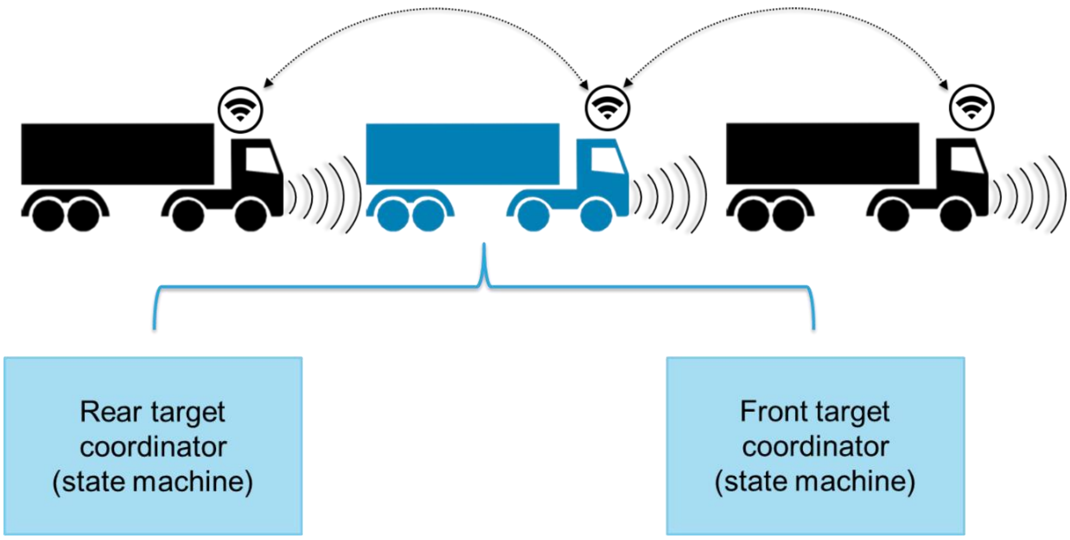


Figure 5: Manoeuvre coordinator consisting of a front and rear target coordinator.

3.2.1. Front target coordinator

The Front target coordinator solely handles the interaction with the vehicle in front, which can either be a platooning (candidate) target or another vehicle. The Front target coordinator is schematically shown in Figure 6.

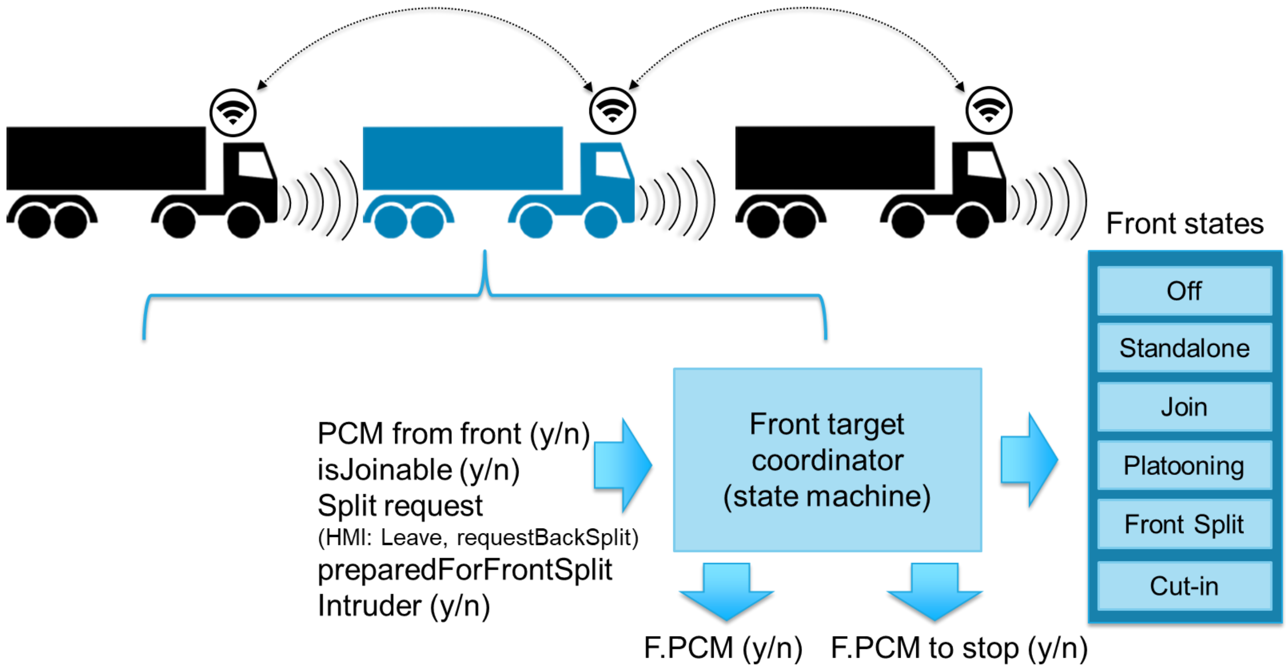


Figure 6: Front target coordinator inputs, states and outputs.

The finite state machine of the Front target coordinator has the following states:

- Off: platooning function is disabled;
- Standalone: platooning function is enabled, but no communication is (yet) setup to receive PCMs of the front target;
- Join: negotiating process to setup PCM communication with the front target;
- Platooning: exchange of PCM communication with the platooning front target is established and PCMs are received;
- Front split: process to establish the standalone distance in order to stop communication with the front platooning target;
- Cut-in: receiving PCMs of front platooning target, which is not equal to the vehicle in front.

The external inputs to the Front coordinator that are used for the state transitions are:

- PCM from front (y/n): Boolean indicating if PCM messages of the front target are received;
- isJoinable (y/n): Boolean indicating if the vehicle in front is joinable for platooning;
- Split request from the platooning front target (y/n): Boolean indicating if the front target wants to split the platoon. This can be the result of the front truck driver triggering a Leave request on his HMI;
- Prepared for front split (y/n): Boolean indicating that the controller has increased the distance to standalone distance;
- Intruder (y/n): Boolean indicating the presence of a cut-in vehicle.

The basic design of the front finite state machine is shown in Figure 7. The states are indicated with the blocks and the arrows represent the state transitions.

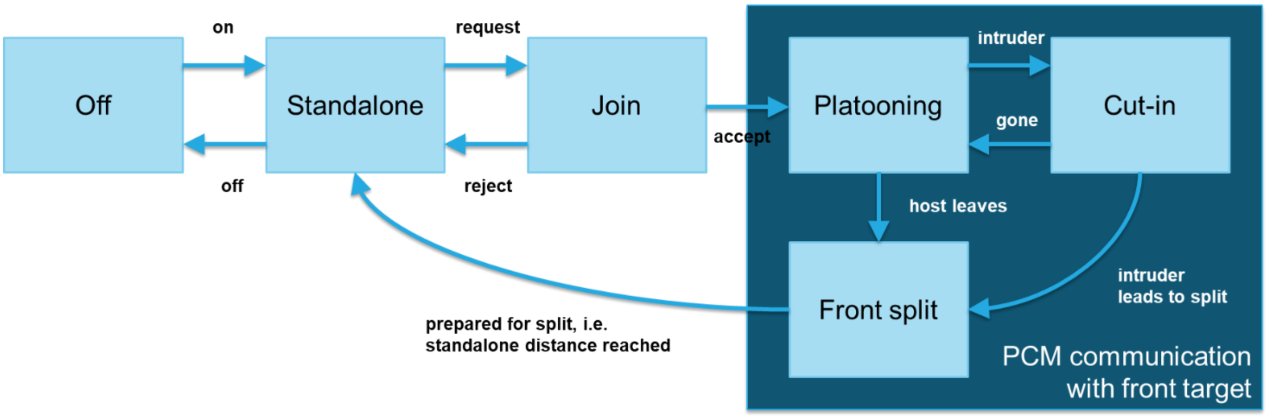


Figure 7: Basic design of the front target finite state machine.

The outputs of the Front coordinator are, besides the states, flags to indicate if PCM exchange with the front target is going to stop (as result of a request for back split from the front) or if PCM exchange is established. The first flag is used to interface with the controller to trigger it to increase the gap to standalone distance. The second flag is used for vehicle role determination. Finally, the situation with an intruder is schematically depicted in Figure 8 to clarify the usage of the 'PCM with front target' flag and the difference between a 'Front target' and 'Platooning front target'.

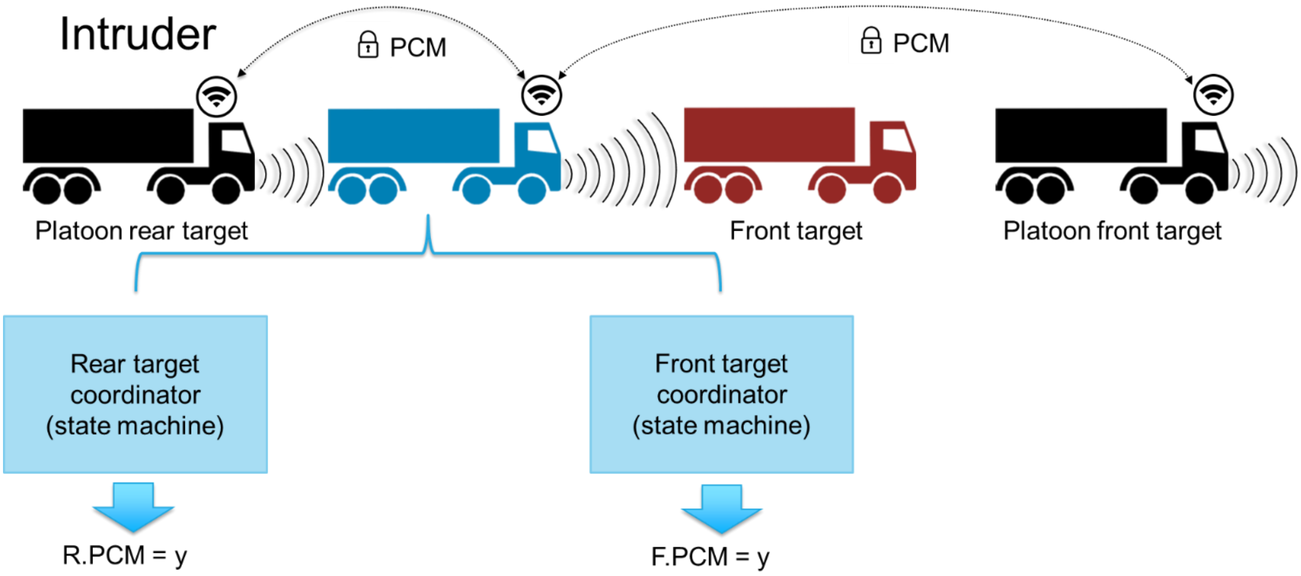


Figure 8: Front target coordinator PCM output flag in case of an intruder (red truck).

3.2.2. Rear target coordinator

The Rear target coordinator solely handles the interaction with the vehicle in the back. The Rear target coordinator is schematically shown in Figure 9.

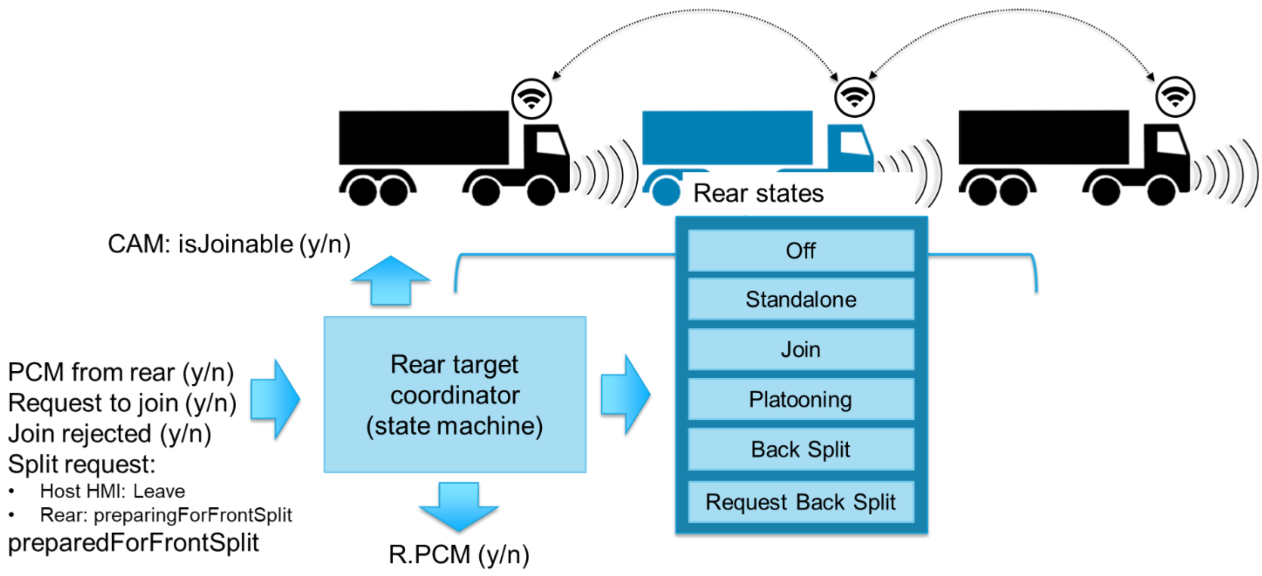


Figure 9: Rear target coordinator inputs, states and outputs.

