Towards a safety concept for cooperative automated driving

Citation for published version (APA):

Document status and date:
Published: 30/09/2015

Document Version:
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:
- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher’s website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the “Taverne” license above, please follow below link for the End User Agreement:
www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:
openaccess@tue.nl
providing details and we will investigate your claim.

Download date: 01. May. 2020
Towards a safety concept for cooperative automated driving

Dimitrios Tzempetzis
September 2015
Towards a safety concept for cooperative automated driving

Eindhoven University of Technology
Stan Ackermans Institute / Automotive Systems Design

The design that is described in this report has been carried out in accordance with the TU/e code of scientific conduct.

Partners

Steering Group

PROJECT OWNER: Ir. Ellen van Nunen
PROJECT MANAGER: Ir. Ellen van Nunen
PROJECT SUPERVISOR: Mark G. J. van den Brand
PROJECT SUPERVISOR: Peter S. C. Heuberger

Date

September 2015

A catalogue record is available from the Eindhoven University of Technology Library


(Eindverslagen Stan Ackermans Instituut ; 2015/056)
Abstract
Towards a safety concept for cooperative automated driving.

Keywords
Automated driving, Cooperative Adaptive Cruise Control, Functional Safety, Safety

Preferred reference

Partnership
This project was supported by Eindhoven University of Technology and TNO.

Disclaimer
Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the Eindhoven University of Technology or TNO. The views and opinions of authors expressed herein do not necessarily state or reflect those of the Eindhoven University of Technology or TNO, and shall not be used for advertising or product endorsement purposes.

Disclaimer Liability
While every effort will be made to ensure that the information contained within this report is accurate and up to date, Eindhoven University of Technology makes no warranty, representation or undertaking whether expressed or implied, nor does it assume any legal liability, whether direct or indirect, or responsibility for the accuracy, completeness, or usefulness of any information.

Trademarks
Product and company names mentioned herein may be trademarks and/or service marks of their respective owners. We use these names without any particular endorsement or with the intent to infringe the copyright of the respective owners.

Copyright
Copyright © 2015. Eindhoven University of Technology. All rights reserved. No part of the material protected by this copyright notice may be reproduced, modified, or redistributed in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the Eindhoven University of Technology and TNO.
Foreword

One of the business cases within TNO focuses on automated driving. Within this field several research projects are defined, amongst which the Eco-Twin project and the safety concept project. The Eco-Twin project aims for two-truck platooning, in which a following truck is automated both longitudinal as well as lateral. However, a driver can no longer be a backup in case of hazardous situations, so a safety concept is needed.

Dimitrios has worked on the approach towards the safety concept for cooperative automated driving. He rapidly developed his knowledge on this topic and easily found his way within TNO. His work resulted in two main contributions; one on the approach towards the design of a fail-safe algorithm. In this approach the ISO and V-model were coupled, and the steps of the Harmony Profile were shown to be beneficial. This resulted in an updated design of the supervisory block (which is the part of the software which decides on the control mode and the settings of a controller).

The second contribution was the definition and coupling of different phases of braking in case of a specific use case. Here, he showed, both in simulation as in practice, that his proposed stateflow definition reduces the amount of possible collisions significantly.

The process of reaching this result was well organized by his planning and risk analysis. He improved his writing and presentation skills during the project and this resulted in this interesting report!

Ellen van Nunen, MSc,
September 2015
Preface

This document forms the report of the project “Towards a safety concept for cooperative automated driving. The project was initiated by the Integrated Vehicle Safety (IVS) department of TNO located in Helmond. IVS performs significant research in the field of automated driving and cooperative mobility. The focus of the project was on the extension of the cooperative automated driving (CAD) system with safety related functionality.

September, 2015
Acknowledgements

This project was the final assignment of the Automotive System Design (ASD) program at Eindhoven University of Technology. Therefore, I would like to thank the program manager Dr. Peter Heuberger for giving me the chance to work in a competitive and challenging topic. Furthermore, I would like to express my gratitude to the university supervisor Prof. Mark van den Brand for the support and advice during the 9-month project.

Next, I would like to thank the TNO supervisor, Ellen van Nunen (MSc) for the successful guidance and coordination of the project. Besides, I would like to thank my TNO colleagues and particularly the graduate student Gerald Koudijs for the interesting discussions and cooperation during the execution of the project.

Last but not least, I would like to thank my family for supporting all my personal and professional decisions.

September, 2015
Executive Summary

Next to the production of Advanced Driving Assistance Systems (ADAS) the automotive companies focus on the development of automated driving solutions. In these solutions the driver cannot be considered to be backup of the vehicle. Therefore, significant effort is required for the advancement of the vehicle intelligence and particularly for the development of safety systems.

TNO aims to develop an automated driving system; the Cooperative Automated Driving (CAD). The Dutch research organization has created a business roadmap aiming to introduce road trains on highways. To achieve this, TNO should upgrade the CAD system with safety functionality and perform extensive testing. In this context, TNO initiated the safety concept project for the CAD system.

This report describes an approach for the safety concept development of the CAD system. The approach incorporates the system engineering methodology and the functional safety ISO26262 standard guidelines. It is described that the ISO26262 is generally compatible with system engineering methodology. However, the standard introduces several work-products which have to be executed during the development of a system. The main additions appear in the requirement analysis of the system. To verify the approach the CAD system was redesigned and a particular use case was executed.

For the design of the system the “Harmony” profile was followed. The steps of the profile include the requirement analysis, the functional analysis and the detailed design of the system. Applying the Harmony profile results in a modular and extensible design. In this project, the design was mainly performed at the decision unit of the CAD system. Next to the structural modifications, the behavior of the system was developed. Several braking actions were introduced in the decision unit of the system. The transitions between these actions are based on the criticality of the situation in case of failures. Therefore, the proposed strategy for the CAD system incorporates fault-tolerance and fail-safety.

To validate the developed CAD functionality a test plan has been created. This plan included a number of critical situations which were tested at both simulation and real-vehicle level. The results of the tests showed that the CAD system has improved safety in case of failures. It is presented that the CAD system can prevent a collision between the vehicles even in extreme situations.

To conclude, the developed design can be considered to be a starting point for further extension of the CAD system on a modular and ISO compatible way. Using the proposed approach to execute more use cases will result in the complete safety concept development.
Glossary

For the understanding of the reader the following terms are defined:

**ASD**: The Automotive System Design PDEng is a 2 year post-Master program established by the Eindhoven University of Technology (TU/e).

**CACC**: Cooperative Adaptive Cruise Control

**CAD**: Cooperative Automated Driving

**OEM**: Original Equipment Manufacturers. The term automotive OEMs addresses the companies that manufacture vehicles (such as DAF, VW, BMW etc.).

**ADAS**: Advance Driving Assistance Systems

**System development process**: The division of specific activities that are required for the development of a system.

**Functional Safety**: The part of the overall safety of a system equipment operating correctly in response to its inputs, including the safe management of likely operator errors, hardware failures and environmental changes.

**Operational Safety**: Safety of the users or other participants who are affected by the operation of a system. In this report the term operational safety corresponds to the safety of driver, passenger and other traffic participants when the CAD system is active.

**Fault**: Abnormal condition that can cause an element or an item to fail.

**Error**: Discrepancy between a computed, observed or measured value or condition, and the true, specified or theoretically correct value or condition.

**Failure**: Termination of the ability of an element to perform a function as required.

**Hazard**: Potential source of harm caused by malfunctioning behavior of the item.

**Fail-Safety**: In the event of failure no harm is caused, or at least a minimum of harm, to other devices or danger to personnel.

**Safety Use Cases**: The scenarios which describe safety related events such as system failures or dangerous traffic situation.

**Fault Tolerance**: In the event of a fault, the system shall continue its operation in the same or degraded mode without changing rapidly if no hazard is caused.

**Headway time**: The distance in sec between the front bumper of a vehicle and the rear bumper of the preceding vehicle.
# Table of Contents

Foreword

Preface

Acknowledgements

Executive Summary

Glossary

Table of Contents

List of Figures

List of Tables

1. Introduction
   1.1 Context
   1.2 Cooperative Automated Driving (CAD) system
   1.2.1 Outline

2. Problem Analysis
   2.1 Introduction
   2.2 Business Roadmap
   2.3 Stakeholder analysis
   2.3.1 EcoTwin stakeholders
   2.3.2 ASD final project stakeholders
   2.4 Project Scope
   2.5 Deliverables
   2.6 Conclusion

3. Current System Description
   3.1 Introduction
   3.2 CAD system description
   3.3 Hardware blocks
   3.4 Software blocks
   3.5 Conclusion

4. Safety concept approach
   4.1 Introduction
   4.2 Safety vocabulary
   4.3 System development process V-model
Bibliography ................................................................. 63
References ........................................................................ 63
Additional Reading .......................................................... 64

Appendix A ........................................................................ 65
Hardware blocks in CAD system ........................................ 65
Software blocks in CAD system ......................................... 66

Appendix B ........................................................................ 68
Functional Safety ISO26262 standard .................................. 68

Appendix C ........................................................................ 71
System development process – Waterfall model .................. 71

Appendix D ........................................................................ 73
System modelling language – SysML ................................. 73

Appendix E ........................................................................ 74
SUC 4: Act in case of GPS inaccuracy ............................... 74

Appendix F ........................................................................ 77
Test cases for SUC 1 ......................................................... 77

About the Author .............................................................. 79
List of Figures

Figure 1-1 Levels of Automation........................................................................................................2
Figure 1-2 Fuel Consumption Reduction in cooperative driving .........................................................2
Figure 2-1 EcoTwin project demonstration..........................................................................................5
Figure 2-2 Projection of CAD system and the projects initiated by TNO............................................8
Figure 3-1 Representation of CAD and test vehicle components.......................................................10
Figure 3-2 Overview of the CAD system..............................................................................................11
Figure 3-3 Supervisor Blocks.............................................................................................................12
Figure 4-1 Operational Safety vs Functional Safety............................................................................14
Figure 4-2 V-model diagram.............................................................................................................16
Figure 5-1 Diagram presenting the requirements types.........................................................................24
Figure 5-2 Use case diagram in SysML.............................................................................................31
Figure 5-3 Example of activity flow diagram – SUC 1.........................................................................33
Figure 5-4 Example of a sequence diagram – SUC1...........................................................................34
Figure 5-5 Example of the statechart diagram...................................................................................35
Figure 5-6 Block definition diagram for CAD...................................................................................36
Figure 6-1 Overview of the CAD system.............................................................................................37
Figure 6-2 The decision units in the supervisor....................................................................................38
Figure 6-3 The Safety Decision Unit as part of the block definition diagram..........................................39
Figure 6-4 The Supervisor block with the Decision Units....................................................................40
Figure 6-5 Depiction of host, lead and preceding vehicle....................................................................43
Figure 6-6 WiFi Mode Selection block...............................................................................................44
Figure 6-7 Toyota prius and DAF XF truck running the CAD system..................................................48
Figure 7-1 Mapping the safety use case parameters.............................................................................51
Figure 8-1 Work-breakdown for safety concept...................................................................................56
Figure 8-2 Timeline of the project plan................................................................................................57
Figure 9-1 First demonstration of EcoTwin project..............................................................................60
List of Tables

Table 5-1 Steady Safe States of the vehicle ................................................................. 20
Table 5-2 Safety Use Case 1 .................................................................................... 25
Table 5-3 Safety Use Case 2 .................................................................................... 26
Table 5-4 Safety Use Case 3 .................................................................................... 27
Table 5-5 Safety Use Case 4 .................................................................................... 28
Table 5-6 Safety Use Case 5 .................................................................................... 29
Table 5-7 Safety Use Case 6 .................................................................................... 30
Table 6-1 Functional Safety Requirements for the wireless module .................. 41
Table 6-2 Current CAD performance when communication fails ...................... 42
Table 7-1 Safety concept performance when communication fails .................. 52
Table 8-1 Risk analysis ......................................................................................... 57
1. Introduction

Abstract - Significant effort in the automotive domain is spent currently on the development of Advanced Driving Assistance Systems (ADAS). These systems are considered to be steps towards automated or autonomous driving which is a key objective for many automotive OEMs and research centers. Multiple benefits can be obtained from the introduction of automated driving on the highways such as safer transportation and travelling, reduction of fuel consumption and increase of traffic throughput. Particularly, for cooperative automated driving (CAD) systems, the technology allows short inter-vehicle distances. However, the short distances raise safety issues because the driver cannot be considered to be backup of the system. Therefore, further investigation has to be made on how the automated driving systems can become fail-safe. This project focuses on the approach that can be followed in order to make the CAD system safer.

1.1 Context
Several companies in the automotive domain currently focus on the development of autonomous or automated driving systems. The objectives of automotive OEMs are to make travelling and transportation safer, environmentally friendlier and more cost effective. Driving from point A to point B safely is the first priority for automated driving systems. According to the European Research and Safety report, 90% of the accidents in manual driving are caused by human error [1]. This number of accidents will be drastically reduced because of the reduction of the driver tasks in automated vehicles. Besides safety, the reduction of the fuel consumption and CO2 emissions as well as the improvement of traffic throughput lead the automotive OEMs and suppliers to focus their research on enhancing the intelligence of the car. This is achieved by adding high-tech systems in the vehicles. These high-tech systems are able to monitor the vehicle surroundings and perform a number of driving tasks. Examples of these systems are autonomous parking, blind spot detection, lane departure warning, lane keeping, lane merging, Adaptive Cruise Control (ACC), etc.

According to the Society of Automotive Engineers (SAE international), certain levels of automation are defined for the automotive systems. An overview of these levels is shown in Figure 1-1. It can be seen that each level of automation is linked to the required time for the driver to take over. This means that the higher the level of automation, the longer the time that the driver has to take over the control of the vehicle.

Most of the ADAS, which have entered the market, are level 1 or 2 systems. This means that the driver has to monitor the system and intervene within 2 seconds from the moment that a failure occurs. In other words, the driver is a backup of the system. Currently, the technology in the automotive domain evolves and more systems with automation level higher than 3 are expected to enter the market. The main challenge for the introduction of systems with level of automation higher than 3 (see Figure 1-1) is safety. The system must be capable of keeping its operation for at least 10 seconds after the event of a failure. Therefore, automotive OEMs and supplier companies, have to emphasize on fail-safe methods, which can guarantee that the system can handle all possible failures.
1.2 Cooperative Automated Driving (CAD) system

An example of a system, which is currently under development, is the Cooperative Automated Driving (CAD) system. This system combines longitudinal and lateral control of the vehicle. For the longitudinal control the CAD system is based on the Cooperative Adaptive Cruise Control (CACC) functionality while for the lateral control it performs Lateral Vehicle Following and Lane Keeping. The CACC incorporates the Adaptive Cruise Control (ACC) system with communication between vehicles.

The main benefit of adding communication is that two or more vehicles can drive on the same lane cooperatively keeping short distances of 0.3 seconds. As shown in Figure 1-2, the fuel reduction by driving at such short distances is significant. It can be seen that for the following vehicle the reduction varies between 8-13% due to less aerodynamic drag, while for the lead vehicle it varies from 2-8%. Moreover, the short inter-vehicle distances have direct impact on the traffic throughput increase.
However, keeping distances of 0.3 seconds can raise several concerns. An important question that arises is how safe the Cooperative Automated Driving (CAD) system in case that such short headway times of 0.3 seconds are set. Taking into account that the reaction time of the driver varies from 0.75-1.5 seconds, it can be easily realized that the driver is no longer a backup of the system. This leads to the need for development of a fail-safe system that can react in case of unexpected system or traffic behavior. This project examines the safety issues that arise in cooperative automated driving. The main goal of the project is to define an approach that can be followed to extend the current CAD system functionality with safety related functionality.

1.2.1 Outline

The report for this assignment is organized as follows. Chapter 2 presents an analysis of the problem including the business roadmap of TNO and a description of the project stakeholders. In Chapter 3, the current CAD system is described in order to provide insight to the reader about the features, the components and the specification of the system. Chapter 4 introduces the development process which is selected in this project and discusses how this process can be extended to include safety aspects. The next chapters of the report are defined based on the phases of the development process. First, Chapter 5 presents the requirement and system architecture development of the safety concept. The emphasis is on the methodology followed for the requirement gathering and the architecture development. Afterwards, Chapter 6 discusses the design of the extended CAD system and explains in detail the functionality for one use case. Furthermore, Chapter 6 gives insight on the implementation on the test vehicle. Chapter 7 includes the verification and validation and describes in detail how the testing of the system was performed. Then, Chapter 8 elaborates on the project management, while Chapter 9 presents the conclusions of the project.
2. Problem Analysis

Abstract – Currently, one of the trends in the automotive sector is automated driving. Hence there are many people, companies and institutions, which are directly interested in projects related to this trend. TNO, which is a research organization, plays a significant role in the development of automated driving systems. Following this goal, TNO has created a roadmap with the specific steps towards the introduction of automated driving on the highways. The chapter describes this roadmap and presents the list of the stakeholders for this project. Then, based on the problem analysis the scope of the project is formulated and the deliverables are set.