The finite state machine of the Rear target coordinator has the following states:

- Off: platooning function is disabled;
- Standalone: platooning function is enabled, but no communication is (yet) setup to receive PCMs of the rear target;
- Join: negotiating process to setup PCM communication with the rear target;
- Platooning: exchange of PCM communication with the platooning rear target is established and PCMs are received;
- Back split: process to stop communication with the rear platooning target;
- Request back split: the host vehicle is leaving and needs to send a request to the vehicle behind to increase the gap to the standalone distance and waits for confirmation before a back split can take place.

The external inputs to the Rear coordinator that are used for the state transitions are:

- PCM from rear (y/n): Boolean indicating if PCM messages of the rear target are received;
- Request to join (y/n): Boolean indicating if a join request of the rear target is received;
- Join rejected (y/n): Boolean indicating if the join request is rejected;
- Split request:

  - Host vehicle Leave request (y/n): Boolean indicating if the host vehicle wants to leave the platoon;
  - Rear target preparing for split (y/n): Boolean indicating that the rear target is preparing for a (front) split, i.e. the rear vehicle indicates that it wants to leave the platoon;
- Rear target prepared for (front) split (y/n): Boolean indicating that the rear target reached the standalone distance and will stop PCM communication.

The basic design of the rear finite state machine is shown in Figure 10.

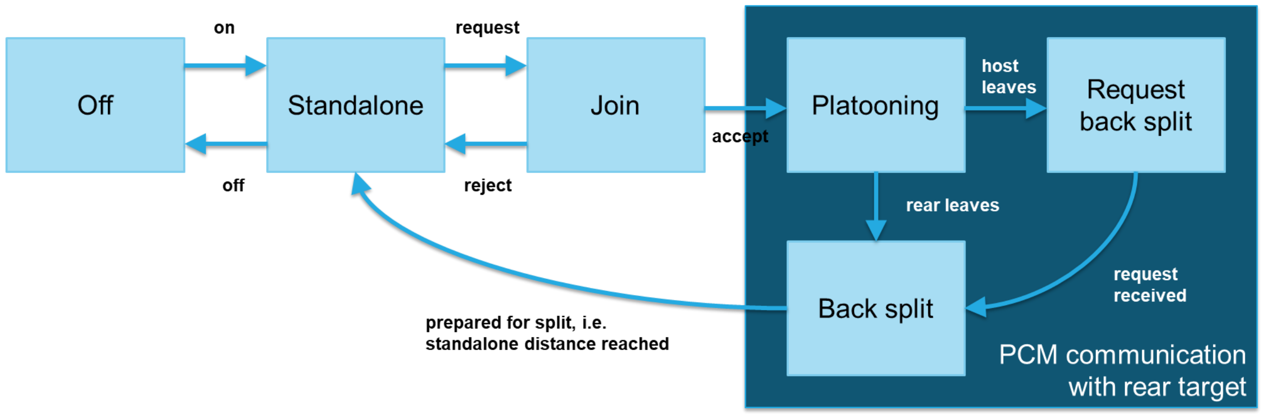


Figure 10: Basic design of the rear target finite state machine.

The outputs of the Rear coordinator are, besides the states, a flag to indicate if PCM exchange with the rear target is established, and a flag to indicate if the host vehicle can be joined (i.e. the isJoinable flag for the CAM message).

3.2.3. Vehicle role determination

Once the states of the front and rear target coordinator are known, the vehicle role in the platoon can be determined based on the availability of exchange of PCMs with the front and rear target (output flags of the front and rear target coordinator) using a Truth Table, see Table 3.

PCMs with front target	PCMs with rear target	Vehicle role
F (n)	F (n)	Standalone
F (n)	T (y)	Leader
T (y)	T (y)	Follower
T (y)	F (n)	Trailing vehicle

Table 3: Truth table for vehicle role determination in the platoon; F = false (n = no), T = true (y = yes).

3.3. Platoon status and property sharing

In D2.4 requirements have been made for information that must be shared with all vehicles in the platoon. Although a single broadcasting domain is considered, the reception of messages of all vehicles in the platoon cannot be guaranteed, in particular if the platoon is long. Therefore, information is forwarded from one truck to the other. Typically the information that is shared with the whole platoon is of tactical nature or for HMI purposes.

The method of data aggregation is applied, where the host vehicle receives information from a neighbouring vehicle, optionally changes this information depending on its own status and forwards this information again to the neighbouring vehicles. The communication topology for data aggregation can be from front to rear, or from rear to front, as is shown in Figure 11.

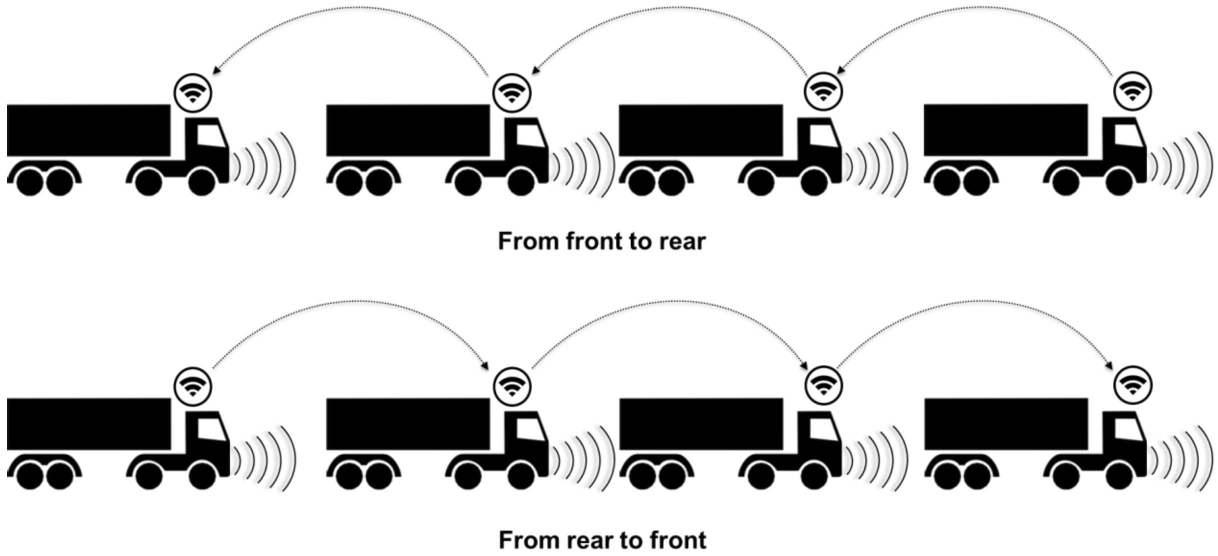


Figure 11: Communication topology for data aggregation.

In the ENSEMBLE communication protocol containers have been defined for data aggregation in the PCM message. These containers are to be updated with a lower rate than the PCM message is sent. For this reason, these containers are OPTIONAL fields in the PCM message. The PCM message is sent every 50 ms, which corresponds to a rate of 20 Hz. The idea of the OPTIONAL field for the containers is that these can be updated at a different frequency between 1 Hz and 20 Hz. The current specifications D2.4 state that the update frequency of the platoon status is initially defined to be 1 Hz, and that the definition of the final value will be the subject of further investigations during the project. Basically, it is said that the lowest update frequency will be first tried, and if this might turn out to be insufficient, the update frequency might be increased for the final specifications. Finally, for the sake of clarity the mechanism of sending the containers is further explained below:

- Send PCM message that includes the OPTIONAL container;
- After that send 19 PCM messages that do not include the OPTIONAL container;

- Next send PCM message that includes the OPTIONAL container;
- Then send 19 PCM messages that do not include the OPTIONAL container.
- Etc.

If a certain field in a container is forwarded from front to rear, or vice versa, and possibly updated before forwarding, is set by some rules specific for that field. In the sections below, the requirements for the containers and the rules for the different fields are discussed.

3.3.1. Platoon status and sharing

The following requirements from D2.4 apply to the platoon status and sharing:

- Tactical_Layer_001: The platoon system over the tactical layer will gather platoon status and data information (Number of trucks in the platoon, Ego-truck's position in the platoon, Cut-in vehicle in the platoon, Platoon set speed and Platoon leader vehicle actual speed) and distribute this information over the platoon.
- Tactical_layer_002: The platoon system status information gathered by the tactical layer is updated cyclically. Since this information is not time critical, the update frequency can be chosen substantially lower compared to control related V2V containers.
- Tactical_layer_003: The platoon system status information within the tactical layer is shared between the trucks.
- Long_Control_008: When the intention is to increase the time gap to the preceding vehicle in the platoon, the relative speed compared to the lead vehicle shall be maximum 3 km/h and the maximum deceleration shall be 3 m/s². The requirement on relative speed does not apply to look ahead functionality (that for example is increasing the time gap before a downhill in order to use a higher rolling speed to close the gap again).
- HMI_004: The driver in a platoon shall be informed about the reasons to speed and gap adjustments.

From these, it is concluded that the following platoon information must be shared between the trucks in the platoon:

- Number of trucks in the platoon
- Ego-truck's position in the platoon
- Platoon speed (i.e. speed of the lead vehicle)
- Reason for speed and gap adjustment (e.g. due to intruder in the platoon)

In the ASN.1 definition of the ENSEMBLE communication protocol, a PlatoonStatusSharingContainer has been defined that contains the following fields:

- numberOfTrucks
- platoonSpeed
- platoonPosition
- reasonForSpeedOrGapAdjustment

For the reasonForSpeedOrGapAdjustment, the following options have been defined:

- safety: each vehicle determines its own 'safe gap'; if circumstances change and the gap needs to be enlarged or decreased speed changes occur;
- efficiency: speed is reduced for efficiency reasons, e.g. rolling out on a hilly road. This is especially relevant if the first truck uses a predictive ACC;
- trafficAhead: the platoon speed may be lowered by the lead vehicle, because of slow driving traffic ahead;
- intruder: when an intruder is in the platoon, gaps need to be opened and closed leading to speed variations;
- emergency: triggered in case hard braking is detected (emergency braking use case); note: not to be used for control but only for displaying to the driver;
- leave: leaving of a following or lead truck leads to increase and decrease of gaps and consequently speed variations;
- cohesion: speed is lowered to allow other vehicles in the platoon to catch up.

The rules for data aggregation of the different fields are listed below:

- The number of truck in the platoon is aggregated / forwarded from the rear to the front. This means the number of trucks value of the ego vehicle is set to the received / communicated number of trucks value from the vehicle behind. Exception: if the ego vehicle is the trailing vehicle, the number of trucks will be set to its platoon position.
- The platoon speed is aggregated / forwarded from front to rear. This means that the platoon speed of the ego vehicle will be set to the received / communicated platoon speed of the vehicle in front. Exception: if the ego vehicle is the platoon leader, the platoon speed is set to the ego vehicle speed.
- The platoon position is the ego vehicle's position in the platoon:
    - When joining a platoon, the platoon position is obtained in the JoinResponsInfo (joiningAtPosition field) of the communication protocol;
    - During platooning, the platoon position value is regularly checked against the reported platoon position of the preceding truck in the platoon. This means the platoon position value of the ego vehicle is updated if the vehicle in front reports a different value than the ego platoon position minus one. In this way, a leave or merge of platoons is accounted for;

  - When becoming the new leader of the platoon, the platoon position is set to 1.
- The reason for speed or gap adjustment is aggregated / forwarded from front to rear. The field is only provided if a speed or gap adjustment occurs. Therefore this field is OPTIONAL in the protocol.
  Note: the protocol only allows to forward a single reason. In case the host vehicle needs to adjust its speed for another reason than reported by the front vehicle, the reason of the host vehicle is forwarded to the rear.

3.3.2. Vehicle property collection and sharing

The following requirements from D2.4 apply to the vehicle property collection and sharing:

- Tactical_layer_004: The platoon system over the tactical layer shares the vehicle property information (Maximum acceleration request (to the platoon), Desired maximum platoon speed), in an equal method within the platoon as the platoon status information.
- Tactical_layer_005: The platoon system property information gathered by the tactical layer is updated cyclically. Since this information is not time critical, the update frequency can be chosen substantially lower compared to control related V2V containers.

From these requirements it is concluded that the following information must be shared with the vehicles in the platoon:

- Maximum acceleration request (to the platoon)
- Desired maximum platoon speed

This information is required for correct functioning of the Cohesion functionality.

In the ENSEMBLE protocol a tacticalPlanning container has been defined in the PlatoonControl.asn (i.e. PCM message). The tacticalPlanning container is part of the PlatoonControlContainer. Currently the tacticalPlanning container only contains the cohesionContainer; the idea is that the tacticalPlanningContainer can be extended in the future. The tactical planning container is (at this time) forwarded/aggregated from the rear to the front (basically because this is required for the cohesionContainer). The naming of the fields in the cohesionContainer read:

- requestedMaxLongitudinalAcceleration
- requestedMaxSpeed

Note that the requestedMaxSpeed is used for two reasons:

- Default: inform the platoon about the maximum speed of the vehicle under actual conditions; mainly the platoon leader vehicle can decide to account for this value by not driving at higher speeds than the maximum speed of any vehicle in the platoon.

- Optional: temporarily request a lower maximum speed than the physical maximum speed to be able to solve an existing cohesion problem. Example: platooning vehicles have all the same maximum speed due to e.g. a speed limiter. After a cut through, a gap exists that needs to be closed. This can only be done if the vehicles in front of the cut-in reduce speed.

Rules for data aggregation of the cohesion functionality:

- Maximum acceleration request (to the platoon): The host vehicle compares the requested maximum acceleration from the vehicle behind with its own desired maximum acceleration and forwards the minimum of the two acceleration values to the vehicle in front as maximum acceleration request.
- Desired maximum platoon speed: The host vehicle compares the requested maximum speed from the vehicle behind with its own desired maximum speed and forwards the minimum of the two speeds to the vehicle in front as desired maximum platoon speed.

Finally, Figure 12 shows a detail of the Architecture of the Platoon Coordinator of Figure 4. The detail depicts the principle of the data aggregation by showing that fields received from the front (F.) or rear (R.) target are used to set the corresponding host vehicle fields. In case of the cohesion functionality, the received coordinator requests are forwarded to the Longitudinal control. The Longitudinal control also provides the cohesion requests of the host vehicle, such that these can be accounted for before setting the fields for the V2V messages.

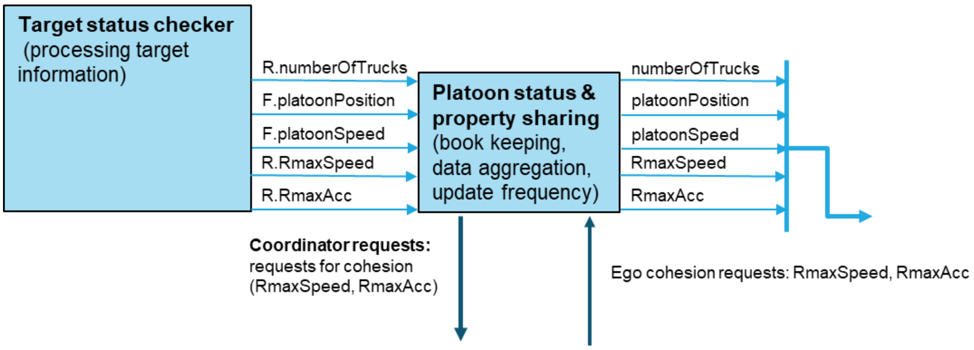


Figure 12: Detail of the architecture of the Platoon Coordinator, showing the interface with the Longitudinal Control for Cohesion requests.

4. SOFTWARE IMPLEMENTATION

The design of the Platoon Coordinator or “Tactical Layer” modules is described in Chapter 3. In this chapter the implementation of the design in software is described. The Simulink programming environment of the MathWorks is used to develop the Platoon Coordinator Simulink model. This model is set up such that it can interface with the white label truck model and the HIL facility (Schmeitz, 2019). Before the model can be embedded in the HIL facility it is compiled using Simulink Coder. The (readable) source code, i.e. Simulink model, of the Platoon Coordinator is available as confidential deliverable for members of the consortium (including the Commission Services).

Simulink was chosen as programming language, because it is a graphical programming tool widely used in the Automotive industry. Several add-on tools are available for Simulink to support the modelling and finally the embedding of the software in rapid control prototypes. For the implementation of the Platoon Coordinator, the Stateflow tool is used, as it is a graphical language that extends Simulink with a design environment for developing state machines and flow charts. The Simulink model of the Platoon Coordinator is created in MATLAB R2017b.

The aim of this chapter is to guide the user of the Platoon Coordinator model through the Simulink model and to make connections with the design described in Chapter 3. Note that the sections of both chapters are aligned to facilitate the comparison. Finally, it is recommended to read this chapter and simultaneously discover the Simulink model in MATLAB.

4.1. Platoon Coordinator overview & interfacing

The top level diagram of the Platoon Coordinator model is depicted in Figure 13. In the (purple) boxes on the left and right, the inputs and outputs of the model are shown. In the middle part of the diagram the following sub-systems exist:

- Target status checker (middle: top left);
- Manoeuvre coordinator (middle: bottom);
- Platoon status & property sharing (middle: top right).

When comparing Figure 13 with Figure 4, the architecture of the design can be clearly recognised in the top level diagram.

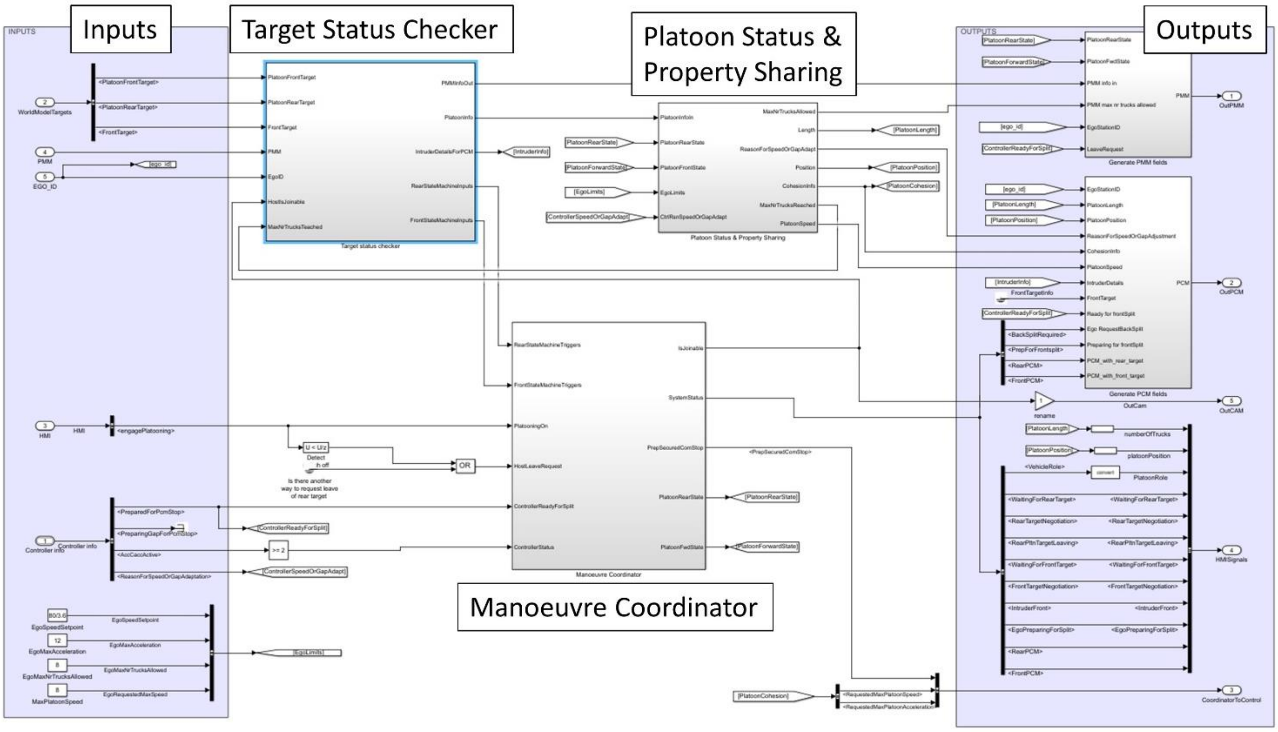


Figure 13: Simulink model of the Platoon Coordinator.

4.2. Manoeuvre coordinator

The Manoeuvre coordinator subsystem is depicted in Figure 14. The main parts of this subsystem are the Rear and Front target coordinator Stateflow charts. The subsystem on the right is used to obtain the system status and contains the vehicle role determination Truth Table (Table 3Table 3).

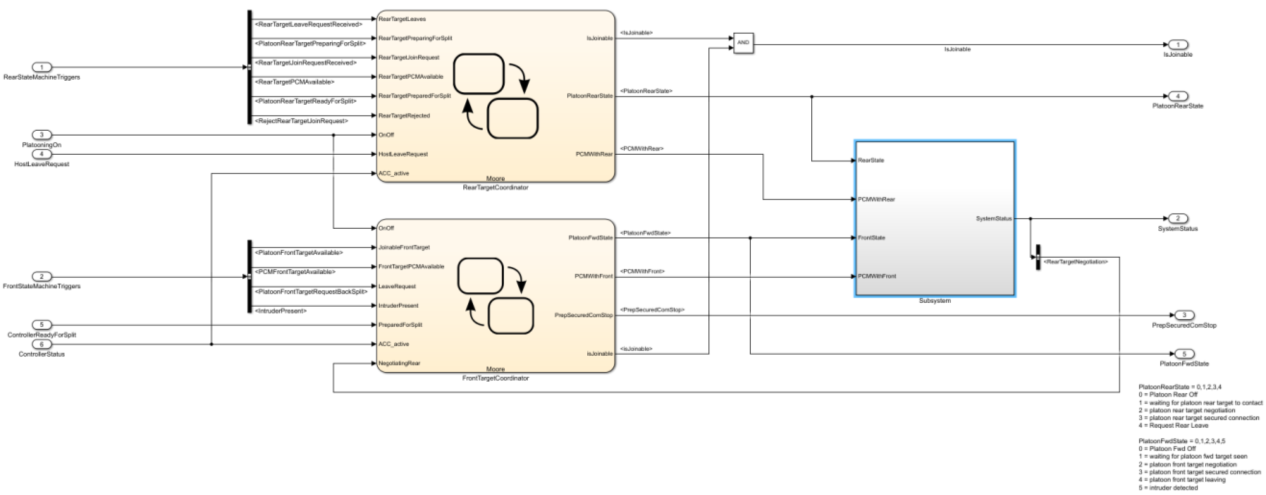


Figure 14: Manoeuvre coordinator subsystem.

4.2.1. Front target coordinator

The front target coordinator Stateflow diagram is shown in Figure 15, where it is also compared with the design of Figure 7.