2.1 Introduction

The Integrated Vehicle Safety (IVS) department of TNO (Netherlands Organization for Applied Scientific Research) performs research in the field of Cooperative Automated Driving (CAD). Particularly, the CVS (Cooperative Vehicle Safety) group is involved in several projects in this domain and focuses on the development of an automated driving system, which will be ranked at automation level 4 (see Figure 1-1).

One of the projects that the IVS department is currently involved in, is the EcoTwin project (see Figure 2-1). This project uses the CAD functionality in trucks in order to achieve truck platooning, which means that two or more trucks can drive cooperatively at less than 1 second time distance. In case of a two-truck platoon, the lead truck drives manually (or in cruise control mode) while the second truck follows automatically releasing the driver from all his driving tasks.

The objective of TNO, which is the main stakeholder of this project, is to introduce “road trains” on highways. For this purpose the organization has set a business roadmap as described in Section 2.2. This chapter aims to provide further understanding to the reader about the “positioning” of the project in the long term plan of TNO. Next to that, the chapter aims to show how multiple stakeholders are directly or indirectly involved. This analysis will converge on the project scope formulation and the presentation of the project deliverables.
2.2 Business Roadmap

Truck platooning and cooperative driving are the concepts for which TNO currently sees great potential and, hence, invests in initiating projects in this domain. EcoTwin is an example of such a project. Therefore, TNO has created a business roadmap in order to handle the challenges for the introduction of this technology on highways.

The final application of the platooning on highways will result in two virtually connected trucks driving at less than a second headway time. Once the driver of the second truck (referred to as host truck) activates platooning, the host truck acts as a virtual trailer of the lead truck and releases the host truck driver from all its driving tasks. As the trucks communicate both ways, the platoon can adjust its speed and inter-vehicle distance without relying on the slow driver response. Moreover, the ad-hoc communication makes the platoon flexible in its formation as it can receive requests for merging with other platoons on the fly. Furthermore, the switching from platooning to manual driving is easily actuated by the driver. This is an example of how TNO captures platooning in the next years.

Even if TNO investigates the final application on the business level, the main goal of the organization still remains knowledge development and not delivery of the complete product to the market. On the other hand, it is important for TNO to adopt in their development process a number of steps which were previously necessary only for companies producing and bringing systems to the market. These steps are i.e. the compliance with the ISO26262 standard and the setup of a structured development process which makes the systems more easily extensible and tested. The reason to apply these changes is that the legislation for automated driving systems requires compliance with standards even for testing a system in its early development phase. This fact makes TNO interested in improving their development processes.

The business roadmap with respect to truck platooning created by TNO consists of the four following phases [2].

1. **Early research and development preparation until 2014.**
   This phase resulted in the development of the current CAD system functionality as described in Chapter 3.

2. **Wide-scale tests, technical feasibility and up-scaling from 2015 till 2019.**
   This phase corresponds to the actual state of the project where safety aspects are examined in parallel with upgrading of the nominal functionality of the system. The project in question is directed towards the goal of this phase, which is the extensive testing of the system in mixed traffic conditions.

3. **First commercial application of driver-guarded truck platooning from 2020 onwards.**
   This is a very crucial milestone in this roadmap, as it will mean that the system is mature enough to enter the market. This will also mean that TNO can further extend the functionality of the CAD system having already the experience of developing safe systems, which comply with the legislation.

4. **Broad commercial application with automated following vehicle from 2030 onwards.**
   This is the long-term goal of this roadmap, which has as prerequisite a lot of adaptations and evolutions in automated driving systems and assumes that the acceptance of such systems is guaranteed. Nevertheless, this goal is compatible with the view of many experts in the automotive domain who claim that automated driving will be fully introduced in 20 years time.

The EcoTwin project has entered the second phase of the roadmap and this phase will be completed in 5 years. Multiple challenges have to be tackled in this phase. TNO aims to handle them in cooperation with other project stakeholders and partners, as presented in Section 2.3.

2.3 Stakeholder analysis

The stakeholders of the project can be categorized in two main groups. The stakeholders that are directly interested in the EcoTwin project from the business point of view and those who are interested in the specific PDEng assignment from the academic point of view. The following section provides an overview of the project stakeholders and their objectives.
2.3.1 EcoTwin stakeholders

TNO
TNO is the main stakeholder of the EcoTwin project, as described in the business roadmap in Section 2.2.

Truck OEMs
The truck manufacturers are the main stakeholders as they want to increase automation in their trucks, they aim to reduce the fuel consumption of the truck and provide an advanced system which guarantees cheaper transportation. The whole system can also be a unique selling point for the OEM that will first introduce it.

Tier suppliers
The suppliers usually provide hardware components and/or complete systems to the OEMs so they play an important role in making such a system a complete product complying with regulations and including advanced technology. This system can be a new product for them or an extension of their existing systems, as CACC is also extension of the already produced ACC.

Truck owners and fleet owners
Truck or truck fleet owners are the main customers of OEMs and can play a significant role in setting up additional requirements for truck platooning. Their main interest is the fast, low cost and safe transportation and the CAD system aims to meet these objectives.

Local governments and European Union (EU)
Local governments and the EU play important role in the legal part of the automated driving system introduction on the public roads as new regulations and laws are required. Most of the automated driving systems are still in research, development or early testing phases. Consequently, it is important for the following phases including mixed traffic testing and safety related tests to have the legislation and policy well defined.

Service providers (Infrastructure management, Inspection, Service and Insurance companies)
The service providers are also important stakeholders as they need to adapt their systems and policy based on the additional features of truck platooning. Especially the insurance companies shall create new programs based on the safety performance of truck platooning applications. This is an additional reason why this project plays significant role in the business roadmap of TNO.

2.3.2 ASD final project stakeholders
The stakeholders for the PDEng assignment are the following.

1) PDEng Trainee: Experience in the field of system design, safety, functional safety, project integration, development and testing.

2) Stan Ackerman Institute (SAI) and Eindhoven University of Technology (TU/e):
The SAI and the TU/e established the post-master programs aimed at offering the trainees the opportunity to improve their knowledge and to earn working experience. This is achieved by assigning a design project which requires multidisciplinary thinking and well-structured approach. Moreover, the TU/e being the host University of the Automotive Systems Design (ASD) post-master program focuses on three main activities.
- Educate and provide highly-qualified engineers
- Perform research and deliver scientific excellence
- Contribute in innovative solutions and impact the society
These activities are also compatible with the goals of the ASD program.

2.4 Project Scope
After introducing the general context of the project as well as the main objectives and plans of TNO it is necessary to formulate the scope of this assignment. The scope of the project is to develop an approach towards a safety concept for the Cooperative Automated Driving (CAD) system. A safety subsystem needs to be designed and coupled with the system that is currently
being developed. This subsystem shall act in case of critical and recurrent failures activating a safety mechanism, which brings the vehicle in a predefined safe state. The vehicle shall remain in the safe state until the driver is able to take over the control.

Moreover, the approach towards the safety concept must take into account the ISO26262 guideline. Figure 2-2 is a graphical representation of how TNO intends to “bridge” the gap between the already developed CAD system and the ISO26262 functional safety standard. The “safety concept” forms an extension of the current CAD system and shall make the system compliant with the ISO26262 standard (see “complies with” association in Figure 2-2). As the development of a complete safety concept requires more time and resources than available for this assignment, an approach towards the safety concept is described in this report including the realization of several scenarios. Furthermore, TNO aims to develop an architectural framework, which shall combine the V-model [3] and ISO26262 standard (see “Architectural framework derives from V-model” in Figure 2-2). Moreover, the safety concept can follow the architectural framework. For this association a dashed line was selected, as during this project the architectural framework was not yet available. However, a possible extension of the safety concept in the future can be realized taking the architectural framework into account.

![Figure 2-2 Projection of CAD system and the projects initiated by TNO](image)

### 2.5 Deliverables
During the 9-month project the following deliverables will be produced for TNO:
- A functional model coupled with the current CAD model is the main deliverable. This functional model will be developed in Matlab Simulink 2014a software.
- The SysML diagrams which were created during the architecture description of the system.
- The testing results of the system. This deliverable actually contains a script which runs the simulated model, compares the results with the logged data of the real test and exports them to an Excel file.
- A detailed report describing the approach followed for the development of the safety concept will be a deliverable for TNO and TU/e as well.

### 2.6 Conclusion
Chapter 2 included an analysis of the problem that initiated this project. TNO has developed a roadmap in order to introduce truck platooning on highways. In the current stage of this roadmap, TNO intends to extend the system with safety related functionality. The objective is to make the system safer and compliant with the functional safety ISO26262 standard. In this context and taking into account the interest of several stakeholders the scope of the project in question was formulated.
3. Current System Description

Abstract – In this chapter the Cooperative Automated Driving (CAD) system, developed before the start of this project, is described. The description starts with an overview of the current system architecture and continues with a detailed presentation of the system components and functional blocks. The goal of this chapter is to provide the reader with the required background on the design of the system so that (s)he can understand the approach and the design decisions which follow in the next chapters.

3.1 Introduction
The objective of CAD is to automatically follow a preceding vehicle at a desired (short-following) distance (as seen in Section 3.2). Therefore, it requires several hardware and software components, which will be described in Section 3.3 and 3.4 respectively.

Within TNO, this system has been implemented on 3 test vehicles (Toyota Prius). This current implementation forms a basis for this chapter.

3.2 CAD system description
The CAD system consists out of longitudinal and lateral control.

The longitudinal control objective of the CAD system is to follow a preceding (lead) vehicle at a desired distance \( d_{des} \), which is described by the following spacing policy:

\[
d_{des} = r + h v
\]

in which \( r \) represents a standstill distance, \( h \) the desired time gap and \( v \) the host (or ego) velocity. Currently, the standstill distance is chosen to be 2.5m and the time gap equals 0.3s. This results in a desired distance of approximately 10m at 80kph. However, such a close following distance is only possible in case the intended acceleration of the preceding vehicle is known. This signal cannot be measured with an onboard-sensor on the host vehicle, so therefore communication between the lead and follower is required. Furthermore, the cruise speed should be set. This cruise speed will be maintained in case of no preceding vehicle and also serves as a maximum velocity of the host vehicle.

The lateral control objective is to follow the bumper of the preceding vehicle. Since the width of a truck is only about 50 cm smaller than the width of a lane, there’s not much margin for control errors. Therefore, it is also very important that the reference signal is as accurate as possible. As a backup (i.e. when no vehicle is detected, or when the detection is inaccurate) lane keeping is applied.

3.3 Hardware blocks
The performance of the CAD system is dependent on the environmental perception. Therefore sensors such as a radar and camera are required. Further, communication is required. In order to fuse a communicated message with the on-board sensors, GPS positions are used. Generally, GPS positions are inaccurate, so several on-board sensors are used (i.e. wheel encoders) to improve the accuracy [4]. An overview of all hardware components is given in Figure 3-1. A detailed description of the CAD system hardware can be found in Appendix A.
3.4 Software blocks

The software running on the platform uses the inputs of the sensors and the wireless module to control the vehicle, as shown in Figure 3-2. In this figure, the main software (blue boxes) and hardware blocks (brown boxes) are shown, including the direction of the data flow between these blocks. The software of the CAD consists of the following:

1. Sensor-processing unit: fuses the measurements of different sources and estimates the states (such as distance, bearing, range rate and acceleration) of surrounding vehicles.
2. Supervisor: decides on the control mode (on/off) and the settings of a controller.
3. High-level control: determines the desired acceleration and steering angle,
4. Low-level control: calculates the required driveline force for the engine to realize the outputs of the desired acceleration (output of high-level control) and represents the interface for the steering.

Each of these blocks is described in detail in Appendix A.

The supervisor block is very much related to the safety concept approach. Therefore, it will be described in more detail.
The supervisor block forms the decision unit of the system. The goal of this block is to set the control mode of the system. Therefore, the following aspects are taken into account:

1. User inputs, such as desired time gap and cruise-speed.
2. System status, such as warnings and errors of the CACC platform.
3. Longitudinal and lateral safety actions [5].

The current design consists of several blocks that incorporate nominal system functionality with safety features. The term nominal functionality includes not only longitudinal and lateral control mode selection but also actions which allow the merging and splitting of platoon vehicles. The blocks with this functionality are described as follows:

- The “Platoon control block” sets the cruise speed and the time gap based the platoon leader desired cruise speed or possible maneuver requests for merging or decoupling.
- The “Control Mode Selection” block includes state chart diagrams with the different modes of the system for longitudinal and lateral control. In the longitudinal direction the system can run in Cruise Control (CC) mode, Adaptive Cruise Control (ACC) mode, and Cooperative Adaptive Cruise Control (CACC) mode. For lateral control, vehicle following and lane keeping are the two options.
- The “Split & Merge request support” block determines the settings for the splitting and merging maneuvers between platoons or individual vehicles and the platoon.
- The “Longitudinal Safety” which includes the safe distance calculation. The safe distance between the vehicles depends on the current kinematic states of the lead and host and on the state of the communication. This block will be referred as “safety checker” in the following chapters. More details about the safe distance calculation can be found in [5].
- The “Lateral Safety” block checks the accuracy of the lateral position of the lead vehicle. The driver will be warned in case this lateral position is too inaccurate, so (s)he can take over.

Figure 3-3 shows how the above-mentioned blocks are connected inside the supervisor block in the current design. The blocks with safety functionality are linked with the nominal functionality blocks and intersect with them due to their required inputs. The outputs of the supervisor control form the input for the high-level control, namely the longitudinal and lateral control of the system.
3.5 Conclusion
This chapter describes the functionality and the components of the current CAD system. The objective of CAD is to follow a preceding vehicle. However, there might be other traffic participants that could bring this system in an unsafe situation. For these situations the CAD system is originally not designed. Furthermore, the system is dependent on its inputs, such as wireless communication. This means that in case of a failure, a hazardous situation can occur. How to bring the vehicle to a safe state in case of hazardous traffic situations and/or failures is the challenge that this project aims to tackle.

Figure 3-3 Supervisor Blocks
4. Safety concept approach

Abstract - Chapter 3 showed that the CAD system is vulnerable for hazardous traffic situations and failures. Because of that, the CAD needs to be extended with a safety concept. The approach for the safety concept development is described in this chapter. First, the definition of “safety” will be clarified. Then, the V-model, which is selected as system development process, is elaborated. Finally, the link between the V-model phases and the ISO26262 parts is presented.

4.1 Introduction
Chapter 3 presented an overview of the CAD system structure and functionality. It was concluded that the system is not designed to cope with hazardous traffic situations and failures. To extend the system with safety related functionality, a systematic approach has to be followed. The questions that arise are the following: “Which system development process is suitable for the design of a safety concept? How can this process be extended in order to include safety?”

The chapter aims to answer these two questions. As the term “safety” is a bit vague, Section 4.2 introduces and explains several safety-related terms. Afterwards, the V-model, which is selected as a system development process, is described. Selecting and following a process does not automatically imply that your system will be safe and compatible with existing standards. Therefore, the steps of a development process have to be extended with certain prescribed guidelines. The connection between the V-model phases and ISO26262 parts is described in Section 4.4.

4.2 Safety vocabulary
The section aims to clarify the safety vocabulary which is often used during a system development. The safety terms can be summarized to fail-safety, functional safety and operational safety.

Fail safety
The term “fail-safety” corresponds to a mechanism that is capable to return a system or an operation to a safe state in case of failure. The term fail-safety is mostly used in avionics and railway systems. There are several designs used for fail-safe mechanisms. These designs mainly differ on whether they consider system availability or safety as priority. Particularly in avionics the fail-safe mechanisms include duplication or triplication of hardware and software components. Applying redundancy to a system results in continuation of its operation for a specified time period before the pilot can take over.

Functional Safety
The automotive standard ISO26262 [6] (further described in Appendix B) provides guidelines in order to prevent or mitigate a hazard that can be caused by a hardware or software failure. This means that the standard covers safety on the functional level. A functionally safe system keeps its operation (no matter what this operation is) even when a failure occurs. All the electrical/electronic systems installed in vehicles shall comply with the functional safety standard.

Operational Safety
“Operational safety” is the safety of the driver, the passenger and other traffic participants. This term aims to capture hazardous situations that are not caused by failures. Such situations often result from dangerous driving maneuvers.