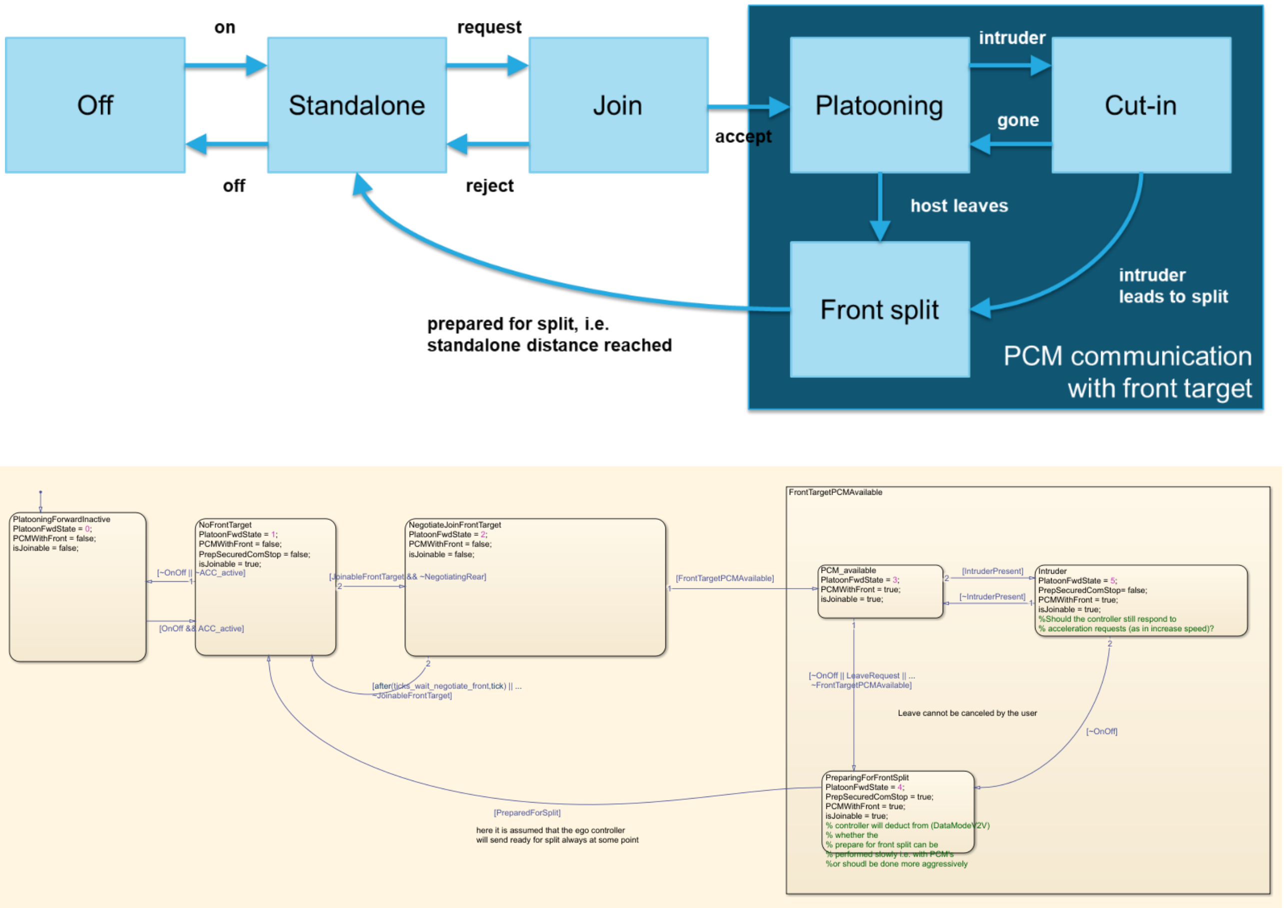


Figure 15: Front target coordinator Stateflow diagram and design.

Note that in the implementation some additional conditions for the state transitions are added. These conditions handle the loss of communication and the turning off of the platooning system. The transition from intruder to front split is handled by the intruder checking, i.e. intruderPresent flag. This flag will become false if e.g. the platoon is split as result of communication loss (out-of-range).

Finally, note that for the implementation of the Platooning function the variant is chosen that the Platooning function is an option of ACC, meaning that ACC must be active before the Platooning function can be enabled.

4.2.2. Rear target coordinator

The rear target coordinator Stateflow diagram is shown in Figure 16, where it is also compared with the design of Figure 10.

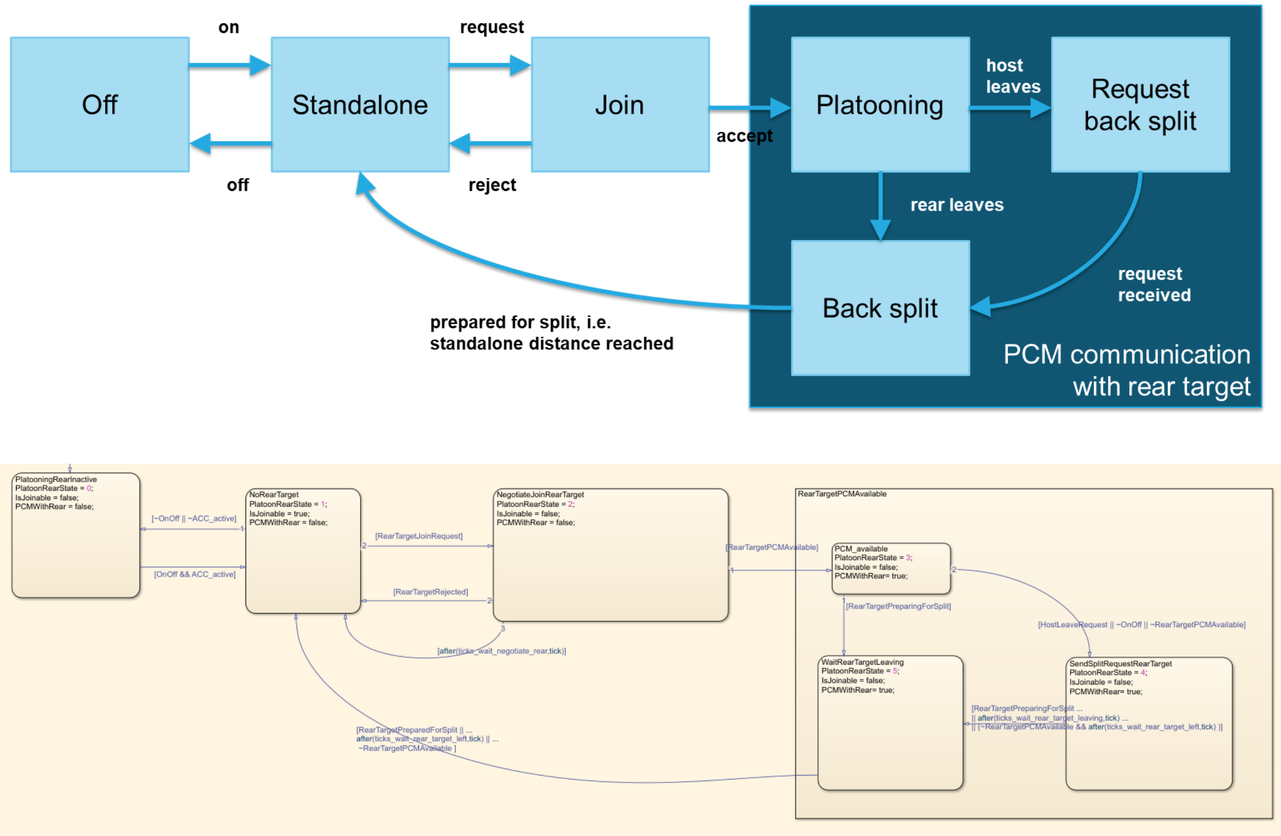
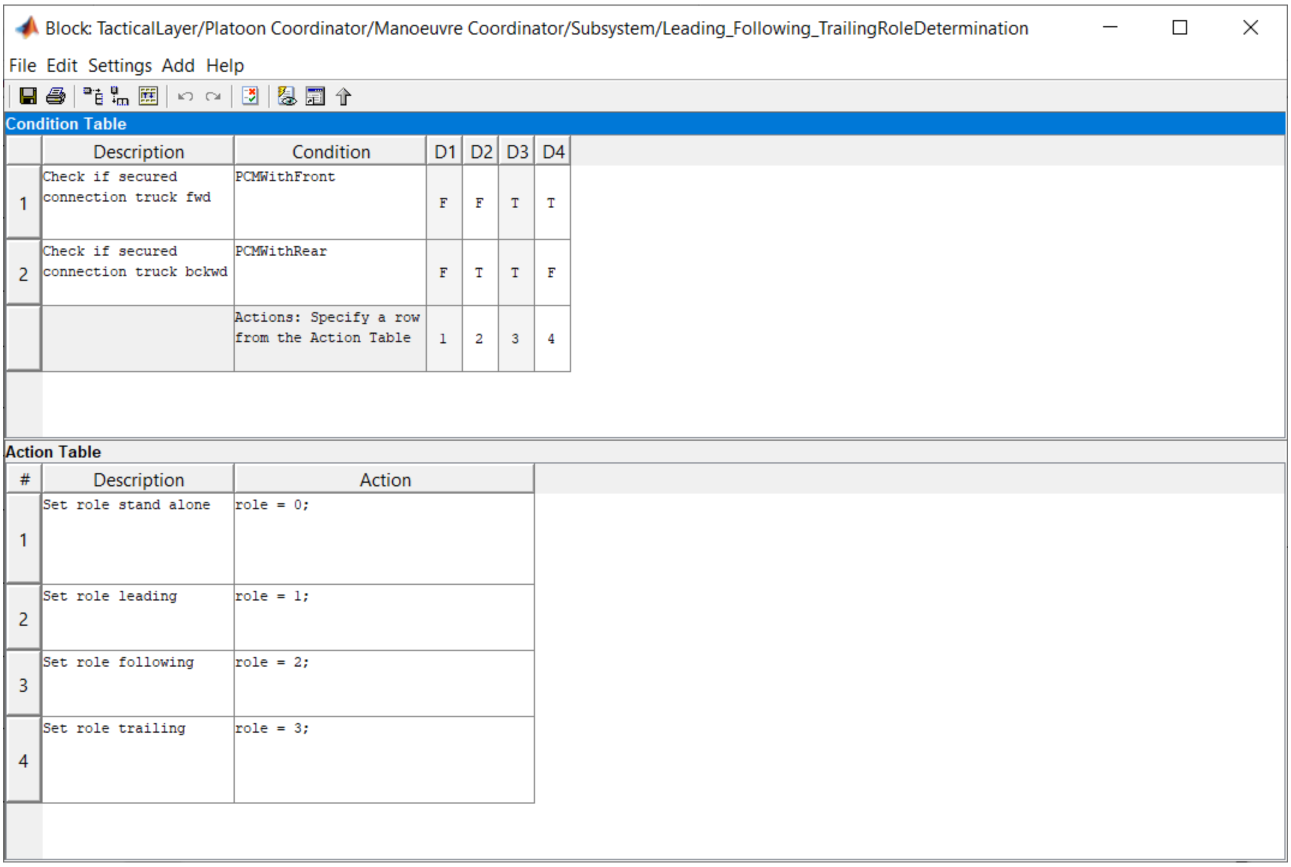


Figure 16: Rear target coordinator Stateflow diagram and design.

Note that, like before in the front target coordinator, in the implementation some additional conditions are added for the state transitions. These conditions handle the loss of communication and the turning off of the platooning system. Note that in the implementation also a request for back split state is made if PCM communication with the rear target is lost. The reason to do so, is that although the host vehicle does not receive PCMs of the rear target anymore, the rear target might still be able to receive PCMs of the host vehicle. Furthermore, several time outs have been added to the state transitions to avoid deadlocks in states if communication fails.

4.2.3. System status determination

In the Simulink model the system status determination subsystem collects the front and rear target coordinator states and performs the vehicle role determination. The role determination is implemented in a Stateflow Truth Table block, as is shown in Figure 17.


Block: TacticalLayer/Platoon Coordinator/Manoeuvre Coordinator/Subsystem/Leading_Following_TrailingRoleDetermination

File Edit Settings Add Help

Condition Table

	Description	Condition	D1	D2	D3	D4
1	Check if secured connection truck fwd	PCMWithFront	F	F	T	T
2	Check if secured connection truck bckwd	PCMWithRear	F	T	T	F
		Actions: Specify a row from the Action Table	1	2	3	4

Action Table

#	Description	Action
1	Set role stand alone	role = 0;
2	Set role leading	role = 1;
3	Set role following	role = 2;
4	Set role trailing	role = 3;

Figure 17: Stateflow Truth Table for vehicle role determination.

4.3. Target status checker

The Target status checker subsystem is depicted in Figure 18. The Front target handler and the Rear target message handler extract the Boolean parameters that are required as input for the Manoeuvre coordinator. Additionally the fields required for the input of the Platoon status & property sharing block are extracted from the available information of the platoon front and rear targets.

The upper subsystem in Figure 18 is used for cut-in / intruder detection. An intruder is detected using the following conditions:

- ID of the platoon front target and front target do not match;
- PCMs are received from the platoon front target:
    - V2V is available & V2V messages contain (decrypted) PCMs.

Apart from the intruder detection, the block also collects the distance and speed to the vehicle ahead that are to be sent in the PCM. Note that the PCM contains two fields: intruderAhead and vehicleAhead that are mutually exclusive:

- intruderAhead: an intruder detected in front of the host vehicle;

- vehicleAhead: any vehicle in front of the host vehicle, i.e. a platooning vehicle but also a normal target if the host vehicle is the leader, but no intruder.

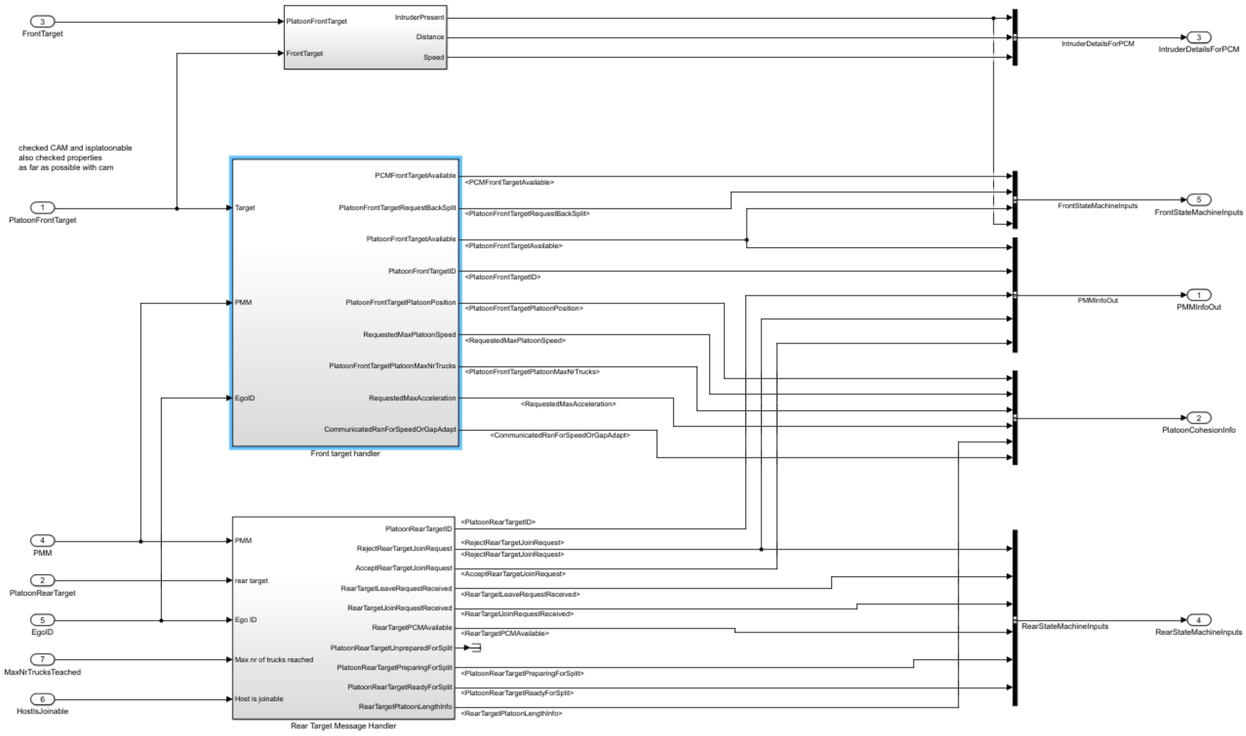


Figure 18: Simulink diagram of the Target status checker.

In the Front target handler and the Rear target message handler, some functionality is available to deal with temporary communication losses. The following principles are applied taking into account the 150 ms of communication loss that is accepted in the defined Watchdog sequence diagram of the ENSEMBLE communication protocol:

- When receiving CAM messages (with isJoinable flag is true) of the front target, reception should be obtained for 150 ms, before the PlatoonFrontTargetAvailable flag is set to true.
- Once CAM and PCM messages are received the PlatoonFrontTargetAvailable, PCMFrontTargetAvailable and RearTargetPCMAvaialble flags are kept true for 150 ms even if these turn to false due to temporarily communication loss within these 150 ms.
- As a leader immediately send PCMs to the rear target after an accepted join response by the host vehicle for at least 5 s. This means the RearTargetPCMAvailable flag is immediately set to true without actually receiving PCMs of the rear target. In this way, the rear target is given 5 s of time to start sending PCMs.

4.4. Platoon status and property sharing

The Simulink diagram of the Platoon status & property sharing block is depicted in Figure 19. The diagram shows the logic for the data aggregation.

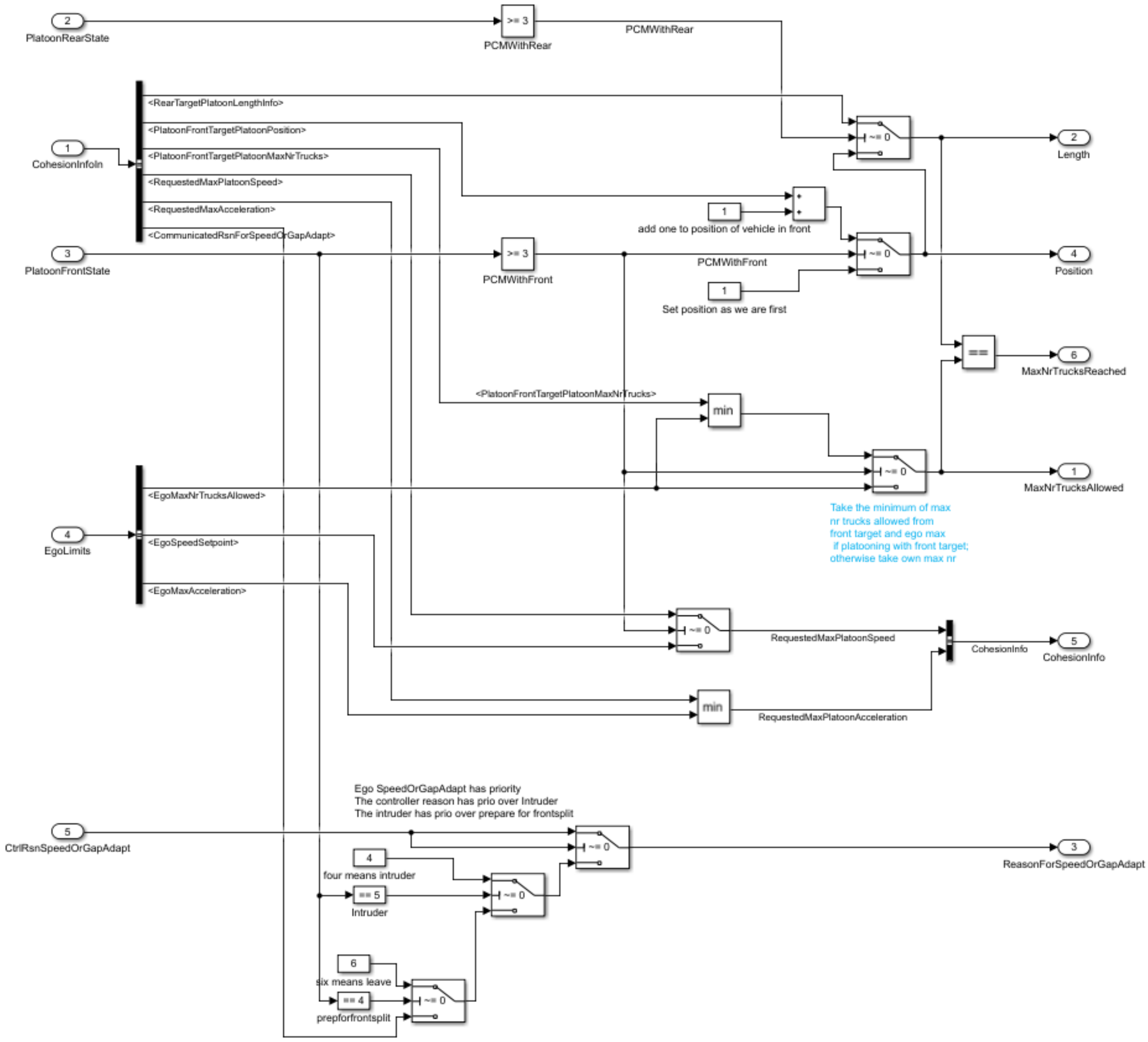


Figure 19: Simulink diagram of the Platoon status & property sharing block.

5. SOTWARE TESTING

In this Chapter, the software tools that can be used to test the Platoon Coordinator are described. It is recommended, to read this chapter with the actual test tools at hand. “Platoon Coordinator” here refers to the Simulink reference model shown in Figure 20.

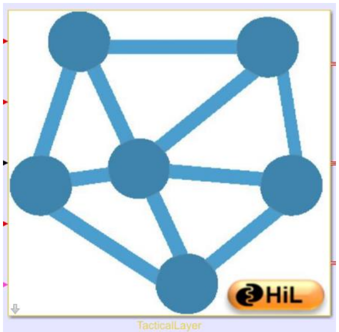


Figure 20: Simulink Block containing the "Platoon Coordinator" software programmed in Matlab/Simulink 2017b.

The following Matlab/Simulink files are provided in the Software folder of the Tactical Layer:

1. **TacticalLayer.slx**: is a Simulink Reference Model, that implements the Platoon Coordinator.
2. **TLBusses.mat***:* is a .mat structure containing the bus definitions for the Tactical Layer. Besides this, the interfaces from the operational controllers, perception modules, communication and human machine interface are also included.
3. **CoordinatorParameters.m***:* is a Matlab script which defines a Simulink Parameter structure used by the TacticalLayer.slx model.
4. **Unit_test_signals.m** : is a Matlab class, which defines a Unit Test Signals (UTS) structure whose fields can be used to generate signals that trigger the platoon coordinator for testing purposes.
5. **unit_test_platoon_coordinator_scenario.m**: is a Matlab script that can be used to create a Unit_test_signals object (for testing of a certain scenario), run a simulation model containing the Tactical Layer, and save the data produced by the simulation.
6. **unit_test_platoon_coordinator.slx:** Simulink Model containing a model of TacticalLayer.slx (reference), look-up tables that use the signals defined in the unit_test_platoon_coordinator_scenario.m, and emulators of front/rear targets which can be enabled or not, depending on the desire to specify more or less signals to test the Platoon Coordinator.
7. **TacticalLayerTestResultPlotting.mlx**: is a Matlab live-script that can be used to visualize data generated by unit tests. It also contains a table regarding the coding of the platoon role of a truck in a platoon, which may be useful for a quick check on the results produced by the Platoon Coordinator.

8. **UnitTestTacticalLayerData**: is a folder containing .mat structures. Each structure contains a variable “TacticalLayerLog”, that comes from the simulation of the Platoon Coordinator. The TacticalLayerLog contains *TacticalLayerOut* and *TacticalLayerIn*, which are structures containing time-series of all the inputs and outputs of the Platoon Coordinator.

In the rest of this chapter, instructions on how to use the testing files are provided(points 4 to 7 above), and explanations are given of some the test results implemented therein.

5.1. Unit test signals

In order to test the Platoon Coordinator input signals need to be designed. The Matlab class named *Unit_test_signals*, in the file *Unit_test_signals.m,* contains properties that represent the major signals that can be given as input to the Platoon Coordinator*.* More signals can be added by expanding the class properties and methods.

Each signal is defined by the signal time span and signal value over the indicated time span. Two type of signals are specified within the class:

- **Step / constant signals**: for example, a signal representing the driver input to enable/disable the platoon control functionality. It is indicated with a signal name for the values, e.g. `platoon_ctrl_on` , and a signal for the time-span over which the signal is defined in the format t_<signal_name>, e.g. `t_platoon_control_on`.
- **Pulse signals:** for example, a signal representing a platoon front target asking to the platoon follower to leave the platoon. It is indicated with a signal name for the values, a signal time-span, and also a pulse duration indicated as <signal_name>_pulse_time, e.g. `ftgt_request_backsplit_pulse_time`.

In order to create an object of the Unit_test_signals class, the following syntax is used, according to the class definition:

```
UTS = Unit_test_signals(<start_time>,<end_time>);
```

UTS will be an object from the Unit_test_signals class, with duration of signals of end_time-start_time [s] and default values for the signals.

Each property of the class Unit_test_signals, is set to a default value by the class constructor, which is executed at the moment that a class object is created. For each property of the class, a method is defined to set:

- Step-time and value in the case of a Step signal;
- Step-time and pulse duration in case of a Pulse signal.

For example, to indicate that platoon_ctrl_on goes to one at time t = 2 s, assuming that a UTS object has been instantiated, the following method can be used:

```
UTS.add_platoon_control_on(2,1);
```

The list of methods applicable to a Unit_test_signals object can be visualized opening the *Unit_test_signals.m* file, or using the . (<dot>) operator on an object instantiated from the Unit_test_signals class.