Relation between functional safety and fail-safe system
After the definition of the three safety related terms it is interesting to see their relation. Starting with the terms functional safety and fail-safety, there might still be confusion for the reader whether there is overlap or differences. Therefore, the following points are added to clarify the two terms describing their scope and applicability.
The functional safety ISO26262 standard covers the electrical/electronic systems that are used in automotive applications. It is not specifically initiated for automated driving applications or advanced driving assistance systems (ADAS). For the ADAS that have reached the maturity level of entering the production a different ISO standard is created (i.e. standard for ACC [7]).

Fail-safe mechanisms can be applied to both mechanical and electrical systems, whereas the ISO26262 standard is valid for electrical systems.

Fail-safe systems, as exemplified in transportation systems, are mainly based on hardware and software redundancy. It has been proven that these solutions are suitable for relatively static environments where no surprising events or rapid scenery changes are expected. However, the highway is not a static environment. Consequently, even if redundancy sounds as a promising solution, it might be not enough to ensure safe automated driving.

During the steps of the ISO where hardware and software development takes place (see Appendix B) fail-safe mechanisms can also be used. This means that fail-safe mechanisms can be part of a system design which follows the ISO26262 standard.

Fail-safe systems do not follow a specific methodology and all the design decisions rely on the system designers.

This short analysis shows that functional safety has a larger scope than fail-safety and can possibly incorporate fail-safe methods.

Relation between functional safety and operational safety
After the comparison between fail-safety and functional safety, the relation between operational safety and functional safety shall be explained. This relation is depicted in Figure 4-1.

Each automotive system has a specific functionality (see Functionality in Figure 4-1). For example, the Clima has to maintain the selected temperature in the vehicle and the Antilock Braking System (ABS) controls the distribution of the braking action at the wheels in order to prevent their locking. Many of these systems incorporate operational safety in their nominal functionality, such as the Adaptive Cruise Control (ACC) and the Collision Avoidance (CA). What is common for all these systems is that they have to be functionally safe, hence comply with the functional Safety ISO26262 standard. The following examples further clarify the above mentioned statements.

Example 1:
The Cruise Control (CC) system operates when the driver sets a certain speed and activates the system. Then, CC keeps the selected constant speed until the driver overrules or deactivates it. This is briefly the functionality of CC and it can be easily realized that the system does not react when the selected speed of the driver is higher than the speed of the preceding vehicle.
Consequently, the system does not guarantee passenger and traffic safety and requires the driver to act in case of a hazardous situation. On the other hand, the system is functionally safe which means that the controller which regulates the speed shall operate as intended and the hardware (i.e. cables, sensors) shall be reliable. Therefore, in Figure 4-1 CC is placed between Functionality and Functional Safety. All automotive systems (clima, radio) which don’t take into account operational safety and leave this task for the driver lie also between Functionality and Functional Safety. These systems are usually classified with level of automation 0 or 1 (see Figure 1-1).

Example 2:
Adaptive Cruise Control (ACC) is an extension of CC. During its operation, the system reaches the speed set by the driver only when a predefined distance is kept from the preceding vehicle. This means that the system can adaptively adjust the vehicle speed taking into account the behavior of other road users. From this brief description of ACC functionality it can be realized that the system incorporates operational safety in its nominal functionality. Moreover, the system, as an automotive system, shall be functionally safe. Therefore, ACC is placed between Operational Safety and Functional Safety, as shown in Figure 4-1.

Another attribute that distinguishes the automotive systems which cover safety aspects is the level of automation, which was introduced in Chapter 1. When a system is partially automated (automation level 2), the driver is allowed to have more time to react than when using a driving assistance system or non-automated system (automation level 0 or 1). This means that systems with level of automation higher than 1 usually have operational safety incorporated in their functionality and consequently lie on the left side of Figure 4-1 (see ACC, Collision Avoidance (CA), etc.). The CAD system belongs to this category as it aims to reach automation level 4. Hence, it can be positioned in the area where “Operational Safety” and “Functionality” overlap and intersect with functional safety.

4.3 System development process V-model

Next to the explanation of the safety vocabulary, it is necessary to select a system development process. Several system development processes exist in the literature [3]. Many companies introduced a systematic process for the development of their products because of the following benefits:
- Limited risk of developing a product which does not meet the customer objectives
- Efficient planning and monitoring of the tasks from the managers
- Repeatable results which do not only depend on the efforts of the designers

In this project, the selected development process is the V-model. The V-model is an extension of the waterfall model (see Appendix C) and includes phases in two directions; top-down and bottom-up. Figure 4-2 shows that each phase of the top-down direction is associated with a phase of the bottom-up direction. This means that during the early development of a system (top-down), the phases regarding the verification and the validation (bottom-up) can be also planned.

The main reason for selecting V-model is that it is used for the functional safety ISO26262 standard (see Figure 2-2).
The V-model phases, as shown in Figure 4-2, are described below:

1. **Planning**
   The main deliverable of this phase is a detailed project plan. Moreover, a project definition document has to be created including the scope, the activities, the deliverables and the risks of the project. More details about the planning can be found in Chapter 8.

2. **Requirements**
   This phase includes the requirements development of the system. The first task in this phase is to translate the customer objectives into well-defined functional and non-functional requirements. It is often the case, in particular for driver assistance systems, that the requirement analysis can also be performed using a use-case-driven approach, as proposed in [8].

3. **Architecture**
   The architecture phase includes the development of an architecture description which forms a representation of a system. The goal of an architecture description is to describe the structure and behavior of a system in a structured way. A structured way is achieved through the introduction of architecture views and levels of abstraction. [9] [10].

4. **Detailed Design**
   During this phase the high level functionality from the architecture phase has to be elaborated in a detailed design. This phase corresponds to the realization of the architecture description. This means that the decisions about software and hardware blocks have to be made in this phase. It often occurs that the detailed design of a system is executed iteratively, per use case as the V-model allows.

5. **Implementation**
   In this phase, the implementation of the system is performed. The detailed design including hardware and software components has to be installed on a real platform.

6. **Unit Testing**
   The next phase after the implementation is testing. Testing has to be performed on three different levels as shown in Figure 4-2. The first is the unit testing which includes individual testing of software and hardware units. The term unit corresponds to a hardware or software component which is the smallest part of a design that can be tested. Designers or test engineers usually create test scripts to check the functionality of system units.

7. **Integration Testing**
   The second level of testing is integration testing. The term integration shows that compatibility between units has to be explicitly checked. In complex automated driving
systems, different suppliers or external partners develop several units. Therefore, integration testing can be a challenging task, as it may identify possible mismatches.

8. System acceptance testing
After the unit and integration testing the next phase is the system acceptance testing. During this phase a number of test cases are generated. These test cases should correspond to the use cases, as defined in the requirement analysis. At this level the complete system functionality is checked in detail. The outcome of this phase forms the validation of the functional requirements.

It is important to mention that the terms verification and validation are often used at the bottom-up side of the V-model instead of unit, integration and system acceptance testing. Unit and integration testing (phase 6 & 7) belong to verification (see Section 7.2), while system acceptance testing belongs to system validation (see Section 7.2).

9. Maintenance
This phase is valid for systems which reach the production phase. The actual project does not reach this level, so this phase will remain out of the scope of this assignment.

Now that the V-model is described a question that arises is the following: Is the V-model enough to be used for the development of a safety concept? This question is answered in Section 4.4.

4.4 Adapting V-model to ISO26262
V-model is a generic system development process. This means that it can be used for the development of several systems, but it is not explicitly created for automotive systems or for safety-related systems. Therefore, using the V-model does not imply that a system will be functionally and/or operationally safe.

However, the goal of this project is to develop a system which will be both functionally and operationally safe. Hence, a number of adaptations have to be performed in the V-model phases in order to use it for the safety concept development. Particularly, the V-model phases, as presented in Section 4.3 have to be extended with the functional safety ISO26262 standard. The section describes how this link can be obtained. This extension is based on ISO26262, for which the reader is requested to read Appendix B.

Requirements
The requirement phase of the V-model usually includes the definition of functional and non-functional requirements. These requirement types are derived from the customer objectives and possible system constraints [11]. The functional safety ISO26262 standard includes two more requirement types on the system level; the functional safety requirements (FSR) and the technical safety requirements (TSR). These requirement types are derived from Safety Goals (see Appendix B).

To distinguish these two types of requirements, two different views are introduced; the “black-box” view and the “white-box” view. The “black-box” view is connected to the functional safety requirements. The designer steps out of the system and defines the required functionality seeing the system as a black box. On the other hand, the TSR form an extension of the FSR and they correspond to the “white-box” view. The white-box view means that the designer details the FSR having as input a preliminary architecture (see Figure B-1 in Appendix B). Therefore, each FSR can be elaborated on one or more TSR. An ISO compatible definition of FSR and TSR includes a number of attributes. The description of these attributes is presented in Chapter 5.

Architecture and Detailed Design
ISO parts 5 and 6 (see Appendix B) propose that well-tested methods and model-based design have to be applied during the software and hardware development of a system. In general, the system engineering methodology for architecture description and design of a system are
compatible with ISO26262. So, in the architecture and detailed design step, no changes to the V-model approach are needed.

**Implementation**

ISO part 7 describes how the production of a safety-related system has to be managed. However, the CAD system is still at prototype level. ISO26262 standard does not introduce specific tasks for the implementation of prototype systems. Therefore, for the current implementation of the CAD system, the ISO26262 standard was not taken into account.

**Unit Testing and Integration Testing**

Unit and integration testing phases follow the implementation phase during a system development. Unit and integration testing phases aim to verify whether a component and the integration meets the requirements. This is explained based on the definition of verification in ISO26262. According to the functional safety standard, there are two objectives in the planning of the verification phase:

1. To check for consistency and completeness of a system
2. To ensure that the work-products (outcome of each ISO part) meet the requirements of the standard. To achieve this objective, the ISO26262 suggests that a verification method must be selected for each work-product.

According to ISO, the proposed methods for verification are reviewing, inspection, model-checking, simulations, engineering analyses and testing. Testing, as verification method, corresponds to specific component (unit) testing or integration testing. Therefore, unit and integration testing are compatible with the ISO guidelines.

**System Acceptance Testing**

After unit and integration testing, the next phase is system acceptance testing. System acceptance testing corresponds to ISO part 4 (system validation). According to this part, validation is performed by testing the complete system (or subsystem) functionality. Therefore, also the V-model step of system acceptance testing fits well with ISO.

So, to adapt the V-model towards a safety related system, most adaptations are needed in the requirement phase. The other phases of the V-model match the ISO already quite well.

**4.5 Conclusion**

Extending a system with safety is challenging. To specify what safety means, the terms operational safety and functional safety had to be introduced. Then, the design of a system which is functionally and operationally safe has to follow a system development process. However, certain adaptations are necessary in the system development process in order to achieve compatibility with the ISO26262 standard. It was shown that most of these adaptations are in the requirement phase. In the remaining development phases, the ISO26262 allows the application of system engineering methodology. The following chapters describe how the development phases were executed.
5. Requirements and System Architecture

Abstract - The introduction of operational and functional safety in automotive systems and the description of the V-model was performed in Chapter 4. The next step is to start with the main work packages of the V-model: the requirements and the architecture. The requirement analysis for this project focuses on the types introduced in ISO26262: the functional safety requirements and the technical safety requirements. The method to define these requirement types is included. Then, a method of executing the architecture phase is presented by describing the Harmony profile in SysML.

5.1 Introduction

As introduced in Chapter 4, the requirement analysis must include four different types of requirements on the system level: functional, technical, functional safety and technical safety requirements. The first two types are widely used and described in literature [8] [11]. However, the functional safety requirements and the technical safety requirements are requirement types of the ISO26262 standard. The derivation of these requirement types is not trivial. Therefore, this chapter presents the attributes that need to be defined in order to obtain a complete and ISO26262 compatible requirement analysis for a system. The detailed list of requirements is not presented in the chapter, but it can be found in [12].

Next to the requirement analysis, an architecture description of the safety concept is added. The modelling language SysML is selected for the development of the architecture description [13]. The way to conduct the SysML diagrams is also presented by introducing the Harmony profile [14].

5.2 Functional Safety Requirements (FSR)

The goal of this section is to describe the necessary attributes for a requirement list which has to be compatible with ISO26262. These requirements have to be traced during the development of the safety concept and refined in the testing phase. According to ISO26262, the definition of FSR belongs to part 3 of ISO (see Appendix B). Therefore, the work-products of part 1 and 2 have to be developed as well as an Item definition, HARA and Safety goals (see Figure B-1). These work-products have been created up to a large extent for the EcoTwin project.

For the FSR which correspond to a “black-box” view, as introduced in Chapter 4, the following attributes were defined:

1. **ID**
   
   The index of the requirement is necessary for traceability and requirement management.

2. **Type**
   
   This attribute is set in order to distinguish the functional from the functional safety requirements. On this level it is often useful to define both types of requirements but a clear distinction is required. This means that for the functional requirements the safety related attributes cannot be filled.

3. **Requirement Description**
   
   The description of the requirements has a certain syntax which is useful for consistency and makes the addition of more entries easier. The selected syntax is the following.
   a. `<Object> shall <Action> when <condition>`
   b. `<Object> shall <Action> if <event>`
   
   The “object” in requirement description corresponds either to a (sub)system or a decision unit.

4. **Function**
The term function here corresponds to the system or sub-system which is active when the “object” performs an action. It is possible that the object mentioned in the description is also the function.

5. Function Operation Mode
The operating mode shows the status of the function at the moment that the event or condition mention in the requirement description applies.

6. Safety Goal / ASIL
The safety goals are the outcome of the Hazard Analysis and Risk Assessment (HARA) as described in Appendix B. Each functional safety requirement must be linked with one or more safety goals. The number of safety goals linked to a requirement depends on the number of hazards that can occur when the requirement is not fulfilled. Furthermore, in HARA one or multiple ASIL levels are assigned to each safety goal and hazard. It depends on the operating conditions of the vehicle (i.e. speed, distance from preceding vehicle, type of road, friction etc) whether a hazard is severe (ASIL D) or negligible (ASIL A). It is very important to assign the ASIL level of a safety goal to the requirement linked to this safety goal. The reasons for this assignment are the following:
- Design decisions (hardware, software) are affected by the ASIL level. ISO includes certain specifications for design of components with different ASIL levels.
- Priority of the requirements is also affected. This means that requirements with ASIL D will be considered first.
- The selection of the “Steady Safe State” (see Attribute 7) is also affected.

7. Steady Safe State
As described in Section 2.4, the safety subsystem shall bring the vehicle to a safe state in case of failure. This means that, already during the requirements development, a number of safe states have to be defined and linked to the requirements. In general, there are multiple states that can be considered to be safe for the vehicle. To distinguish a safe state from the selected safe states the term “steady safe state” is introduced. Therefore, a number of steady safe states has been defined in order to describe the final state that the vehicle must reach. Table 5-1 shows the selected steady safe states.

<table>
<thead>
<tr>
<th>Index</th>
<th>Steady Safe States</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Default : keep the selected headway time</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>Lane Keeping (Stay in the middle of the lane)</td>
<td>Lateral</td>
</tr>
<tr>
<td>2</td>
<td>Truck CAD ON with a distance of $d_{\text{safe}}$ m (Safety Checker)</td>
<td>Long.</td>
</tr>
<tr>
<td>3</td>
<td>Truck CAD ON with a distance of $d_{\text{safe}}$ m and Lane Keeping</td>
<td>Long. &amp; Lateral</td>
</tr>
<tr>
<td>4</td>
<td>Truck Degrades to ACC (1.5 - 2.2 sec headway time)</td>
<td>Long.</td>
</tr>
<tr>
<td>5</td>
<td>Truck Degrades to ACC (1.5 - 2.2 sec headway time) and Lane Keeping</td>
<td>Long. &amp; Lateral</td>
</tr>
<tr>
<td>6</td>
<td>Standstill in the same lane (CA)</td>
<td>Long.</td>
</tr>
<tr>
<td>7</td>
<td>Standstill in emergency lane</td>
<td>Long. &amp; Lateral</td>
</tr>
<tr>
<td>8</td>
<td>Truck drives in adjacent lane with ACC (1.5 - 2.2 sec headway time)</td>
<td>Long. &amp; Lateral</td>
</tr>
<tr>
<td>9</td>
<td>CAD ON for TBD sec with redundant CPU</td>
<td>Long. &amp; Lateral</td>
</tr>
</tbody>
</table>

Note: $d_{\text{safe}}$ is the safe distance calculated by the “safety checker” block [5].

The steady safe states are selected taking into account the operating conditions that usually apply on highways. It can be seen in Table 5-1 that some of the steady safe states aim at degradation of the system functionality (steady safe states 1-5), while the remaining correspond to more severe actions (steady safe states 6-9). The link between steady safe states, ASIL levels and specific requirements can be seen in the requirement document [12].
It is important to mention that the list with the steady safe states can be updated in a later development stage. These updates are expected during the verification of the requirement analysis. Furthermore, it might be necessary to select two or more steady safe states for a requirement. In such cases, the best solution is to further elaborate the functional safety requirement or split it to two or more requirements.

8. Priority
After completing the FSR list, it is still questionable with which requirement the development of the system shall start. The “Priority” attribute aims to help towards this decision. The value of this attribute shall be dependent on the ASIL level but it can also be a designer decision. For example, for requirements with the same ASIL, the designer has to consider which will be executed first. The priority value ranges from 1 to 3 where 1 is assigned to the highest priority requirement.

9. Status
The status of each requirement is important for the requirement management and traceability. The status of a requirement can be one of the following:
   a. Proposed: This means that the requirement has been requested (the start point)
   b. Approved: The requirement has been reviewed by the owners, and responsible and is agreed on.
   c. Implemented: The requirement has been covered in the development process either from a software block or a hardware block.
   d. Verified: A requirement is verified with one of the methods suggested in ISO26262 (see Chapter 7).

10. Category
This field aims to group the FSR into three categories.
   a. Preventive: The requirement aims to prevent an event before it becomes dangerous.
   b. Mitigating: The requirement aims to reduce severity and impact of an accident.
   c. Dynamic: The requirement describes actions related to safe handling and ride characteristics. These characteristics usually correspond to the vehicle capabilities and the dynamic boundaries, i.e. maximum braking, maximum lateral acceleration, etc. [15].

5.3 Technical Safety Requirements (TSR)
As described in Section 4.4, the technical safety requirements belong on the system level and form an elaboration of the functional safety requirements. Similar to the FSR, there are certain attributes assigned to each technical safety requirement.

1. ID
   The unique index of the technical safety requirement.

2. FSR_ID
   The index of the functional safety requirement that is linked to the technical safety requirement.

3. Requirement Description
   The description includes the same syntax with the corresponding FSR attribute.

4. Functional Dependencies
   This attribute includes the hardware components and software components which will be activated or used during the implementation of the TSR.

5. System Operation (i.e. headway time)
   This attribute includes the operating conditions for which this requirement is valid. According to the safety use cases an operating condition can be the headway time between the two trucks and/or the speed of the truck.

6. Safety Mechanism
   The safety mechanism is directly related with the steady safe states, as described in FSR. It addresses the transition from an unsafe state to a steady safe state. To define this transition the following parameters are necessary:
      a. Detection Measures
The detection mechanism includes sensor(s) parameters and/or functions developed for system awareness.

b. Steady safe state measures
This attribute includes the actions that need to be taken in order the vehicle to reach a steady safe state.

c. Fault tolerant time interval
The required time period that the system can handle a failure. For example, if a steady safe state is reached, the fault tolerant time interval corresponds to the required time period to reach it.

It is important to mention that in the phase of the project when the technical safety requirements are defined it is not always possible to include a complete detection mechanism or safety mechanism. This means that many of the attributes have to be filled or updated on a later stage.

7. Hardware reliability
In this attribute the hardware reliability is addressed by using the accuracy of the hardware component. Moreover, the result of a Failure Mode Effect Analysis (FMEA) could be added for several sensors or components.

8. Status
This attribute is the same as described in FSR attributes.

The list with the technical safety requirements can be found in the requirements document [12].

5.4 System Architecture
After completing the requirement definition, the next step is to create an architecture description of the safety concept. An architecture description expresses the system architecture, therefore defines the system structure and behavior. A complete architecture description should cover multiple abstraction levels; functional, logical and technical [10]. During this phase, a functional architecture description was developed. The logical and technical architecture description correspond to the detailed design phase of the system, as described in Chapter 6.

The selected modelling language for the functional architecture description is SysML [13]. SysML is a graphical modelling language including structural and behavioral diagrams (see Appendix D). The reasons for selecting SysML are listed below:

- SysML is compatible with the V-model. All the phases of the V-model can be executed using this language. Other languages (C++, MATLAB), which are often used for system, are not sufficient for requirement analysis or architecture description.
- SysML supports model-based design. This is important for the compliance of a system with the ISO26262 standard, as described in Section 4.4.
- SysML supports models including software and hardware components.
- SysML diagrams can be useful for all hierarchical levels of a company; from management till technicians.
- There are software tools for SysML which are compatible with other modelling languages. For example, a MATLAB block can be executed from a software tool that supports SysML.

The next step is the selection of a software tool that supports SysML. IBM Rhapsody was selected for two reasons. First, it is a powerful tool that gives multiple options to the designer. Second, it was available in TU/e, therefore no time (and money) was spent in purchasing of another tool. The reader should know that analysis of existing tools was not a goal of the project.

The next challenge is to select a method to create the SysML diagrams. According to ISO26262, model-based development of a system is desired. Furthermore, ISO26262 suggests that a system has to be modular and extensible. To achieve these, a method based on the “Harmony” profile was followed, as described in Section 5.4.1.
5.4.1 Harmony profile in SysML

The “Harmony” profile is a model-based system engineering approach that provides a step-by-step way for a functional architecture description development. The key objectives of this profile are the following:

- **Identification and derivation of required system functions.**
  Harmony profile supports model development per use case. This results in the derivation of system functions per use case. Moreover, the developed models per use case can be independently executed.

- **Identification of associated system modes and states.**
  Based on the defined system functions, the specific modes and states can be defined. These modes and states can be allocated to a specific block of a system, i.e., the decision unit of a system. If the system structure is not known, this allocation is not possible. However, Harmony suggests in such case that modes and states should be assigned to a block named as the use case.

- **Allocation of the identified system functions and states to a subsystem.**
  After creating a model including functions and modes, the next step is to define the structure of the system. Then, the modes and the states should be allocated on the defined blocks or subsystems.

From these key objectives, it can be concluded that Harmony is mostly useful for the design of a new system of which the structure is not known. Since the CAD system is already developed, the system structure is already known. So, the Harmony profile was mainly followed during the behavioral analysis. For the structure of the system, a block definition diagram (see Structural diagrams in Appendix D) was created. This diagram includes the blocks of the current system and additions based on the requirement analysis. A detailed description of the structure is added in Chapter 6.

The steps of the Harmony profile and how they were applied to this project are described below.

**Harmony Step 1. Requirement Analysis**

The main purpose of the requirement analysis step is to analyze the stakeholder objectives, translate them into system requirements and define the system use cases. Figure 5-1 shows the relation between the different types of requirements. As already mentioned, the technical safety requirements are derived from the functional safety requirements. Moreover, the horizontal line between the functional and the functional safety requirements (similarly the one between technical and technical safety requirements) shows that there is association between these two types of requirements. The association arises because the requirement types belong to the same abstraction level (system level).

At this step, the requirements for the safety concept were introduced in IBM Rhapsody in four tables. These tables correspond to the packages Functional Requirements, Functional Safety Requirements, Technical Requirements and Technical Safety Requirements.
After the requirement development, the next task is the definition of the system use cases. The use cases for this project are executed based on the analysis of operational safety and functional safety (see Section 4.2). Therefore, they were named as “safety use cases”. It must be mentioned that it is up to the designer to define the safety use cases first and then to derive the FSR and TSR. In that case, the safety use cases have to be linked with the previous work-products of ISO26262; HARA and Safety goals. An example of this link is shown in Figure B-1 (see Appendix B). The following tables list the safety use cases on which the project focuses.
<table>
<thead>
<tr>
<th>ID</th>
<th>SU_C_1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Act in case of WiFi failure when following at 0.3s</td>
</tr>
<tr>
<td>Use case description</td>
<td>The host truck follows with CAD system active the lead truck with a velocity of 80 km/h. The nominal case is when everything is functional. This use case is relevant when communication (WiFi) is not functional for more than ( x ) seconds, with ( x \in {0.1, 1, 10} )</td>
</tr>
<tr>
<td>1) Lead truck continues its previous behavior</td>
<td></td>
</tr>
<tr>
<td>2) Lead truck brakes (heavily or smoothly)</td>
<td></td>
</tr>
<tr>
<td>Background attributes</td>
<td>Area type</td>
</tr>
<tr>
<td></td>
<td>Road segment</td>
</tr>
<tr>
<td></td>
<td>Environmental conditions</td>
</tr>
<tr>
<td></td>
<td>Speed range</td>
</tr>
<tr>
<td>Participant 1 attributes</td>
<td>Type of participant</td>
</tr>
<tr>
<td></td>
<td>Start of position</td>
</tr>
<tr>
<td></td>
<td>Maneuvre</td>
</tr>
<tr>
<td>Participant 2 attributes</td>
<td>Type of participant</td>
</tr>
<tr>
<td></td>
<td>Start of position</td>
</tr>
<tr>
<td></td>
<td>Maneuvre</td>
</tr>
<tr>
<td>Sketch</td>
<td>![Sketch of vehicle interaction]</td>
</tr>
<tr>
<td>Table 5-3 Safety Use Case 2</td>
<td></td>
</tr>
<tr>
<td>-----------------------------</td>
<td></td>
</tr>
<tr>
<td><strong>ID</strong></td>
<td>SUC 2</td>
</tr>
<tr>
<td><strong>Name</strong></td>
<td>Act in case of Cut-in when following at 0.3s</td>
</tr>
<tr>
<td><strong>Use case description</strong></td>
<td>The host truck follows with CAD system active the lead truck with a velocity of 80 km/h. The nominal case is when everything is functional. This use case is relevant when the system is functional</td>
</tr>
<tr>
<td><strong>Background attributes</strong></td>
<td></td>
</tr>
<tr>
<td>Area type</td>
<td>N270+A270 highway</td>
</tr>
<tr>
<td>Road segment</td>
<td>Detectable lane markings</td>
</tr>
<tr>
<td>Environmental conditions</td>
<td>Weather is such that lane markings can be well detected by the on-board camera.</td>
</tr>
<tr>
<td>Speed range</td>
<td>0-90 km/h</td>
</tr>
<tr>
<td><strong>Participant 1 attributes</strong></td>
<td></td>
</tr>
<tr>
<td>Type of participant</td>
<td>Truck which communicates its desired acceleration</td>
</tr>
<tr>
<td>Start of position</td>
<td>Driving at the beginning of the N270 at a velocity of 80 km/h.</td>
</tr>
<tr>
<td>Manoeuvre</td>
<td>Manual driving and manual steering. This can include braking actions up to -6 m/s².</td>
</tr>
<tr>
<td><strong>Participant 2 attributes</strong></td>
<td></td>
</tr>
<tr>
<td>Type of participant</td>
<td>Automated Truck</td>
</tr>
<tr>
<td>Start of position</td>
<td>Driving in 0.3sec behind truck 1 (max 1m lateral offset), also at a velocity of 80 km/h.</td>
</tr>
<tr>
<td>Manoeuvre</td>
<td>Follow lead truck on a headway time of 0.3s (as small as possible) and avoid collision in case of lead vehicle braking</td>
</tr>
<tr>
<td><strong>Participant 3 attributes</strong></td>
<td></td>
</tr>
<tr>
<td>Type of participant</td>
<td>Unequipped vehicle</td>
</tr>
<tr>
<td>Start of position</td>
<td>On the left lane of the highway</td>
</tr>
<tr>
<td>Manoeuvre</td>
<td>Cut-in</td>
</tr>
</tbody>
</table>

**Sketch**

![Diagram showing the scenarios of the safety use case](attachment:image.png)
<table>
<thead>
<tr>
<th>Table 5-4 Safety Use Case 3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ID</strong></td>
</tr>
<tr>
<td><strong>Name</strong></td>
</tr>
</tbody>
</table>
| **Use case description**    | The host truck follows with CAD system active the lead truck with a velocity of 80 km/h. The nominal case is when everything is functional. This use case is relevant when the system is functional  
1) A vehicle without wireless communication cuts through (enters the truck platoon and leaves)  |
<p>| <strong>Background attributes</strong>   | <img src="image" alt="Sketch" />  |
| <strong>Area type</strong>               | N270+A270 highway  |
| <strong>Detectable lane markings</strong>| Detectable lane markings  |
| <strong>Environmental conditions</strong>| Weather is such that lane markings can be well detected by the on-board camera.  |
| <strong>Speed range</strong>             | 0-90 km/h  |
| <strong>Participant 1 attributes</strong>| <img src="image" alt="Participant 1" />  |
| <strong>Type of participant</strong>     | Truck which communicates its desired acceleration  |
| <strong>Start of position</strong>       | Driving at the beginning of the N270 at a velocity of 80 km/h.  |
| <strong>Manoeuvre</strong>               | Manual driving and manual steering. This can include braking actions up to -6m/s².  |
| <strong>Participant 2 attributes</strong>| <img src="image" alt="Participant 2" />  |
| <strong>Type of participant</strong>     | Automated Truck  |
| <strong>Start of position</strong>       | Driving in 0.3sec behind truck 1 (max 1m lateral offset), also at a velocity of 80 km/h.  |
| <strong>Manoeuvre</strong>               | Follow lead truck and avoid lateral following of cut-through vehicle  |
| <strong>Participant 3 attributes</strong>| <img src="image" alt="Participant 3" />  |
| <strong>Type of participant</strong>     | Unequipped vehicle  |
| <strong>Start of position</strong>       | On the left lane of the highway  |
| <strong>Manoeuvre</strong>               | Cut-through  |</p>
<table>
<thead>
<tr>
<th>Table 5-5 Safety Use Case 4</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ID</strong></td>
</tr>
<tr>
<td><strong>Name</strong></td>
</tr>
</tbody>
</table>
| **Use case description** | The host truck follows with CAD system active the lead truck with a velocity of 80 km/h. The nominal case is when everything is functional. **This use case is relevant when GPS input is not accurate**  
1) Lead truck continues its previous behavior  
2) Lead truck brakes (heavily or smoothly) |
| **Background attributes** | **Area type**  
N270+A270 highway |
| | **Road segment**  
Detectable lane markings |
| | **Environmental conditions**  
Weather is such that lane markings can be well detected by the on-board camera. |
| | **Speed range**  
0-90 km/h |
| **Participant 1 attributes** | **Type of participant**  
Truck which communicates its desired acceleration |
| | **Start of position**  
Driving at the beginning of the N270 at a velocity of 80 km/h. |
| | **Manoeuvre**  
Manual driving and manual steering. This can include braking actions up to -6m/s². |
| **Participant 2 attributes** | **Type of participant**  
Automated Truck |
| | **Start of position**  
Driving in 0.3 sec behind truck 1 (max 1m lateral offset), also at a velocity of 80 km/h. |
| | **Manoeuvre**  
Follow lead truck and avoid collision in case of lead vehicle braking |

**Sketch**

[Diagram of a highway with two trucks following each other, illustrating the scenario.]
### Table 5-6 Safety Use Case 5

<table>
<thead>
<tr>
<th>ID</th>
<th>SUC_5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Act in case of Radar failure when following at 0.3s</td>
</tr>
</tbody>
</table>

#### Use case description

The host truck follows with CAD system active the lead truck with a velocity of 80 km/h. The nominal case is when everything is functional. **This use case is relevant when Radar is not functional**

1. Lead truck continues its previous behavior
2. Lead truck brakes (heavily or smoothly)
3. Lead truck performs lane change

#### Background attributes

<table>
<thead>
<tr>
<th>Area type</th>
<th>N270+A270 highway</th>
</tr>
</thead>
<tbody>
<tr>
<td>Road segment</td>
<td>Detectable lane markings</td>
</tr>
<tr>
<td>Environmental conditions</td>
<td>Weather is such that lane markings can be well detected by the on-board camera.</td>
</tr>
<tr>
<td>Speed range</td>
<td>0-90 km/h</td>
</tr>
</tbody>
</table>

#### Participant 1 attributes

<table>
<thead>
<tr>
<th>Type of participant</th>
<th>Truck which communicates its desired acceleration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start of position</td>
<td>Driving at the beginning of the N270 at a velocity of 80 km/h.</td>
</tr>
<tr>
<td>Manoeuvre</td>
<td>Manual driving and manual steering. This can include braking actions up to -6m/s².</td>
</tr>
</tbody>
</table>

#### Participant 2 attributes

<table>
<thead>
<tr>
<th>Type of participant</th>
<th>Automated Truck</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start of position</td>
<td>Driving in 0.3sec behind truck 1 (max 1m lateral offset), also at a velocity of 80 km/h.</td>
</tr>
<tr>
<td>Manoeuvre</td>
<td>Follow lead truck on a headway time of 0.3s (as small as possible), avoid collision in case of lead vehicle braking or follow the lateral manoeuvre</td>
</tr>
</tbody>
</table>

#### Sketch

![Sketch of safety use case 5](image-url)
Table 5-7 Safety Use Case 6

<table>
<thead>
<tr>
<th>ID</th>
<th>SUC 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Act in case of Camera failure when following at 0.3s</td>
</tr>
</tbody>
</table>
| Use case description | The host truck follows with CAD system active the lead truck with a velocity of 80 km/h. The nominal case is when everything is functional.  
This use case is relevant when Camera is not functional  
1) Lead truck continues its previous behavior  
2) Lead truck brakes (heavily or smoothly)  
3) Lead truck performs lane change |
| Background attributes | Area type | N270+A270 highway |
| | Road segment | Detectable lane markings |
| | Environmental conditions | Weather is such that lane markings can be well detected by the on-board camera. |
| | Speed range | 0-90 km/h |
| Participant 1 attributes | Type of participant | Truck which communicates its desired acceleration |
| | Start of position | Driving at the beginning of the N270 at a velocity of 80 km/h. |
| | Manoeuvre | Manual driving and manual steering. This can include braking actions up to -6m/s². |
| Participant 2 attributes | Type of participant | Automated Truck |
| | Start of position | Driving in 0.3sec behind truck 1 (max 1m lateral offset), also at a velocity of 80 km/h. |
| | Manoeuvre | Follow lead truck on a headway time of 0.3s (as small as possible), avoid collision in case of lead vehicle braking or follow the lateral manoeuvre |

As shown in Table 5-2 to Table 5-7, the safety use cases include not only sensor failures (SUC 1, 4, 5 and 6), but also hazardous traffic situations (SUC 2 and 3). The cut-in and cut-through scenarios are potentially hazardous especially when the host truck is in lateral vehicle following mode.