An example of the possible usage of the Unit_test_signals class to create a scenario for testing of the Platoon Coordinator is available in the file *unit_test_platoon_coordinator_scenario.m*, which will be described further in the next section.

Signals defined within a `Unit_test_signals` object are imported in the unit_test_platoon_coordinator.slx Model, by using look-up tables, as shown in Figure 21.

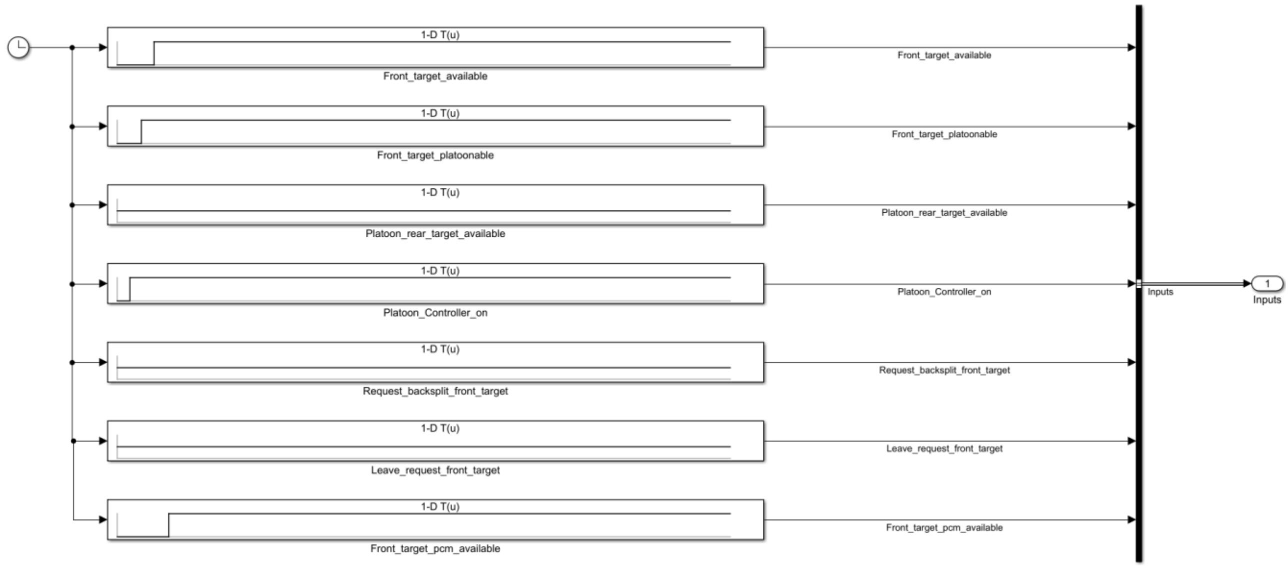


Figure 21: Example of signals sets imported in Simulink for Unit Testing of the Platoon Coordinator. The signals are imported by using look-up tables that use properties of an object from the Unit_test_signals class.

5.2. Scenario Generation for Unit Testing of the Platoon Coordinator

The scenario generation file *unit_test_platoon_coordinator_scenario.m* allows the user to define a sequence of signals to trigger the Platoon Coordinator module under test. Furthermore, it is possible to use emulators for the responses of front/rear targets.

The file contains some scenarios defined according to the document (Atanassow, 2019b).

By selecting the constant named SCENARIO (from 1 to 5) at the top of the script, it is possible to generate the unit test signals, execute the unit test and save the corresponding data for 5 pre-defined scenarios. These pre-defined scenarios do not make use of the emulators, which can be explored by looking into the unit test model.

Below, the usage of the aforementioned script is described and the corresponding test results are discussed through an example:

The first scenario (SCENARIO = 1;) is meant to test the platoon coordinator when the ego truck wants to start Platooning with another truck in front. There is no other truck behind, so we expect the Platoon role of the vehicle to go to *trailing* (which is coded as PlatoonRole = 3).

The test signals to set-up in order to test this scenario can be figured out from the Sequence Diagram specified in the aforementioned ENSEMBLE technical specification document (Join sequence diagram, p.6).

The code to generate the scenario, run the test and save the data is shown in Figure 22.

```
49 -        use_emulator = false;
50
51          % Create Object of class Unit_test_signals, uses tstart,tend as inputs to construct the object
52 -        UTS = Unit_test_signals(0,50);
53
54          % Platooning functionality enabled for the ego-vehicle
55          % "Pre-condition" for the Join sequence diagram
56 -        UTS.add_platoon_control_on(1,1);
57
58          % Front Target is platoonable - Broadcast CAMs (isJoinable = true)
59 -        UTS.add_ftgt_platoonable(2,1);  % t = 2 [s]
60
61          % Front Target is available at t = 3 [s] (detected by on-board sensors)
62 -        UTS.add_ftgt_available(3,1);
63
64          % Join Request is produced automatically by the TacticalLayer
65
66          % Front target sends PCM messages (for current implementation of
67          % Tactical Layer, this is also a JoinResponse = true)
68
69 -        UTS.add_ftgt_join_response(4,1);
70
71 -        UTS.add_ftgt_pcm_available(4.2,1); % we simulate a delay of 200ms
72          % for the front truck to start sending PCM messages
73
74          % Test Tactical Layer
75 -        sim('unit_test_platoon_coordinator.slx',50);
76
77          % Save data
78
79 -        save_data(SCENARIO,data_directory_path,TacticalLayerLog);
```

Figure 22: Scenario Generation, Testing and Data storing for Join manoeuvre (follower truck perspective). No other follower vehicle is present (two truck platooning).

From Figure 22, it can be seen that the ego vehicle sets platooning functionality on (line 56), then it is simulated that a Front Target is available (line 59), and therefore CAM messages are transmitted.

A join request from the ego-vehicle perspective is generated automatically as soon as the Front Target is detected by the on-board sensors of the ego-vehicle and the CAM message is received (line 62). At line 69 it is simulated that the Front target sends a positive Join Response. Finally, at line 71, the reception of PCM messages from the ego-vehicle to the front vehicle is triggered. Now it is expected that the ego-vehicle is going to Platooning, and its role will become “Trailer” (PlatoonRole = 3).

Data are stored in the folder UnitTestTacticalLayerData. The data are saved with a name corresponding to the scenario number. For example, for the Join scenario above, we save the data with the name: *TacticalLayerTest_Scenario1.mat*.

The .mat structure contains the logout of the Simulink file for unit testing, which is named TacticalLayerLog. This variable is loaded in the Base Workspace of Matlab as soon as the simulation is terminated, or the .mat structure saved in the folder can be loaded at a later time instant.

Loading of a certain data structure can be done with the TacticalLayerTestResultPlotting.mlx file. This file is a Live Script, which also contains the platooning role table, to remind the used coding for the platooning role, and quickly check if, under a certain scenario, the PlatoonCoordinator is reacting as expected. For example, after the Join request scenario above, TacticalLayerTestResultPlotting.mlx can be run, and it visualises the platooning Role as is shown in Figure 23.

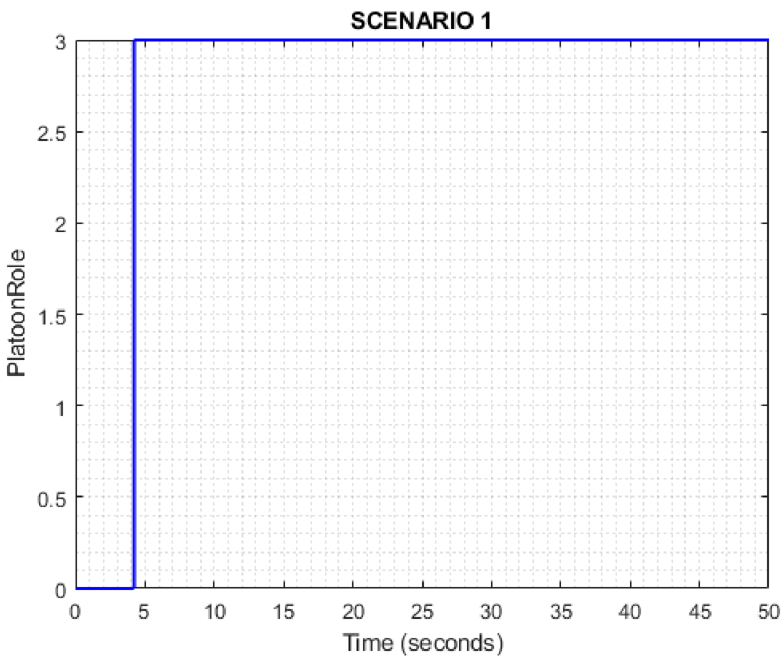


Figure 23: Platoon Role of the ego-vehicle performing a Join scenario (case 1).

At t = 4 s the front target sends a True JoinResponse, and therefore PCM communication can be established and the Platoon is formed. The ego-vehicle is for the trailing vehicle of the platoon, and this is coded with PlatoonRole = 3. Therefore, it can be concluded that a Join scenario is performed correctly, at least from the Role assignment perspective.

A more complete assessment of the Platoon Coordinator can also be made by looking at the sequence of state activation generated by a certain scenario. This sequence should overlap with the specification.

For example, from the ENSEMBLE Technical specification (Atanassow, 2019b), the Join Sequence Diagram is shown in Figure 24.

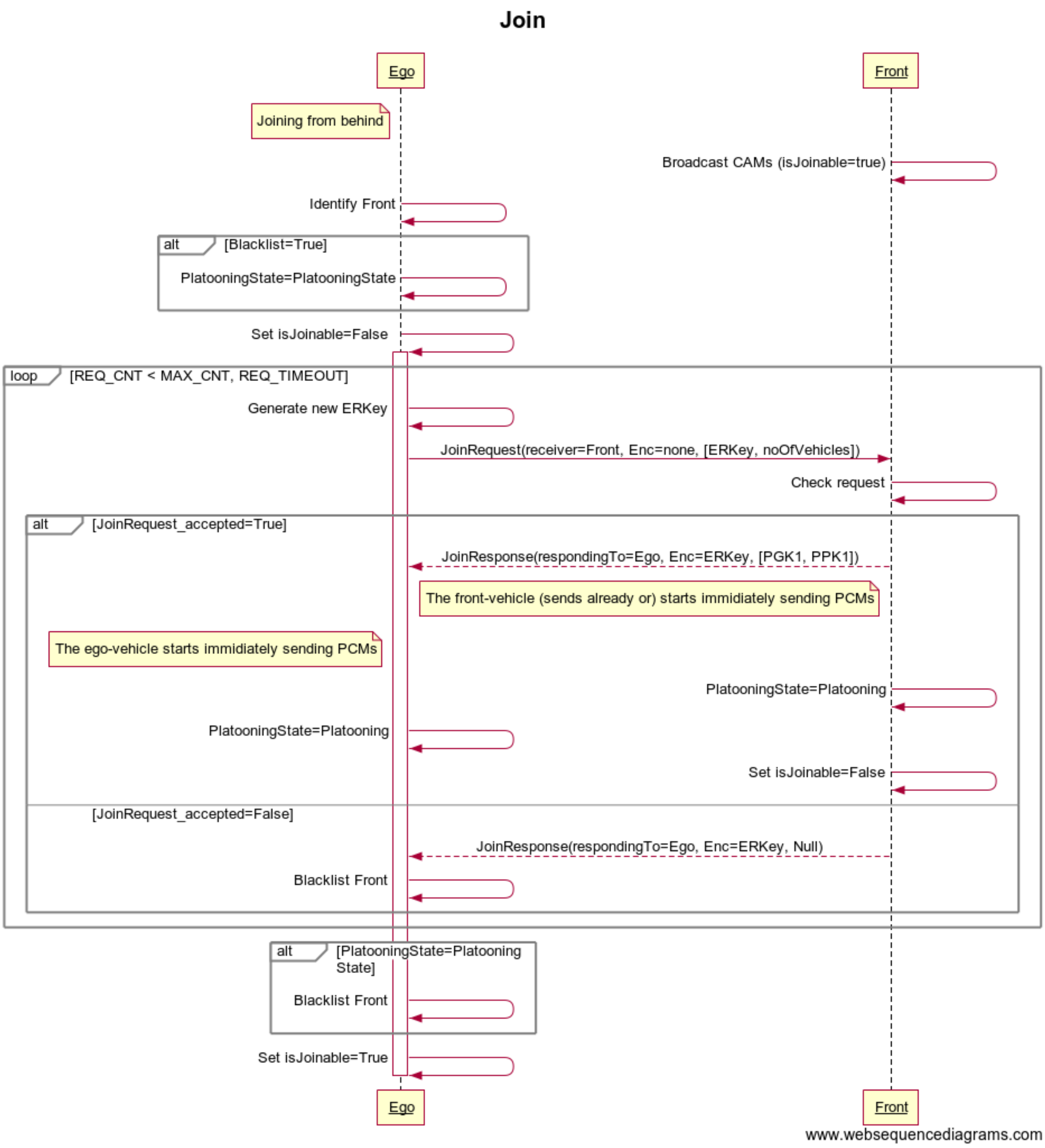


Figure 24: Join Platooning sequence as specified in the ENSEMBLE technical specifications. Both Ego and Front Vehicle perspective are in the sequence diagram.