The focus of this report is to extend the CAD system design to cover the safety use case 1:
1. a communication failure (of different durations) in combination with an emergency brake of the lead vehicle  
2. a communication failure (of different durations) in combination with no brake action of the lead vehicle. This situation could be considered as a test for “false positives”; it should be avoided that the vehicle brakes heavily every time the communication fails.

In SysML, the use cases can be projected in a Use Case Diagram (UCD). A UCD includes use cases, actors and the relation between use cases. The purpose of UCD is to describe the functionality of a system in a graphical way. To achieve this for the CAD system, the nominal use cases and the safety use cases have to be included. Figure 5-2 shows the use case diagram for the CAD system.

As seen, the actors for the defined use cases are the host truck, the lead truck and the drivers of the trucks. The main use case is the “Drive in Vehicle Following Mode” and it “extends” towards the two other use cases in the nominal functionality, namely the “Lead Vehicle Brakes” and the “Perform Lane Change”. The “extend” relation means that the use cases happen occasionally but not always.
After the nominal use case definition (green frame in Figure 5-2), the safety use cases were added in the UCD. The safety use cases were grouped in two categories: functional safety (yellow frame in Figure 5-2) and operational safety (blue frame in Figure 5-2). The first category includes the cases of sensor failures while the second consists of the cut-in and cut-through.

It is important to mention that the safety uses cases assigned to operational safety are not directly connected with the nominal use cases. To handle cut-in and cut-through scenarios efficiently, the CAD system requires accurate monitoring of the traffic on adjacent lanes. Therefore, an additional use case is defined, namely the “Get traffic status of adjacent lanes”.

The relation between the main nominal use case (“Drive in vehicle following mode”) and the “Get traffic status of adjacent lanes” is “include” as it is always necessary to monitor the traffic on adjacent lanes. However, the “Act in case of cut-in” use case is an extension (extend in Figure 5-2) of the “Get traffic status of adjacent lanes” as it can happen occasionally. Similarly, the “Act in case of Cut through” use case can take place only if a cut-in event has occurred, so it is an extension of the “Act in case of cut-in” safety use case.

An additional relation which can be seen in Figure 5-2 is the “refine”. As described in Chapter 1, the time period that the system can continue its operation in case of a failure determines its level of automation. Therefore, it is important to specify a number of scenarios which can be used in order to validate the performance of the system. The defined scenarios in UCD (see “refine” relation in Figure 5-2) are added as examples. Chapter 7 describes the method of obtaining test cases.

The last task of the requirement analysis is to link the defined requirements with the use cases. The IBM Rhapsody tool allows this by creating the so called “Matrix view” which projects two different data types (in this case requirements and use cases) in rows and columns respectively. This task is performed in order to assure that all the requirements have been defined correctly, therefore for reviewing the requirement analysis [16].

Harmony Step 2. System Functional Analysis (Use-case based)

The objective of this step is to transform functional and functional safety requirements into a description of system operations. In the Harmony profile the functional analysis is use-case based. This means that each use case is translated into a functional model. The term functional
model corresponds to a set of behavioral diagrams. Each functional model is a part of the complete functional architecture description of the system.

There are three diagrams in SysML that can be created to capture the functionality of each use case; the Activity flow diagram, the Sequence diagram and the Statechart diagram. The order that can be followed for the creation of these diagrams depends on the designer choice and the available information of the system. In this project where safety functionality has to be incorporated with the current CAD functionality the following order was selected:

1. Black-box activity flow diagram
2. Sequence diagrams that correspond to each path of the activity diagram. Path is a part of the activity diagram from the “Initial flow” sign till the “Activity final” sign
3. Statechart diagram for each use case

The reason for starting with the activity flow diagram is that it describes “high-level” functionality of a system without including many details. Therefore, it can be easily created from the requirement analysis. The term “high-level” functionality means that operations are defined but their “body” is not yet specified. To define the body of the operations in detail, an additional activity flow diagram can be created per operation including the detailed algorithm and the required calculations. This task either belongs to the technical architecture abstraction level or to the next step of the V-model, the detailed design. As an example, the activity diagram of SUC 1 is shown in Figure 5-3.
Each path in the activity flow diagram forms a different sequence diagram. This means that four different sequence diagrams were created for the SUC 1, named as “nominal scenario”, “fault tolerance scenario”, “comfortable braking” scenario and “collision avoidance” scenario. More details about the functional analysis for the SUC 1 can be found in Chapter 6. Figure 5-4 shows an example of the sequence diagram that corresponds to the “collision avoidance” path.

In the sequence diagrams, the operations were already allocated in specific blocks. This occurred because the block definition diagram (BDD), which includes all the CAD system blocks, is already created. Consequently, even if the activity flow diagram belongs to “black-box” view, the sequence diagram with operations allocated to blocks corresponds to “white-box” view. In case the BDD is not created, all the operations would automatically (from the Harmony profile) be allocated to a block with the name of the use case. Then, the sequence diagram would also be part of the black-box view [14] [16].
The final task of this step is to create the statechart diagram. This diagram includes the states of the system and the transitions between these states. Statechart diagrams are usually part of the decision unit of a system. For the CAD system, the decision unit is the supervisor block as described in Section 3.4. An example of a statechart diagram is shown in Figure 5-5.
Harmony Step 3. Design synthesis

The final step of the Harmony profile is design synthesis. In this step, the high level functionality described in the “black-box” activity flow diagram can be further elaborated by allocating the operations to specific functional blocks or components. To achieve this, a detailed description of the system structure is necessary which is provided by the block definition diagram (BDD). For the safety concept project the BDD includes the following:

- The blocks of the current CAD system, as described in Chapter 3
- Additional blocks and modifications in the decision unit (Supervisor) and the High level Control.

Figure 5-6 shows the block definition diagram of the CAD system as an example. More details about the design synthesis follow in Chapter 6.
This chapter described the approach for the requirement analysis and the functional architecture description of the safety concept. The requirement analysis for a system, which shall comply with the ISO26262 guideline, includes the functional safety and the technical safety requirements. The correct definition of these requirement types is achieved with the introduction of specific attributes.

Next, the SysML and the Harmony profile of IBM Rhapsody were introduced for the functional architecture description of the system. The functional architecture description forms a structured way to describe the high-level functionality of a system on which the detailed design is based.
6. System Design & Implementation

Abstract – In this chapter the detailed design and the implementation of the safety concept is presented. Chapter 5 introduced the functional architecture description and the way to develop it. To extend this architecture towards the next abstraction layers (logical and technical [9]), specific design decisions have to be made. These decisions will actually result in a detailed structure and behavior of the system. The behavior of the system is described for safety use case 1. Next to the design, the implementation of the system is described.

6.1 Introduction

The structure of the current CAD system was described in Chapter 3. The extension of the system with safety related functionality required a number of modifications and additions. One of the modifications was the architectural description (see Chapter 5) of the CAD system and to extend it in a detailed design. According to ISO26262, this design must be modular and extensible.

The terms “modularity” and “extensibility” are introduced in part 6 of ISO26262; “Product development at the software level” [6]. The development of a modular system enables the assignment of the ASIL to each block. The ASIL assignment is important as it connects a safety goal with a specific component. Besides the ASIL assignment, it is often desired to duplicate a number of software blocks on the same or redundant hardware. To duplicate a complete block of the modular design is more efficient than to duplicate part of a block. Decoupling part of a block functionality may result in unexpected errors. Finally, modularity is also important for faster debugging of a system.

The following sections present how the system design for the safety concept was developed. Then, the design decisions for the safety use case 1, “Act in case of WiFi failure” (see Section 5.4.1 Table 5-2) are described in detail. Next to that, a brief description of the implementation is added. The implementation phase did not require a lot of effort, as the existing hardware was used.

6.2 Structure of CAD system

As the safety concept project focuses currently on software design, the main modifications were applied to the supervisor and the high-level control block of the CAD, as shown in Figure 6-1.
Starting with the supervisor block, the goal was to separate the safety related functionality from the nominal functionality in order to achieve modularity. Therefore, two decision units were defined in the supervisor block; the safety decision unit (SDU) and the performance decision unit (PDU). The definition of safety decision unit results in the composition of a block which includes the whole safety-related functionality. Furthermore, the SDU output can be used as input of the performance decision unit. Therefore, the calculations of the PDU can be affected by the SDU output. Figure 6-2 shows the two decision units which are located in the Supervisor block.

![Figure 6-2 The decision units in the supervisor](image)

### 6.2.1 Safety Decision Unit

Moving to the safety decision unit, it is important to consider which blocks this unit shall include. To explain these design decisions a link with the requirement analysis is required. As described in Chapter 5, there are specific attributes assigned to functional safety and technical safety requirements. These attributes are necessary for a requirement analysis that is compatible with ISO26262. Some of the introduced attributes are the following:

- Safety goal
- ASIL level
- Steady safe state of the vehicle (see Table 5-1)
- Safety mechanism including the detection measures, the safety measures and the fault tolerant time interval

The structure of the safety decision unit is influenced by these attributes as follows:

1. The safety mechanism, as stated above, includes the definition of detection measures. The detection measures consists of two categories; detection of the vehicle surroundings and monitoring of system status. Therefore, “Situation Awareness” is necessary for the CAD.

2. Next to situation awareness, the CAD has to make a correct decision for the possible safety actions. This decision has to include a relevant steady safe state. The relevant steady safe states are selected by the designer based on the emergency of the situation. To apply a correct decision, the SDU needs to contain a statechart diagram including the different modes of the SDU. This functionality is assigned to the “Safety Mode Selection” block.
3. The required output of the SDU depends on the safety measures which need to be applied. This output depends on the “situation awareness” and the “safety mode selection” blocks. These calculations are introduced in the “Safety Algorithm” block. The layout of the SDU blocks is shown in Figure 6-3. The “Safety Mode Selection” block is presented with the statechart symbol on the safety decision unit block. Further elaboration of the three blocks of the SDU is given in Section 6.2.2.

![Figure 6-3: The Safety Decision Unit as part of the block definition diagram](image)

### 6.2.1 Performance Decision Unit

The blocks which correspond to the nominal functionality of the CAD system are included in the performance decision unit. Since the nominal performance of the CAD system does not change, this block does not require any further additions.

### 6.2.2 Coupling with existing CAD system

The current CAD system is developed in MATLAB Simulink. Therefore, it was decided to continue the detailed design in MATLAB Simulink and not in IBM Rhapsody. Another option could have been to transfer the current CAD system to IBM Rhapsody, but the complexity of the system made such a transfer impossible within the available time. Moreover, the system would not be able to run real-time on the current CACC platform.

After the introduction of the decision units (SDU and PDU) of the supervisor block, the next task was to couple the new structure with the current CAD system structure. This integration task was challenging since the nominal functionality blocks have proven to be reliable and should therefore not change significantly.

However, this cannot be easily realized as the performance related blocks (grey blocks in Figure 3-3) have to be separated from the safety related blocks (red blocks in Figure 3-3). A possible solution could be to use these signals as feedback. Such a solution is not recommended, as it would require an additional timestep in the calculations. Moreover, it would result in real-time issues in case of running the system with variable time step. So, the separation between performance and safety related blocks is based on the allocation of the longitudinal safety block and the lateral safety block to the safety decision unit as part of the safety algorithm. Next to
that, the performance decision unit includes the “Control Mode Selection” block and the “Split & Merge Support” block. These blocks were part of the supervisor in the current CAD system.

![Figure 6-4 The Supervisor block with the Decision Units](image)

Then, the Safety Decision Unit includes three blocks; the situation awareness, the safety mode selection and the safety algorithm. Figure 6-4 shows the layout of the supervisor block structure.

The **Situation Awareness** block includes the following:
1. The **Object classification block** where the detected vehicles by the sensors are classified as “Most Important Objects”.
2. The **Platoon control** block which is responsible for setting the cruise speed and time gap based on the platoon leader speed, the infrastructure information and possible platoon maneuvers.
3. The **Threat Assessment** block which includes several blocks monitoring the system status and the traffic behavior. More details about this block follow in Section 6.3.

These three subblocks of Situation Awareness provide the required detection measures to the system, as proposed in ISO26262.

The **Safety Mode Selection** block takes as input the SituationAwareness block output and sets the system safety mode. This is performed by introducing the statechart diagrams, which were created per safety use case during the functional architecture description development. Currently, it includes the “WiFiModeSelection” block (described in Section 6.3) and the “GPSModeSelection” block (described in Appendix E to solve SUC 4).

To obtain modular and extensible design, separate statechart diagrams per safety use case are created and not one composite covering all safety use cases. Another advantage of creating separate statechart diagrams is that testing and debugging of the system will become easier. Testing and debugging are crucial because the safety concept development is in early development stage. Furthermore, it was agreed with the project owner that only independent failures will be considered and not combinations of them. However, in a later development stage (after this project) combinations of failures have to be considered. Hence, an additional block must be introduced in order to prioritize the failures.

The last block of the SDU is the **Safety Algorithm**. This block includes the safety actions that shall be performed. These actions depend on the “SafetyModeSelection” output and the current system status. For the longitudinal control of the vehicle, these actions can be either the calculation of a safety distance from the lead vehicle or the calculation of a desired deceleration based on the threat severity. The strategy for selecting the safety action is described in Section 6.3. For the lateral control, the actions include switching between lateral vehicle following mode and lane keeping. To conclude, the safety algorithm output forms an input for the high-level control of the CAD system in case a failure occurs.
As already mentioned, the objective of the new design is to have the safety related functionality “centralized” in the safety decision unit. To achieve this, one more modification had to be applied. This modification is the replacement of the “collision avoidance” block from the high-level control (of the current CAD system) block with new functionality in the safety decision unit.

The section provided an overview of the new CAD system design focusing on the structure of the system. More details about the behavior of the blocks can be found in Section 6.3, where the safety use case 1 (“Act in case of WiFi Failure”) is described.

6.3 SUC 1: Act in case of WiFi Failure

As explained in Chapter 3, the CAD system is very dependent on the wireless communication. This occurs because additional information, which cannot be measured by on-board sensors, is shared through wireless communication. Next to that, the communication is the main enabler of short following distances. So, in case the wireless fails, a critical situation arises (especially when the preceding vehicle issues an emergency brake at the same time with the failure). The following questions will be answered in this section: how the system shall react when communication fails? Is the safety checker (see Section 3.4) enough for avoiding a collision if the lead vehicle brakes? Is the driver able to take over and how fast can this happen? Is it required to add redundancy in the communication signals?

These questions led to the definition of six functional safety requirements relevant with the SUC 1. These requirements including some of their attributes can be seen in Table 6-1.

<table>
<thead>
<tr>
<th>Table 6-1 Functional Safety Requirements for the wireless module</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
</tr>
<tr>
<td>----</td>
</tr>
<tr>
<td>16</td>
</tr>
<tr>
<td>16a</td>
</tr>
<tr>
<td>16b</td>
</tr>
<tr>
<td>20</td>
</tr>
<tr>
<td>20a</td>
</tr>
<tr>
<td>20b</td>
</tr>
</tbody>
</table>

*intervene = return the vehicle to a steady safe state (see Table 5-1) which guarantees a safe situation

Table 6-1 shows that the functional safety requirements which correspond to the SUC 1 are assigned with ASIL D when the lead vehicle brakes. On the other hand, ASIL A-C (depending on the operating conditions) is assigned when the lead vehicle continues with constant speed when communication fails. This means that a communication failure can possibly be harmful, but this does not always apply. Therefore, an adaptive strategy must be defined in the CAD system in order encounter a potentially hazardous failure.

Moreover, it is interesting to see how the current system would act in case of a wireless communication failure. As described in Chapter 3, the current system operates with the CACC controller in its nominal case and uses also the safety checker to increase the distance when a communication failure occurs. However, during the transition towards a higher headway time the situation might still be unsafe in case of braking of the preceding vehicle. Table 6-2 shows how the system would perform in case of communication failure with varying duration. Furthermore, it is assumed that the lead vehicle starts braking at the moment of the failure. It can be seen that there are multiple cases (each case is a line of Table 6-2) that the current system cannot avoid a collision. Collisions also occur when the safety checker is active. The impact
speed for these collisions ranges from 7 to 21 km/h. This is due to the fact that the controller used for the CACC (even when safety checker is active) was not designed to compensate initial condition errors [5].

<table>
<thead>
<tr>
<th>Table 6-2 Current CAD performance when communication fails</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Scenario Attributes</strong></td>
</tr>
<tr>
<td><strong>Failure Duration (s)</strong></td>
</tr>
<tr>
<td>0.1</td>
</tr>
<tr>
<td>0.4</td>
</tr>
<tr>
<td>0.7</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>10</td>
</tr>
</tbody>
</table>

Although this result is expected based on the current CAD design, it is surely not desired. Possible collisions with the above mentioned impact speed could cause serious injuries to drivers or passengers. Therefore, collisions even in extreme cases need to be avoided or at least mitigated. An easy solution to avoid collisions would be to apply maximum braking at the moment that wireless communication fails. Even if this solution sounds as more safe, it is not because of the risk of unintended braking (i.e. when the communication fails, but the lead vehicle does not brake). Unintended braking can also result in a hazardous situation, particularly for the following traffic participants. Besides, it is not desired for the driver to use a system which will brake intensively every time that a failure occurs. This analysis results in the fact that a new strategy is required balancing fail-safety and fault tolerance.

### 6.3.1 Software Design for SUC 1

Now, the system needs to be extended such that SUC 1 will be solved. Moving again to the Safety Decision Unit and especially the Situation Awareness block, it was mentioned in Section 6.2.2 that a “Threat Assessment” block is defined. The functional blocks which belong in the “Threat Assessment” are listed below:

- **Monitor Communication**: This block checks whether the wireless communication with the lead vehicle is active and outputs a flag named “WiFiPacketReceived”. The output flag depends on the reception of the wireless signal and on the host GPS status. The GPS is important for the identification of the vehicles which communicate their position. The CAD system, which operates in the host vehicle, locates the received wireless signals based on the distances in x and y direction. These distances are calculated based on the host position and the received position of the other vehicles. This means that a received signal with no or inaccurate positioning can be considered to be wireless communication failure and this is not desired. Therefore, the SUC 4 (see Appendix E Act in case of GPS inaccuracy) was defined.

- **Break Threat Number (BTN) Calculation**: This block calculates the required acceleration to avoid a collision with the lead vehicle and scales it with respect to the minimum acceleration. The outputs of this block are the following:
  1. The BTN which is the braking ratio (required deceleration upon maximum deceleration) that shows how much of the vehicle braking capacity must be applied in order to avoid a collision. The BTN is calculated assuming that the lead vehicle will start deceleration with its maximum capacity taking into account the vehicle dynamics.
2. The impact speed in case of collision cannot be avoided. The accurate braking path calculation takes into account the following: sensor and actuator delays, the distance between host and lead vehicle, and their relative velocity.

- **Probability of Lead Vehicle braking**: This block calculates the probability of the lead vehicle to brake. The calculation is based on the distance and the relative velocity between the lead and its preceding vehicle (see Figure 6-5). These two values are sent via the wireless communication and at the moment that a communication failure occurs the last stored values are used.

![Figure 6-5 Depiction of host, lead and preceding vehicle](image)

The probability calculation is based on the Poisson process [17]. The Poisson process is a stochastic process which is often chosen for describing critical events. A characteristic term introduced in this process is the event rate $\lambda$, which is the instantaneous event probability if there is knowledge of the system states. The probability $P_d$ based on the time-dependent event rate is formulated below:

$$P_d = \lambda_d[d(t)] \delta t,$$

in which $\delta t$ is the time period in which the probability of an event is calculated and $\lambda_d$ is the event rate, which is dependent on the distance $d(t)$. For this project $\delta t$ corresponds to the fault tolerant time period as introduced in “EmergencyOff_Degradation” (see Page 45). The states, on which the event rate depends, are the distance $d$ and the relative velocity $\dot{d}$. The event rate for the distance $d$ is defined as follows:

$$\lambda_d = \lambda_{d,0}e^{-\beta_d(d-d_{min})},$$

in which $d_{min}$ is the minimum distance after which it is assumed that the lead vehicle will brake, $\lambda_{d,0}$ is the initial event rate defined such as $P_d = 1$ when $d \leq d_{min}$. The parameter $\beta_d$ is configurable and affects the steepness of the exponential equation.

Similar to the probability based on the distance, the probability depending on the relative velocity $\dot{d}$ is calculated. The equations are reconstructed as follows:

$$P_d = \lambda_d[\dot{d}(t)] \delta t$$

$$\lambda_d = \lambda_{d,0}e^{-\beta_d(d-d_{min})}$$

The $d_{min}$ was set 10 m, while the $\dot{d}_{min}$ was selected to be -0.1 m/s. The negative threshold of the relative velocity implies that the lead vehicle approaches its preceding vehicle. The final probability is calculated following the weight function below:

$$P = w \frac{P_d}{(1 - w)P_d},$$

in which $w$ is a weight which ranges from 0 to 1. The value of $w$ is defined as 0.7 meaning that relative velocity between the two vehicles affects more the final probability than their distance. The parameters $\beta_d$ and $\beta_{\dot{d}}$ were set to 3 and 2 respectively based on simulations.

Next to the Situation Awareness block, the Safety Mode Selection block has to be described. As mentioned in Section 6.2.1, the WiFiModeSelection block includes the statechart diagram for SUC 1. An overview of this diagram can be seen in Figure 6-6. The next step is to describe the states, the outputs and the transitions of this diagram.
Starting with the defined states of the SUC 1, the objective was to design a gradual reaction of the CAD system when communication fails. The term “gradual reaction” means that the system must apply smooth braking when the situation is classified as not dangerous (see Table 6-1) and more intensive braking in severe cases.

Each state of the WiFiModeSelection block outputs two parameters; the “WiFiStatus” and the “SteadySafeState” (see Table 5-1). The “WiFiStatus” ranges from 0 to 3 and corresponds to the severity of the system reaction. The “SteadySafeState” addresses where the vehicle will end up. The states of the diagram in Figure 6-6 are described below:

1. **CommCheckOff** (Communication Check Off): This state is introduced to prevent unintended system actions, i.e. braking, when CACC is not active. This means that if the system operates in CC or ACC all the safety actions will be inactive. The output in this state is WiFiStatus = 0 and SteadySafeState = 0 as no action has to be applied.

2. **CommCheckStandby** (Communication Check Standby): This state corresponds to the nominal functionality of the CACC. In the case that CACC is active and no communication failure is detected, the system remains in the standby mode. Similarly
with the state “CommCheckOff” the output in this state is WiFiStatus = 0 and SteadySafeState = 0.

3. **CommCheckError** (Communication Check Error): This is a composite state and includes the following substates:
   a. **EmergencyOff_Degradation**: This is the default state that the system enters when a communication failure occurs. In this state, the system applies the functionality of the safety checker. This functionality can be summarized as smooth degradation by giving as input to the CACC controller the safe distance. This state is often mentioned as “Fault tolerance” as the system will stay there for a predefined time interval even if the situation is critical. The reason is that for failures with short durations (up to 0.5 sec) the smooth degradation can guarantee avoidance of a crash with the lead vehicle. The output in this state is WiFiStatus = 1 and SteadySafeState = 2.
   
   b. **EmergencyOn**: This is a composite state that the system enters in critical situations and includes
      i. **ComfortBraking**: As the name reveals, this state corresponds to a comfortable braking action in which the deceleration and the jerk that the vehicle can apply is limited. The limits which are \(-3.5 \text{ m/s}^2\) for the deceleration and \(-2.5 \text{ m/s}^3\) for the jerk, are defined based on the ISO15622 standard for the Adaptive Cruise Control [7]. It is important to mention that this state sets directly the desired deceleration to the system and not a desired distance. The output in this state is WiFiStatus = 2 and SteadySafeState = 4.
      
      ii. **CollisionAvoidance**: This state forms the most intensive action of the system in order to prevent a collision or to mitigate a collision. In this state the deceleration is not limited so the vehicle can apply braking up to the maximum braking capacity and as fast as its actuation allows (no limit in the jerk as well). The output in this state is WiFiStatus = 3 and SteadySafeState = 6.
   
   c. **SteadySafeState**: This state is entered when the vehicle enters one of the predefined safe states. The actions in this state remain the same as in the “EmergencyOff_Degradation” as the safety checker functionality is active. The output in this state is WiFiStatus = 1 and SteadySafeState = 0.

It can be observed from the description of the states that the gradual reaction of the system is obtained by activating first the EmergencyOff_Degradation, then the ComfortBraking and finally the CollisionAvoidance state. This sequence of states is not always true as the transitions are dependent on the failure duration and the situation severity, as identified from the “Situation Awareness” block. So, it is important to describe when each state is active and why.

The transitions between the states of the WiFiModeSelection block are listed below. The letters projected in Figure 6-6 are used and the terms “OR” and “AND” correspond to the logical operators.

1) **Transition A**: CACC becomes active. This action is performed by the driver.

2) **Transition B**: CACC becomes inactive either when the driver selects another mode “OR” when the driver overrules the system

3) **Transition C**: Wireless communication failure occurs “AND” GPS is initiated “AND” the lead vehicle is still visible (from the camera and/or the radar)

4) **Transition D**: Communication is functional “AND” it is safe to switch back to the CACC controller “OR” the driver overrules “OR” the host vehicle is at standstill.

It is important to check under which conditions it safe to switch back to the CACC controller. During the design of the system, it was identified that switching from a braking action to the nominal CACC controller while the lead vehicle still brakes is not always safe. The reason is that the CACC controller is not designed to handle initial condition errors safely. Hence, three conditions are added in order to guarantee that the system can switch back to the CACC functionality:
a. $e > 0$, where $e$ is the distance error $d_{\text{actual}} - d_{\text{desired}}$.

b. $e_{\text{dot}} > 0$, where $e_{\text{dot}}$ is the relative velocity error $d_{\dot{\text{actual}}} - d_{\dot{\text{desired}}}$ and $d_{\text{dot}}$ is the relative velocity between host and lead vehicle.

c. $d_{\text{ddot}} > 0$, where $d_{\text{ddot}}$ is the relative acceleration between host and lead vehicle.

5) **Transition E**: Failure of duration more than the fault tolerant time “$t_{\text{FT}}$” (0.5 sec) “AND” the BTN is more than the “BTNEmergency” threshold (0.85).

As discussed above, a way to achieve fault tolerance is to stay for a specified period in the **EmergencyOff_Degradation** where smooth braking is applied. The maximum period that the system can stay in this state is vehicle dependent and defined following the next statement: “The lead vehicle starts decelerating till standstill with its maximum braking capacity, namely -6 m/s$^2$, exactly the moment that communication failure occurs. The host vehicle will certainly avoid a collision if the communication restores at $t_{\text{FT}}$ sec while for this period it operated only in the **EmergencyOff_Degradation** state.” The value for this time period, which was obtained from simulations, is 0.5 sec.

However, in the case that the wireless communication does not restore and the lead vehicle performs such severe braking action it is possible to end up with a collision between the two platoon vehicles. Hence, it would be beneficial to know the intention of the lead vehicle. However, the deceleration intention of the lead vehicle is not available without wireless communication and the sensor delays are too long to allow fast system reaction. Therefore, it was decided to use the traffic information in front of the lead vehicle. For this purpose the probability of the lead vehicle to brake, as calculated in the Situation Awareness block, is used in order to allow the transition E to be performed earlier than the $t_{\text{FT}}$ sec. Consequently, the system can perform the transition to the **EmergencyOn** state based on traffic input and the threat assessment reflected in the BTN.

The selected “BTNEmergency” threshold is 0.85 for this transition which means that the host vehicle must brake with 85% of its capacity in order to avoid a collision. This value was selected for the following reasons:

- The required braking when communication is active and the two vehicles drive at 0.3 sec headway time is around 83% of the host vehicle capacity assuming that the lead will apply 100% of its deceleration. The threshold is slightly higher than this value so as to avoid the transition to “EmergencyOn” state when communication is restored and the lead vehicle does not brake.

- It was decided to apply more conservative braking than the calculated in order to compensate for possible sensor delays or vehicle dynamics which are not modelled. Hence, the strategy is that the applied deceleration shall be 15% more than the calculated from the Threat Assessment block. This means that the applied braking will be 100% when the BTN is 0.85 which is used for the transition to the Emergency On state.

6) **Transition F**: If the probability of the lead vehicle to brake is below a threshold “AND” the EmergencyOn state is entered for the first time since the last communication failure.

7) **Transition G**: Default path entered if the conditions of Transition F are not true.

8) **Transition H**: BTN more than the “CollisionAvoidance” threshold which is 0.95 “AND” the distance between the two vehicles decreases “OR” the calculated impact speed exceeds the limit of 20 km/h.

9) **Transition I**: Steady safe state is reached which means for the SUC 1 that the host vehicle is either in standstill “OR” drives at the same speed with the lead “AND” their distance is safe.
10) **Transition J**: BTN is more than the “BTNEmergency” threshold (0.85) “AND” the distance between the two vehicles decreases.

11) **Transition K**: Steady safe state is reached which means for the SUC 1 that host vehicle is either in standstill “OR” drives at the same speed with the lead “AND” their distance is safe.

The analysis of the statechart diagram, which was developed for the SUC1, shows that there are multiple thresholds defined which affect the transitions between the states. These thresholds are configurable and determine how conservative or tolerant the strategy is and how sensitive to false positives. The currently selected values are the ones that prevent collision even in the worst case scenarios. In the future steps of the safety concept development, the requirement might be to allow collision up to certain impact speed. In such a case, the system could be tuned to be less conservative and much more comfortable for the passengers.

### 6.3.2 Hardware Design for SUC 1

The main focus of this assignment is the development of software algorithms which will make the CAD system safer. However, it is important in the complete safety concept to consider the possibility of meeting a safety goal by adding redundancy to the system. Particularly for the wireless communication which plays a crucial role in the CAD functionality the failure rate would decrease significantly.

It is usually desired when selecting a redundant hardware, that the two components have different operating principles in order to avoid common failures. For example, if one camera is added as redundant to an existing, then it is possible that both will fail simultaneously if they are both sensitive to shadows and lighting. In such cases redundancy does not meet the objective or reducing the probability of failure as introduced in ISO26262. According to the functional safety standard ISO26262, for components with ASIL D the failure rate must be less than $10^{-8}$ per hour. The term “components with ASIL D” means that the components are selected to meet a safety goal with ASIL D.

TNO initialized recently an analysis of the hardware requirements for the CAD system. The inputs for this analysis are the nominal use case and the safety uses cases as well. At the moment, several concepts are examined and the investigation is in progress. Particularly, for the wireless communication a redundant system which delivers the other traffic participants intention would be very beneficial. The strategy described above would be still valid with a number of adaptations in the “MonitorCommunication” block.