The unit test model **unit_test_platoon_coordinator.slx** contains a sequence diagram viewer block, which allows to visualize the sequence diagram realised by the platoon coordinator triggered by a certain scenario, and also, for the lower level controllers (like ACC and CACC) by a Control switches emulator, i.e. an additional block mimicking the switches among different controllers that can be triggered by the platoon coordinator. For example, considering the Join scenario, from ego vehicle perspective, it is expected that the Platoon coordinator generates a sequence similar to the one defined in Figure 24 above from the ego vehicle perspective.

In the sequence above, the ego vehicle sends a JoinRequest, receives a JoinResponse and therefore starts platooning. The behaviour of the Platoon Coordinator under test can be visualized looking at the sequence diagram generated by the Manoeuvre Coordinator for Front Target, as reported in Figure 25.

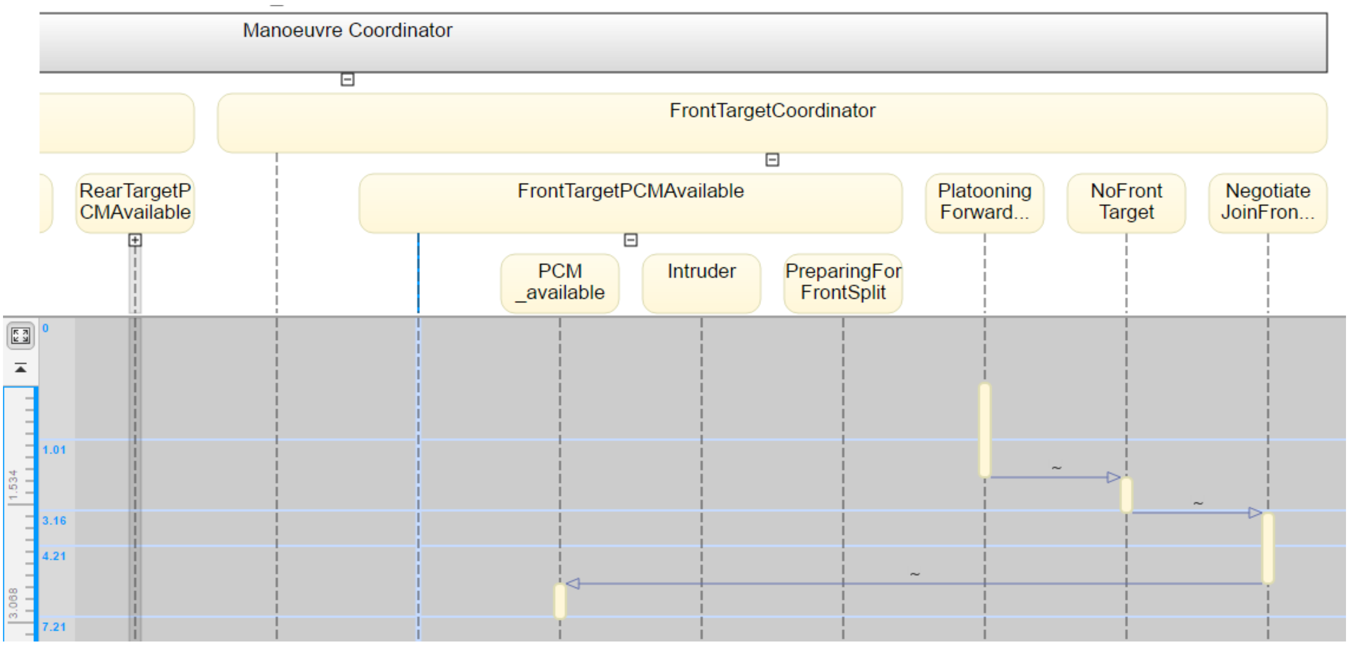


Figure 25: Front Target Coordinator sequence diagram for a Join manoeuvre from the ego-vehicle.

It can be seen that at the beginning the Front Target is not available (up to t = 3 s, after which the Front target is detected by the on-board sensors and CAMs are received, line 62 in Figure 22).

When this happens, as platoon functionality has been activated, a JoinRequest and JoinResponse are transmitted and received. Around t = 4.21 s PCM messages are received, then the Front Target Coordinator goes to the state PCM_Available, and therefore the ego-vehicle can start platooning and close the gap . This transition is highlighted with a yellow arrow in Figure 26.

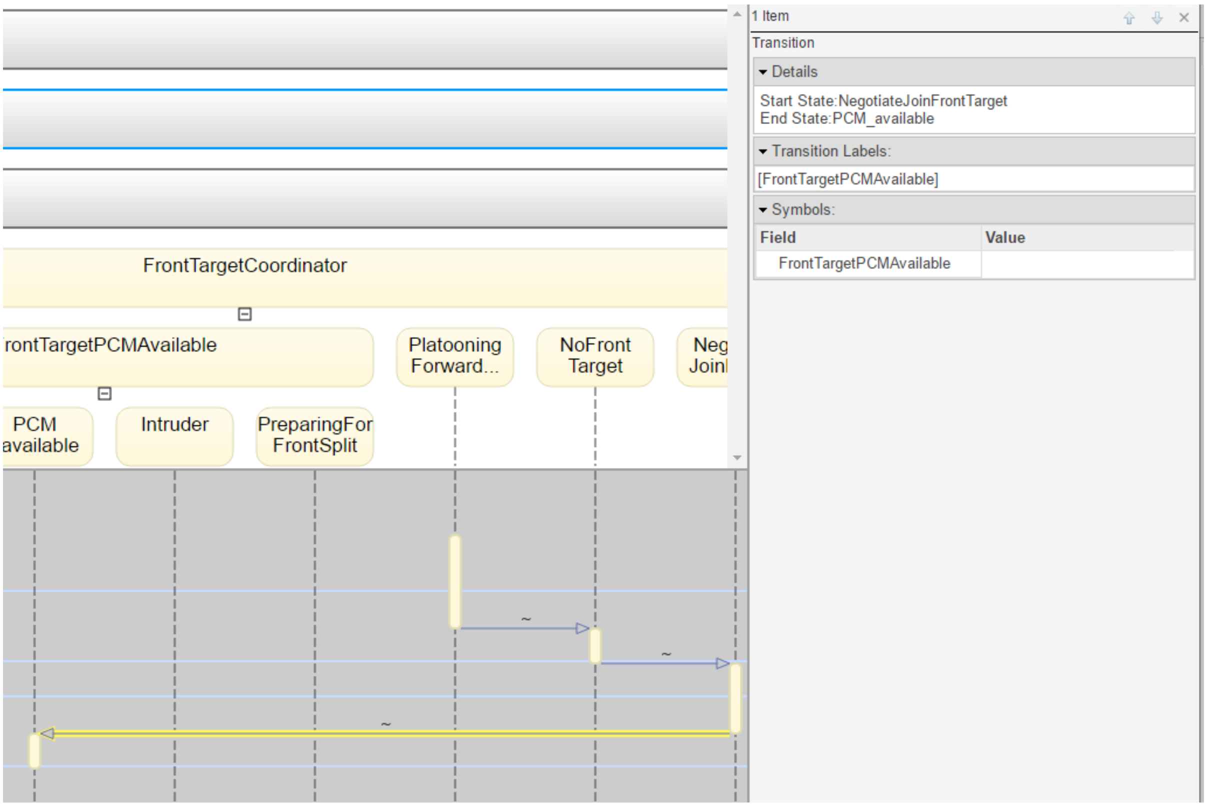


Figure 26: Manoeuvre Coordinator detecting PCM from Front target, and going to PCM Available state.

The lower level controllers are also emulated in the unit testing, and therefore, the process of gap-closing and CACC can also be visualised in terms of sequence diagrams, as is shown in Figure 27. It must be noted that the low level control modes shown are just an example and OEM specific implementations may be different.

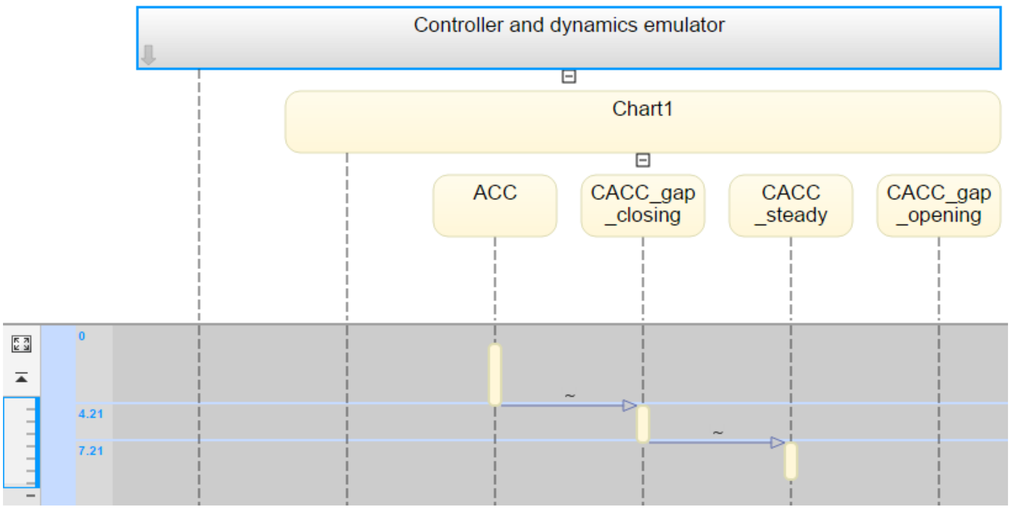


Figure 27: Sequence Diagram of the emulator of the lower level controllers for an ego-truck during a Join Manoeuvre.

6. SUMMARY AND CONCLUSION

This Deliverable 3.1 (D3.1) consists of 1) a reference implementation in software of the tactical layer modules, i.e. the Platoon Coordinator, and 2) a report describing this design in detail (this document). This report is also Project Milestone 3 (MS3): “Reference design of the tactical layer”. The source code of the software of the Platoon Coordinator is made available to the members of the Consortium (including the Commission Services) via the ENSEMBLE SharePoint site.

The relation of D3.1 with other T3.1 deliverables is that the V2X reference design is D3.2 and the prototyping of the total reference design, i.e. tactical layer and V2X design, in a rapid control prototyping setup is D3.3.

In this document the design of the tactical layer modules is described, starting from the specifications defined in WP 2. The tactical layer modules consist of a manoeuvre coordinator and a platoon status and property sharing module. The manoeuvre coordinator is responsible for handling the sequences of the interaction protocol to conduct the manoeuvres to join, leave or split the platoon. The platoon status and property sharing module ensures the collection and sharing of information that must be available to all vehicles in the platoon.

Next to the design, the implementation of the design in software is described. The Simulink programming environment of the MathWorks is used to develop the Platoon Coordinator model. This model is set up such that it can interface with and be integrated in the white label truck model, specified in WP 2. This white label truck model can be implemented in a HIL facility, which includes the reference V2X communication device (as hardware unit), for prototyping the reference design. Note that before the model can be embedded in the HIL facility it has to be compiled on a Rapid Control Prototyping device. In this compiling step, the readability of the software is lost. Therefore, this deliverable focusses on describing the (readable) source code, i.e. the Simulink model, of the Platoon Coordinator.

In order to understand and test the Platoon Coordinator model without having to integrate it in a truck environment, a Simulink environment for testing is provided. This test environment is also described in this document.

This document describes the reference design and software implementation of the tactical layer modules, i.e. the Platoon Coordinator. The design has to be considered as the first version, which is the baseline for implementation and further testing. It is likely that new insights obtained during implementation and testing in WP 3 and/or WP 5 will lead to future updates of the reference design. Version management will be used to track these changes. Finally, feedback to WP 2 is continuously provided if required changes affect the requirements and specifications.