### 6.4 Implementation

After the detailed design of the CAD system with the safety related functionality, the next step is to implement the functionality on the test vehicle. As described in Chapter 1, the safety concept is initiated as part of the EcoTwin project which aims at the realization of truck platooning on highways. Hence, the development of the system was performed taking the truck model parameters on the simulation level. However, due to unavailability of the two trucks, it was decided to use two Toyota Prius III as test vehicles. Figure 6-7 shows the one of the Toyota Prius III on which the implementation was performed, and the DAF XF truck which was used for the EcoTwin project demonstration. The changes in the configuration of the system due to the different vehicle are mainly related to the vehicle dynamics (time constant of a truck is 0.4 versus 0.1 of a Prius). This results in a slightly different value in Fault tolerant time ($t_{FT}$, as defined on Page 45: 0.45 for a truck versus 0.5 for a Prius).
6.5 Conclusion
Chapter 6 presented the design applied to the CAD system based on the safety concept. The emphasis was on the description of the supervisor block. This block was redesigned in a modular way by separating the nominal functionality from the safety related functionality. Then, the design decisions and the strategy for the SUC1 was presented in detail. Finally, a brief description of the system implementation on the test vehicle was added.
Abstract – After the implementation of the system on the test vehicle it is necessary to verify and validate the CAD system. There is often confusion between the terms verification and validation. Therefore, an explanation of these terms is given in this chapter. Furthermore, the chapter describes the methods applied to verify and validate the safety concept for the CAD system.

7. Introduction

According to the V-model, implementation is followed by the bottom-up side of the development process. The bottom-up phases can be summarized as Verification and Validation (V&V) of a system. These two terms are often confused or misused. The confusion is caused because several engineers think that V&V is one phase. Moreover, there is often misunderstanding at the abstraction level that V&V are addressed. An explanation of these terms is added in Section 7.2.

This chapter aims to show how these methods were applied to the CAD system. The chapter focuses on the system acceptance testing for the SUC 1. A number of test cases has been derived in order to test the system functionality in case of communication failure. The results of these real test are collected and compared with the simulation results, as presented in the chapter.

7.2 Verification and Validation

Several definitions of the terms verification and validation (V&V) exist in the literature [18] [19]. Depending on the nature and the status (i.e. prototype, production) of a system, there are various methods proposed for verification and validation. The following analysis aims to provide understanding to the reader about V&V and the methods to perform V&V.

Verification is the process for determining whether or not the products of a given phase of development fulfill the requirements established during the previous phase [18]. Verification refers to building the "system right" meaning that it is process-oriented [19]. Verification can include two sub-phases; logical and engineering verification.

Logical verification usually consists of completeness check and consistency check. It can be applied to all phases of the V-model top-down side (Requirements, Architecture and Detailed design).

Engineering verification checks for criticality, sensitivity, efficiency, maintainability, portability and reliability. Criticality analysis identifies the most critical software blocks in order to test them thoroughly. Then, sensitivity addresses the ability of the software to gracefully degrade. Efficiency and reliability are performance metrics for components. For example the failure rate of a hardware component shows the reliability of the component. Afterwards, portability is a similar term with modularity. A portable software block can be easily used in different processing units. Similarly, a portable hardware block can be reused in different systems. Finally, maintainability corresponds to systems that are on the product level. These systems have to be easily maintained, so maintainability is important verification metric. More details about these terms can be found in [19].

Validation is the evaluation of the capability of the delivered system to meet the customer objective. Validation focuses on the accurate performance with respect to the customer needs and refers to building the “right system”. This means that validation is result-oriented. Hence, validation focuses mainly on system performance and usability. The term usability addresses how the user experiences the system. The method for applying system validation is the testing of specific scenarios (test cases) derived from the requirement analysis.
7.3 CAD system V & V

After the definition of verification and validation, it is important to present how these phases were executed for the CAD system.

7.3.1 Unit and integration testing

The CAD system is currently at prototype level. Therefore, it cannot be verified as a system entering production. For example, maintainability and portability cannot be addressed for the complete CAD system but only at component level. Next to that, the verification of the hardware was out of scope for this project, as existing and already verified hardware blocks have been used. Hence, the verification was mainly focused on the software blocks which were added during this project. The applied methods for the CAD system verification are summarized below:

1. Requirement analysis
   A review of the requirements and check for consistency was performed.
2. Architecture and Design
   The software blocks created in these phases were verified by simulations, model-checking and testing on the real setup. The simulations were performed in MATLAB Simulink. MATLAB also includes a diagnostic tool performing model-checking (i.e. looking for deadlocks, divisions by zero, etc.).

The verification of the CAD system for consistency was limited to reviewing by senior scientists within TNO. The complete verification of the system for completeness and consistency has to be performed at a later stage, when the whole safety concept is developed. MATLAB supports model verification with the “Simulink Design Verifier” tool.

7.3.2 System acceptance testing

Next to unit and integration testing, the CAD system must be validated. System validation or system acceptance testing has to be performed on a real vehicle. Testing at this level should prove that the system performs as required by the customer (TNO for this project). Moreover, it will show how driver and passengers experience the CAD system.

The nominal functionality of the CAD has already been tested. In this project, the objective was to perform system acceptance testing for the case that communication fails. Particularly, it was necessary to evaluate the host vehicle behavior in a two-vehicle platoon when communication fails. To achieve this, a test plan was created, as described in the next steps:

1. Assumptions
   First, several assumptions have to made for the system and vehicle operating conditions. In the system validation phase of the CAD system, it is assumed that both vehicles have the same braking capacity (up to \(-6 \text{ m/s}^2\)) and they drive with constant speed at 0.3 sec headway time.

2. Safety use case parameters
   Next to the assumptions, it is important to identify which parameters can vary when a communication failure occurs. These parameters, which are obtained from the design of the systems described in Chapter 6, are summarized below:
   a. The communication failure duration (\(t_{\text{fail}}\))
      The duration of the failure directly affects the host truck response. Particularly, it determines whether the vehicle will stay in the “Fault Tolerance” state or it will enter the “Emergency On” state.
   b. The lead truck deceleration start time (\(t_{\text{lead.br.start}}\))
      When a communication failure occurs, the behavior of the lead vehicle is crucial. The worst case scenario is when the lead vehicle brakes exactly when a communication failure occurs.
   c. The lead truck deceleration value (\(a_{\text{lead}}\))
      The lead vehicle deceleration value affects the host vehicle response as well. The values selected for the tests range from \(-6 \text{ m/s}^2\) to \(0 \text{ m/s}^2\).
   d. The lead truck braking probability (\(P\))
      This parameter also influences the transition from “Fault tolerant” state to “Emergency” state and the activation of “Collision Avoidance”.

3. Test cases
After the definition of the safety use case parameters, the next step is the test case derivation. Test cases are usually derived from the use cases (see Use case diagram Section 5.4.1) of the system. The number of test cases depends on the number of combinations of the safety use case parameters and the criticality of the system. For this project that includes safety related functionality the criticality is high. This means that the test cases must include all critical scenarios (if not all possible scenarios).

The next step is to develop a structured approach to derive the critical test cases for SUC 1. These test cases should cover all different responses that the host vehicle can perform. For example, it is expected that the host vehicle will react with less braking when communication fails for 0.1 sec than in case it fails for 2 sec. The selected structured approach to define test cases is based on the mapping of the safety use case parameters, as shown in Figure 7-1.

Each row of Figure 7-1(from 1 to 4) corresponds to a safety use case parameter. Each column (from A to D) corresponds to a parameter threshold. These thresholds were selected based on the design of the Safety Decision Unit (see Section 6.3.1). The critical value(s), for which a change in the system reaction is expected, is set in mapping of the parameters.

Therefore, a test case can be derived from a combination of the parameter values. For example, the combination “1A 2C 3B 4A” results in the test case 1.1 with description:”WiFi Failure for 0.1 sec and lead vehicle brakes at t=0 sec with \( a_{lead} = -6 \text{ m/s}^2 \) and \( P=0 \)” The words in bolded format correspond to the safety use case parameters. The detailed list with the 23 defined test cases can be found in Appendix F.

### 7.3.3 Results

The system validation has been performed based on the defined test cases at two levels; simulation and on real testing. Part of the results of the system validation phase is shown in Table 7-1. It can be seen that with the new design there is no collision between the host and the lead vehicle. A comparison between Table 7-1 and Table 6-2 shows that there is significant improvement in terms of safety. In Table 6-2, it was shown that in four cases collision occurred when the safety checker was active and in five cases when the CACC controller was active.
With the new design, collision can be avoided in all these cases as shown in Table 7-1. This result is evaluated from simulations and real tests.

As stated in Section 6.3 the system response in case of communication failure must be a trade-off between fail-safety and fault-tolerance. The system must respond adaptively depending on the lead vehicle braking action. To validate this performance a number of test cases was derived. The last two lines of Table 7-1 include two of these test cases. It is shown that the host vehicle does not brake heavily with the new design, as the minimum deceleration reaches the value of -3.5m/s² in the simulation results (see last two rows of Table 7-1).

However, in the real test the system response was not exactly the same. The reasons that more conservative braking was applied were the sensor delays and inaccuracy. According to the developed strategy, the measurement of the relative velocity affects the transition from “ComfortBraking” to “CollisionAvoidance” (see transition H in Figure 6-6). After analyzing the test data, it was observed that this measurement was inaccurate and close to the selected threshold value. Therefore, the transition H was wrongly enabled and the “CollisionAvoidance” state was entered allowing more severe braking.

To conclude, the system validation has been performed successfully. In most of the cases, similar behavior has been observed between simulations and real tests. In few cases that the system did not perform as expected, the reasons were the following:

- a. Braking action of the test vehicle
  The test vehicle could not always apply continuous braking from 80km/h till standstill. This seems to be a vehicle issue or feature already known in TNO.
- b. Radar and camera performance
  It is important to mention that the distance between the host and the lead vehicle was increased by 15m for safety reasons during the real testing. This longer distance can also be a factor that affected the measurement inaccuracy.

A complete overview of the system validation and a comparison between real and simulation data can be found in the internal document of TNO [20]. It shall be mentioned that a script has been developed that automatically runs the simulation and the real test results.

<table>
<thead>
<tr>
<th>Test case parameters</th>
<th>System Validation Results</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Simulations</td>
</tr>
<tr>
<td></td>
<td>Host vehicle min deceleration (m/s²)</td>
</tr>
<tr>
<td>Failure Duration (s)</td>
<td>Collision</td>
</tr>
<tr>
<td>Lead vehicle Acceleration (m/s²)</td>
<td>Collision</td>
</tr>
<tr>
<td>0.1</td>
<td>-6</td>
</tr>
<tr>
<td>0.4</td>
<td>-6</td>
</tr>
<tr>
<td>0.7</td>
<td>-6</td>
</tr>
<tr>
<td>1</td>
<td>-6</td>
</tr>
<tr>
<td>10</td>
<td>-4</td>
</tr>
<tr>
<td>10</td>
<td>-6</td>
</tr>
<tr>
<td>0.1</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
</tr>
</tbody>
</table>
7.4 Conclusion

This chapter presented the verification and validation phases in a system development. The verification of a system is usually process-oriented, while the validation is result-oriented. Next, the chapter described how the way that verification and validation phases were applied to this project. For the system validation, a structured approach has been defined for the derivation of test cases. Following this approach, 23 test cases have been investigated at simulation level and real testing. The results showed that the level of safety increased with the new design.
8. Project Management

Abstract — It is widely known that successful results in a project do not depend only on the skills of the project engineers. The management of the project is also important, as it guarantees that a correct process is followed. This chapter includes two parts. First, it lists the necessary work-products for efficient management of a project. Then, it presents how the management of this project was executed.

8.1 Introduction

Several aspects have to be considered for the execution of a development project. For example, the number of people that will contribute, the technical competencies of those people, the required budget, the customer needs etc. The discipline that handles these aspects is project management. A formal definition of project management is added below:

“Project management (PM) is the discipline of carefully planning, organizing, motivating and controlling resources to achieve specific goals [21]”

From the definition of PM it can be concluded that several work-products have to be created in order to manage a project efficiently. These work-products are the following:

1. Project proposal
   This document has to be created before the initialization of a project. It shall define and justify the needs of a project. Moreover, it shall ensure that the proposed project is aligned with a strategic or business plan of the company. Next to these, it must specify the outcome and the benefits of this outcome. Based on this document, the management decides whether the project will start and makes an initial estimation of the project budget.

2. Project definition document
   This document has to be created at the start of a project. It must clearly state the scope and the goals of a project. Next to these, it has to specify the desired deliverables. Finally, it must include an initial plan and an initial risk analysis.

3. Detailed project plan
   The detailed plan shall be created after the initialization of the project when the project arrangements (resources, budget, etc.) have been performed. It must include precise task distribution, resource allocation and a number of milestones. This document has to be communicated within the team and regularly updated.

4. Progress reports
   The project reports have to be delivered at predefined time slots. These reports can also be aligned with project milestones. In any case, the progress reports must include possible issues occurred, preliminary results and the next steps that have to be followed. Moreover, an update of the project plan and update of the risk analysis can be included. Next, the budget status must be presented.

5. Post-project Review
   This document has to be created after the closure of a project. It must include an analysis of the achieved outcome. Moreover, it is a measure of the degree to which the benefits have been achieved. Moreover, it can include recommendations for potential improvements and extensions of a project.

It shall be mentioned that the number of work-products and their content often vary between companies. This occurs because of the different nature of the projects. For example, a project in a research company requires more effort in the project proposal document than a project in a production company. Due to these customizations, the companies usually have templates for the described work-products.
8.2 Project Management for the safety concept project

After the formal definition of project management and the overview of the required work-products, it is interesting to see how the safety concept project was managed. It must be clarified that this project (“Towards a safety concept for CAD”) is initiated within the EcoTwin as described in Chapter 2. Figure 8-1 shows the initial work-breakdown for this project. Based on the task decomposition, the work package 1 (WP1 in Figure 8-1) was assigned to the PDEng trainee.

![Figure 8-1 Work-breakdown for safety concept](figure)

Since this project is part of a large, already initiated project, a project proposal document was created in an earlier stage (before this assignment). The project proposal document included the strategy of TNO, as described in the business roadmap (see Section 2.2).

After the project initialization, a project definition document was created including the following:

- a. General information, i.e. title, department and project time frame (from 05.01.2015 – 30.09.2015)
- b. Project scope and deliverables, as presented in Sections 2.4 and 2.5 respectively.
- c. Team structure and meeting management as discussed in Section 8.2.1.
- d. Project plan and risk analysis, described in Section 8.2.2 and 8.2.3 respectively.

It is important to mention that budget management was not in the scope of the project.

8.2.1 Team structure and Meeting management

The team members that were involved in this project and their role are listed below:

1. **Project owner and company supervisor**: Ir. Ellen van Nunen
   Role: Support, guidance and supervision of this assignment and management of the safety concept.

2. **PDEng Trainee**: Dimitrios Tzempetzis
   Role: Responsible for the approach towards the safety concept and the integration of the defined work packages (see Figure 8-1)

3. **TU/e supervisor**: Prof. Mark van den Brand
   Role: Support, guidance and supervision of this assignment focusing on the selected process.

4. **MSc student**: Gerald Koudijs
   Role: Development of control methods for collision avoidance in CAD (see WP3 in Figure 8-1).

5. **Senior Scientist**: Jeroen Ploeg
   Role: Advisory
6. **Scientist**: Laurie Bax  
   Role: Focus on scenario detection (see WP2 in Figure 8-1)

7. **System Engineer**: Per Ambrosiussen  
   Role: Focus on sensor selection for CAD (part of WP4 in Figure 8-1)

To obtain efficient communication between the project members a meeting schedule has been created. Weekly progress meetings took place in TNO between the PDEng trainee, the Company supervisor and the MSc student. Then, biweekly meetings were arranged between the team members located in TNO (safety concept meetings). The purpose of these meetings was the presentation of results and exchange of ideas. Finally, biweekly meetings took place with the TU/e supervisor focusing on the process followed during the project.

### 8.2.2 Project Plan

An initial project plan was created in the project definition document based on the phases of the V-model. It can be realized that a project plan cannot be accurate when defined in the initialization phase of a project. Therefore, this plan was regularly updated and detailed during the project. The frequency of these updates was synchronized with the timing of the progress reports, namely every six weeks. The timeline of the project plan can be seen in Figure 8-2.

![Figure 8-2 Timeline of the project plan](image)

### 8.2.3 Risk Analysis

Risk analysis is also crucial for the management of a project. The risk analysis was performed initially in the project definition document including the following:

- **Risks**
- Impact of the risks which was selected to be low, medium or high.
- Risk response planning. The risk response planning includes the actions that should be taken in order to limit the impact of a risk. It can be realized that these actions can vary depending on the impact. A possible action is to avoid a risk by eliminating its cause. An alternative action is to mitigate a risk by reducing its impact. Then, for risks with low or medium impact possible actions could be to accept them or transfer them to another party (person, company, etc.).

The identified risks were tracked during the project period. This was obtained by assigning a new impact to risks every two months.