7. BIBLIOGRAPHY

Atanassow, B. (2019b), *ENSEMBLE Platooning Technical Specification (M13)*.

Atanassow, B., Sjöberg, K. (2019a). *Platooning protocol definition and Communication strategy*. D2.8 of H2020 project ENSEMBLE, (www.platooningensemble.eu).

de Jongh, J. and van de Sluis, H.J.D. (2019). *Operational (non-automotive grade) ITS-G5 communication, supporting the multi-brand truck*. D3.2 of H2020 project ENSEMBLE, (www.platooningensemble.eu).

Konstantinopoulou, L., Coda, A., et al. (2019). *Functional specification for white-label truck*. D2.4 of H2020 project ENSEMBLE, (www.platooningensemble.eu).

Schmeitz, A.J.C., Yalcinkaya, S.I. and Van den Brand, D.J. (2019). *Generic open-source RCP-level reference implementation of the Tactical Layer.* D3.3 of H2020 project ENSEMBLE, (www.platooningensemble.eu)

Vissers, J., et al. (2018). *V1 Platooning use-cases, scenario definition and Platooning Levels*. D2.2 of H2020 project ENSEMBLE, (www.platooningensemble.eu).

APPENDIX A

Definitions

Term	Definition
Convoy	A truck platoon may be defined as trucks that travel together in convoy formation at a fixed gap distance typically less than 1 second apart up to 0.3 seconds. The vehicles closely follow each other using wireless vehicle-to-vehicle (V2V) communication and advanced driver assistance systems
Cut-in	A lane change manoeuvre performed by vehicles from the adjacent lane to the ego vehicle’s lane, at a distance close enough (i.e., shorter than desired inter vehicle distance) relative to the ego vehicle.
Cut-out	A lane change manoeuvre performed by vehicles from the ego lane to the adjacent lane.
Cut-through	A lane change manoeuvre performed by vehicles from the adjacent lane (e.g. left lane) to ego vehicle’s lane, followed by a lane change manoeuvre to the other adjacent lane (e.g. right lane).
Ego Vehicle	The vehicle from which the perspective is considered.
Emergency brake	Brake action with an acceleration of <-4 m/s2
Event	An event marks the time instant at which a transition of a state occurs, such that before and after an event, the system is in a different mode.
Fail-safe	A fail-safe in engineering is a design feature or practice that in the event of a specific type of failure, inherently responds in a way that will cause no or minimal harm to other equipment, the environment or to people.
Following truck	Each truck that is following behind a member of the platoon, being every truck except the leading and the trailing truck, when the system is in platoon mode.
Leading truck	The first truck of a truck platoon
Legal Safe Gap	Minimum allowed elapsed time/distance to be maintained by a standalone truck while driving according to Member States regulation (it could be 2 seconds, 50 meters or not present)
Manoeuvre (“activity”)	A particular (dynamic) behaviour which a system can perform (from a driver or other road user perspective) and that is different from standing still, is being considered a manoeuvre.

Term	Definition
ODD (operational design domain)	The ODD should describe the specific conditions under which a given automation function is intended to function. The ODD is the definition of where (such as what roadway types and speeds) and when (under what conditions, such as day/night, weather limits, etc.) an automation function is designed to operate.
Operational layer	The operational layer involves the vehicle actuator control (e.g. accelerating/braking, steering), the execution of the aforementioned manoeuvres, and the control of the individual vehicles in the platoon to automatically perform the platooning task. Here, the main control task is to regulate the inter-vehicle distance or velocity and, depending on the Platooning Level, the lateral position relative to the lane or to the preceding vehicle. Key performance requirements for this layer are vehicle following behaviour and (longitudinal and lateral) string stability of the platoon, where the latter is a necessary requirement to achieve a stable traffic flow and to achieve scalability with respect to platoon length, and the short-range wireless inter-vehicle communication is the key enabling technology.
Platoon	A group of two or more automated cooperative vehicles in line, maintaining a close distance, typically such a distance to reduce fuel consumption by air drag, to increase traffic safety by use of additional ADAS-technology, and to improve traffic throughput because vehicles are driving closer together and take up less space on the road.
Platoon Automation Levels	In analogy with the SAE automation levels subsequent platoon automation levels will incorporate an increasing set of automation functionalities, up to and including full vehicle automation in a multi-brand platoon in real traffic for the highest Platooning Automation Level. The definition of “platooning levels of automation” will comprise elements like e.g. the minimum time gap between the vehicles, whether there is lateral automation available, driving speed range, operational areas like motorways, etc. Three different levels are anticipated; called A, B and C.
Platoon candidate	A truck who intends to engage the platoon either from the front or the back of the platoon.
Platoon cohesion	Platoon cohesion refers to how well the members of the platoon remain within steady state conditions in various scenario conditions (e.g. slopes, speed changes).
Platoon disengaging	The ego-vehicle decides to disengage from the platoon itself or is requested by another member of the platoon to do so. When conditions are met the ego-vehicle starts to increase the gap between the trucks to a safe non-platooning gap. The disengaging is completed when the gap

Term	Definition
	is large enough (e.g. time gap of 1.5 seconds, which is depends on the operational safety based on vehicle dynamics and human reaction times is given). A.k.a. leave platoon
Platoon dissolve	All trucks are disengaging the platoon at the same time. A.k.a. decoupling, a.k.a. disassemble.
Platoon engaging	Using wireless communication (V2V), the Platoon Candidate sends an engaging request. When conditions are met the system starts to decrease the time gap between the trucks to the platooning time gap. A.k.a. join platoon
Platoon formation	Platoon formation is the process before platoon engaging in which it is determined if and in what format (e.g. composition) trucks can/should become part of a new / existing platoon. Platoon formation can be done on the fly, scheduled or a mixture of both. Platoon candidates may receive instructions during platoon formation (e.g. to adapt their velocity, to park at a certain location) to allow the start of the engaging procedure of the platoon.
Platoon split	The platoon is split in 2 new platoons who themselves continue as standalone entities.
Requirements	Description of system properties. Details of how the requirements shall be implemented at system level
Scenario	A scenario is a quantitative description of the ego vehicle, its activities and/or goals, its static environment, and its dynamic environment. From the perspective of the ego vehicle, a scenario contains all relevant events. Scenario is a combination of a manoeuvre (“activity”), ODD and events
Service layer	The service layer represents the platform on which logistical operations and new initiatives can operate.
Specifications	A group of two or more vehicles driving together in the same direction, not necessarily at short inter-vehicle distances and not necessarily using advanced driver assistance systems
Steady state	In systems theory, a system or a process is in a steady state if the variables (called state variables) which define the behaviour of the system or the process are unchanging in time. In the context of platooning this means that the relative velocity and gap between trucks is unchanging within tolerances from the system parameters.

Term	Definition
Strategic layer	The strategic layer is responsible for the high-level decision-making regarding the scheduling of platoons based on vehicle compatibility and Platooning Level, optimisation with respect to fuel consumption, travel times, destination, and impact on highway traffic flow and infrastructure, employing cooperative ITS cloud-based solutions. In addition, the routing of vehicles to allow for platoon forming is included in this layer. The strategic layer is implemented in a centralised fashion in so-called traffic control centres. Long-range wireless communication by existing cellular technology is used between a traffic control centre and vehicles/platoons and their drivers.
Tactical layer	The tactical layer coordinates the actual platoon forming (both from the tail of the platoon and through merging in the platoon) and platoon dissolution. In addition, this layer ensures platoon cohesion on hilly roads, and sets the desired platoon velocity, inter-vehicle distances (e.g. to prevent damaging bridges) and lateral offsets to mitigate road wear. This is implemented through the execution of an interaction protocol using the short-range wireless inter-vehicle communication (i.e. V2X). In fact, the interaction protocol is implemented by message sequences, initiating the manoeuvres that are necessary to form a platoon, to merge into it, or to dissolve it, also taking into account scheduling requirements due to vehicle compatibility.
Target Time Gap	Elapsed time to cover the inter vehicle distance by a truck indicated in seconds, agreed by all the Platoon members; it represents the minimum distance in seconds allowed inside the Platoon.
Time gap	Elapsed time to cover the inter vehicle distance by a truck indicated in seconds.
Trailing truck	The last truck of a truck platoon
Truck Platoon	Description of system properties. Details of how the requirements shall be implemented at system level
Use case	Use-cases describe how a system shall respond under various conditions to interactions from the user of the system or surroundings, e.g. other traffic participants or road conditions. The user is called actor on the system, and is often but not always a human being. In addition, the use-case describes the response of the system towards other traffic participants or environmental conditions. The use-cases are described as a sequence of actions, and the system shall behave according to the specified use-cases. The use-case often represents a desired behaviour or outcome. In the ensemble context a use case is an extension of scenario which add more information regarding specific internal system interactions, specific interactions with the actors (e.g. driver, I2V) and will add different flows (normal &

Term	Definition
	alternative e.g. successful and failed in relation to activation of the system / system elements).

Acronyms and abbreviations

Acronym / Abbreviation	Meaning
ACC	Adaptive Cruise Control
ABS	Anti-lock Braking System
ACSF	Automatically Commanded Steering Function
ADAS	Advanced driver assistance system
ADR	Agreement concerning the International Carriage of Dangerous Goods by Road
AEB	Autonomous Emergency Braking (System, AEBS)
ASIL	Automotive Safety Integrity Level
ASN.1	Abstract Syntax Notation One
BTP	Basic Transport Protocol
C-ACC	Cooperative Adaptive Cruise Control
C-ITS	Cooperative ITS
CA	Cooperative Awareness
CAD	Connected Automated Driving
CAM	Cooperative Awareness Message
CCH	Control Channel
CPU	Central Processing Unit
DEN	Decentralized Environmental Notification
DENM	Decentralized Environmental Notification Message
DITL	Driver-In-the-Loop
DOOTL	Driver-Out-Of-the Loop
DSRC	Dedicated Short-Range Communications

Acronym / Abbreviation	Meaning
ESF	Emergency steering function
ESP	Electronic Stability Program
ETSI	European Telecommunications Standards Institute
EU	European Union
FAD	Fully Automated Driving
FCW	Forward Collision Warning
FLC	Forward Looking Camera
FSC	Functional Safety Concept
GN	GeoNetworking
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GPU	Graphics Processing Unit
GRVA	Working Party on Automated/Autonomous and Connected Vehicles
GUI	Graphical User Interface
HAD	Highly Automated Driving
HARA	Hazard Analysis and Risk Assessment
HIL	Hardware-in-the-Loop
HMI	Human Machine Interface
HW	Hardware
I/O	Input/Output
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Organization for Standardization
ITL	In-The_Loop
ITS	Intelligent Transport System
IVI	Infrastructure to Vehicle Information message
LDWS	Lane Departure Warning System

Acronym / Abbreviation	Meaning
LKA	Lane Keeping Assist
LCA	Lane Centring Assist
LRR	Long Range Radar
LSG	Legal Safe Gap
MAP	MapData message
MIO	Most Important Object
MRR	Mid Range Radar
MVC	Modular Vehicle Combinations
OBD	On-Board Diagnostics
OS	Operating system
ODD	Operational Design Domain
OEM	Original Equipment Manufacturer
OOTL	Out-Of The-Loop
OTA	Over The Air
PAEB	Platooning Autonomous Emergency Braking
PMC	Platooning Mode Control
QM	Quality Management
RCP	Remote Control Parking
ROS	Robot Operating System
RSU	Road Side Unit
SA	Situation Awareness
SAE	SAE International, formerly the Society of Automotive Engineers
SCH	Service Channel
SDO	Standard Developing Organisations
SIL	Software-in-the-Loop
SOTIF	Safety of the Intended Function
SPAT	Signal Phase and Timing message

Acronym / Abbreviation	Meaning
SRR	Short Range Radar
SW	Software
TC	Technical Committee
TF	Task Force
TOR	Take-Over Request
TOT	Take-Over Time
TTG	Target Time Gap
UNECE	United Nations Economical Commission of Europe
V2I	Vehicle to Infrastructure
V2V	Vehicle to Vehicle
V2X	Vehicle to any (where x equals either vehicle or infrastructure)
VDA	Verband der Automobilindustrie (German Association of the Automotive Industry)
VECTO	Vehicle Energy Consumption Calculation Tool
VMAD	Validation Method for Automated Driving
WIFI	Wireless Fidelity
WLAN	Wireless Local Area Network
WP	Work Package