<table>
<thead>
<tr>
<th>Table 8.1– Risk analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real implementation</td>
</tr>
<tr>
<td>Number of Iterations</td>
</tr>
<tr>
<td>Work breakdown</td>
</tr>
<tr>
<td>Wrong assumptions</td>
</tr>
<tr>
<td>Project focus</td>
</tr>
</tbody>
</table>
Table 8-1 shows the risks and their impact on different stages of the project. The identified risks in the project definition project were five; real implementation of the design on the test vehicle, the number of iterations within the 9-month project, work breakdown between the team members, correctness of the assumptions and project focus. A detailed description of the risk impacts and the risk response planning can be found in the project definition document.

8.3 Conclusion

The chapter presented how the project was managed. As the project was part of a bigger one, namely the EcoTwin, it was not necessary to create all the work-products that are usually required for the management. The main goal of the project management was to organize the work in this 9-month project. Moreover, it was necessary for the clear formulation of the project scope and deliverables.
9. Conclusions

Abstract — The conclusions of the project are shown in this chapter. The focus is on the benefit that TNO obtained from this project. Next, recommendations for further extension of the project are presented. Afterwards, the lessons learned by the trainee are mentioned. Finally, the project retrospective section forms the reflection part of the project.

9.1 Conclusions
An approach towards a safety concept for cooperative automated driving systems was proposed, based on system engineering methodology and ISO26262 guideline. This approach was applied to extend the CAD system with safety related functionality aiming at a system with automation level 4. The outcome after following the approach was a modular and extensible design at the supervisor level of the CAD system.

Next to the defined approach, the extension of the CAD system has improved safety in case of failures. A solution for the case of wireless communication failure has been developed and extensively tested. Moreover, an algorithm to handle GPS inaccuracy has been proposed. Next to these, an initial functional architecture description for the cases of unexpected cut-in and cut-through has been developed.

The testing of the system with real vehicles was successful. It was observed a collision could be prevented even in extreme scenarios. Besides, further development is required for the CAD system to reach automation level 4, which is the objective of TNO. This will be achieved when all the safety goals (defined in HARA) will be covered.

To conclude, the proposed approach provides insight into safety vocabulary and forms a basis for extensions on the CAD system. The options for possible extensions and future work are described in Section 9.2.

9.2 Recommendations
After the execution of the project there are several recommendations for extension and future work, as listed below:

1) TNO could follow and further improve the proposed approach. A structured approach will result in consistent designs for all projects. Moreover, the development of projects related to automated driving and safety will be more efficient. Next, the certification of a system will be acquired easier.

2) TNO could combine the approach with the architecture framework developed in parallel [22]. The developed framework is generic and addresses the necessary steps for the development of a new system. However, for the extension of an existing system several steps can be merged or omitted depending on the focus of the extension. It could be useful to compare the steps of the developed approach with those of the architecture framework.

Next to the combination, TNO should create a team working on the extension of the safety concept for the CAD system. This team shall include scientists and/or engineers from all the groups of the IVS department; the Cooperative Vehicle Safety (CVS) group, the Vehicle Dynamics Estimation & Control (VDEC) group and the Safety Assessment & Methodologies (SAM).

3) If TNO aims to develop a functionally and operationally safe CAD system on a product level, the following should apply:

b. Explore the option of a completely new architecture using the current as reference. A new architecture would result in the development of a complete system incorporating functionality, functional safety and operational safety.

c. The sensor concept has to be reconsidered. In the current implementation, the CAD system performance is dependent on the sensor characteristics. TNO will have direct benefit from a clear specification of sensor requirements considering nominal functionality and safety (functional and operational).

4) Apply the new or the current architecture in a tool supporting SysML (IBM Rhapsody could be an option). After performing this, executable models can be created and verified in the new software tool. In parallel, component design can be still developed in MATLAB and then linked to SysML.

5) TNO could establish further cooperation and partnerships with companies developing sensors. For automated driving systems it is mandatory to perform extensive testing in order to prove performance and safety. Moreover, it is beneficial to explore the possibility of connecting various sensor sets (different sensing concepts as well) to the CAD system. In such case, the CAD system will be proven to be modular and flexible to customizations. EcoTwin (Figure 9-1) sounds as a good project to attract partners.

![EcoTwin Truck](image)

Figure 9-1 First demonstration of EcoTwin project

9.3 Lessons Learned

After the project execution, multiple lessons learned can be identified:

1. Knowledge development on safety systems design was achieved. Moreover, better insight into the ISO26262 guideline was managed and its applicability in automated driving systems was explored.
2. Knowledge development on automated driving systems was achieved. Working as system engineer with the CAD system, you learn about sensor concepts, control methods and actuating techniques.

3. Applying a system development process for extension of a complex system was also a challenge.

4. Designing a (sub)system which can be tested real-time was another learning point.

9.4 Project Retrospective

The project topic is at the edge of automotive technology. Automated driving and safety are interesting topics in which I am glad to be involved. Particularly, the assignment required the competencies that the PDEng program aims to develop; system engineering, structured way of working, non-technical skills (management, presentations, meeting skills). Therefore, I am satisfied that I had the chance to apply these competencies in a competitive environment.

The guidance and supervision of the project was efficient and effective. The experience of both supervisors (in their domains) contributed positively to the project execution. The company supervision was regular and the cooperation was efficient. The meetings with the TU/e supervisor were biweekly. Hence, it was more challenging to keep the professor up-to-date, particularly when I worked at a very technical level. More effort was required in the preparation of those meetings. In any case, the critical feedback of the TU/e supervisor was appreciated.

The close cooperation with the MSc student who worked in control methods was efficient and productive. Several discussions resulted in working solutions in the proposed design.

Reflecting on the team structure, I would mention that working in a larger team from an early stage would be preferable. More people involved in the first half of the project would result in parallel execution of more safety use cases and possibly a better approach. Moreover, the cooperation between the VD group and the CVS could be closer. Reconsidering the same problems or re-implementing similar solutions can be avoided if one team will be created.

Finally, reflecting on the 2-year ASD program I think the program made me more “mature” professionally and personally. Executing several projects in the automotive domain, attending technical and non-technical workshops and working in a multinational environment were the key factors of the PDEng that contributed to my development.
Bibliography

References

[20] Test Results and Simulation Results_SUC1, TNO, 2015.
Additional Reading

Appendix A

Hardware blocks in CAD system

The hardware components used in the current CAD system are described below and shown in Figure A-1.

Sensors

The sensors of the vehicle and the CAD provide to the system the necessary input for the traffic and vehicle status. The different sensor types and the parameters that they measure are listed below:

- In-vehicle sensors measuring a.o the longitudinal speed of the vehicle, the longitudinal acceleration, the steering angle and the heading angle rate.
- Sensors monitoring the traffic in forward direction
  - **Radar** placed in the front of the vehicle measuring the distance $d$ [m], the relative velocity $ddot$ [m/s] and the heading angle $\alpha$ [rad] between the vehicle and its preceding object. The radar can measure objects within a range of 120m and its field of view is 34° (+/-17°).
  - **Camera** which is placed on the windshield measuring the distance $d$ [m], the relative velocity $ddot$ [m/s] and the heading angle $\alpha$ [rad]. The range of the camera can be up to 120m but for accurate results the detection shall not exceed the 50m. The angle field of view of the camera is 76° (+/-38°).

Positioning system (GPS)

The GPS receiver has an update rate of 2Hz and is used as an input for the tracking algorithm. Moreover, the position obtained by the GPS is communicated through the wireless module and used for the accurate localization of other traffic participants.

Wireless module (WiFi)

The WiFi device operates according to the IEEE 802.11a standard in ad-hoc mode with a frequency of 25Hz. This device enables the communication of the vehicle status (position, acceleration, etc) with other equipped vehicles. Due to the high frequency of the communicated signal, the intention of the lead vehicle can initiate a fast response of the following vehicle. This is the reason why the wireless module is the enabler of the platooning with short inter-vehicle time distances (~0.3sec). Hence, the robust functioning of the wireless module is crucial for the performance of the system when short distances are realized.

Actuators

Many of the actuators of the CAD system are hardware components which are already present in the vehicle such as the acceleration, the braking and the steering actuator. It is important to mention that the actuators introduce a delay in the system response. This delay has been measured to be 0.2 sec for the test vehicle and it is assumed that it is the same for the truck.
**Software blocks in CAD system**

**Sensor-processing unit**

The sensor-processing unit takes as input the sensor data and gives as output the tracking output for the host vehicle, the target vehicle (or lead vehicle) and the lanes of the highway. In particular, the host tracking algorithm is developed for the estimation of the host vehicle states (position, speed, heading angle etc). The measurement data are obtained from the GPS and the in-vehicle sensors. Then, the target tracking block is used for the estimation of the surrounding vehicle states (position, speed, heading angle etc). The measurement data for the target tracking are obtained from the camera, the radar and the received communicated signal. The data from these different sources are clustered and used as input for the classification of the surrounding traffic participants. This classification, referred to also as “Object classification”, is based on the relative position and the relative speed between the surrounding vehicles and the host vehicle. The term MIO (Most Important Object) is used for the classified objects. This results in the naming of the objects as shown in Figure A-2. It is important to notice that for a platoon with two vehicles the accurate detection and classification of MIOA is mainly required. Apart
from Host tracking and Target tracking. Lane tracking is performed based on the detection of lanes on the highway. The lane detection relies currently on the Mobileye camera.

Figure A-2 Object classification in CAD system

**High-level control**

The high-level control of the CAD forms the actual functionality of the system which is the vehicle longitudinal and lateral vehicle following. The longitudinal control or “CACC control” is defined based on the spacing policy

\[ d_r = r + h \cdot v_{host}. \]  

(1)

Where \( d_r \) is the desired distance, \( r \) is the standstill distance with respect to the lead vehicle, \( h \) is the desired headway time and \( v_{host} \) is the current velocity of the host vehicle.

The high-level control system takes as inputs first the user modes such as the desired headway time and then the sensor signals in order to calculate the required acceleration/deceleration and steering angle for the vehicle actuators. The detailed design of the high-level control can be found in [22].

The lateral controller is also part of the high-level control and consists of two different modes; the lateral vehicle following and the lane keeping. Both modes can be selected by the user. In the case of absence of a lead vehicle, the system automatically switches to lane keeping if the lanes are detected. If none of the two lateral control modes can be applied, then the system warns the driver to take over the steering of the vehicle.

**Low-level Control**

The required acceleration and steering angle are calculated in the high-level control. The next step is to convert these signals in order to be compatible with the vehicle actuators. For this purpose, TNO has developed a gateway which is the interface between the vehicle systems and the real-time CACC platform. This gateway runs at 100Hz and apart from the conversion of the control signals, it performs the initial processing of the sensor data and a number of safety checks on the component level (sensors, real-time platform).
Appendix B

Functional Safety ISO26262 standard

The ISO26262 standard provides the guidelines to apply functional safety in automotive applications. This standard includes several steps which are applicable throughout the development of all automotive electronic and electrical safety-related systems. All these systems shall be compatible with the standard in order to enter the production phase and therefore the market.

ISO26262 forms an extension of the ISO61508 functional safety standard emphasizing on the automotive sector. Similar to its parent, it is a risk-based safety standard. It defines the safety measures needed to:

3. avoid or control systematic failures
4. detect or control random hardware failures, or mitigate their effects.

It should be noted that the ISO26262 standard was not designed specifically for automated driving systems. Although the ISO defines the Automotive Severity Integrity Levels, it assumes that the driver is actively monitoring the system and takes over when required. For systems with automation level 4 (such as the CAD) or 5, the driver is no longer backup. Nevertheless, the ISO26262 standard remains the generic guideline that should be taken into account for the certification of those systems.

The ISO26262 consists of 10 parts which are briefly described below:

1. **Vocabulary**
   The first part includes the terms and definitions which are necessary to know before the steps of the ISO are read.

2. **Management of functional safety**
   During this part a plan with task distribution is created. Usually a safety team is formed in order to execute the created safety plan.

3. **Concept phase**
   In this part the goal is to initialize the functional safety analysis by performing the following tasks.
   
   a. **Item definition**
      A document that describes the item (the CAD system). This document includes a nominal use case describing the behavior of the vehicle when the item operates.
   
   b. **Hazard Analysis and Risk Assessment (HARA)**
      The goal of this step is to identify and classify the hazards of the item and specify the safety goals. HARA is performed on the vehicle level and the classification of the hazards is based on **Severity**, **Exposure** and **Controllability**. The final metric is called Automotive Safety Integrity Level (ASIL). ASIL ranges from A to D, where level D dictates the highest integrity requirements on the product. The CAD system can be classified as an ASIL D system.
   
   c. **Functional Safety Concept (FSC)**
      The main goal from the FSC is to derive the functional safety requirements. The inputs for this step are the safety goals as obtained from the HARA and the preliminary architecture of the item. The functional safety requirements are defined at the system level. The specific attributes that are linked with these requirements will be described in Chapter 4.

4. **Product development at the system level**
   The main objective of this part is to develop the technical safety requirements. These requirements are also on the system level and form a refinement of the functional safety requirements. This part describes also the system validation.

5. **Product development at the hardware level**
This part of the ISO26262 specifies the requirements at the hardware level. It suggests hardware design guidelines and architecture metrics. Moreover, it includes an evaluation of possible violation of the safety goals due to random hardware failure.

6. **Product development at the software level**
   The objective of this part is to plan the functional safety activities for the software development. Software requirements need to be derived, and the architecture, design and implementation of the software components needs to be specified. Moreover, the tools and the methods that are going to be used need to be selected during these sub-phases.

7. **Production and operation**
   The goal of this part is to define the requirements for the production of safety related projects. Moreover a production plan and a production control plan are created during this step.

8. **Supporting processes**
   Several agreements and distribution of responsibilities have to be performed during this part. The different parties or companies that contribute in the development of a system have to agree how the ISO standard guidelines will be followed. Moreover, several verification methods are introduced for the complete system and for every work-product.

9. **Automotive Safety Integrity Level (ASIL)-oriented and safety-oriented analysis**
   The objective of this part is to provide rules and guidance for decomposing safety requirements into redundant ones.

10. **Guideline on ISO 26262**
    The last part of the ISO26262 provides a guideline for how the several work products of the previous steps can be created.

Figure B-1 shows a number of ISO26262 work-products covering three abstraction levels; vehicle, system and component level. Particularly, the item definition, the HARA with safety goals and the functional safety concept are presented. These work-products belong to the concept phase (part 3 of the ISO26262) and result in the development of functional safety requirements (FSR). Next to the FSR, Figure B-1 shows the Technical Safety Requirement Specification (TSRS). The TSRS belongs to the part 4 of ISO26262; product development on the system level. It can be seen that both FSR and TSRS are defined at system level.
Figure B-1 ISO26262 work products
Appendix C

System development process – Waterfall model

The development of a new system or the extension of an existing system which incorporates software and hardware components is often challenging even for experienced engineers. There are examples of developed systems which have limited or different functionality than the required by the stakeholders and the business. There are several reasons that can lead to this situation such as defining no or very generic requirements, starting directly with the implementation of the system, relying just on the effort of the developers, skipping regulations and laws, etc. Consequently, many companies have introduced a development process that can be followed in all the projects and can guarantee repeatable and successful results.

There are many available development processes which usually cover the following tasks.

- System conceptualization
- System requirements and benefits analysis
- Project adoption and project scoping
- System design
- Specification of software requirements
- Architectural design
- Detailed design
- Unit development
- Software integration & testing
- System integration & testing
- Installation at site
- Site testing and acceptance
- Training and documentation
- Implementation
- Maintenance

Waterfall model

The waterfall approach which is shown in Figure C-1, is the earliest method of structured system is a top-down approach which includes five main steps. These steps are executed sequentially, so each phase is entered only when the previous is completely finished. This development process has a number of advantages, however the drawbacks of using it outweigh them.

The advantages of the waterfall process are listed below.
The process is simple, easy to understand and use for small projects where requirements are clearly specified.

- It is easy to manage due to the rigidity of the model – each phase has specific deliverables and a review process.
- In this model phases are processed and completed one at a time. Phases do not overlap.

The disadvantages of this sequential process are summarized below.
1. There is no version (software or system) released until the late stage of implementation
2. There is not the flexibility to update the system functionality when the system is in testing phase.
3. It is not suitable for complex systems or projects in which the risk of changing requirements is relatively high
4. The cost of adopting new customer objectives is high due to the lack of flexibility of the model.
Appendix D

System modelling language – SysML

SysML includes several diagrams for the description of an architecture as shown in Figure D. The most important diagrams are listed below:

Structural diagrams
The structural diagrams are the block definition diagram (BDD) and the internal block diagram (IBD).
1. BDD depicts the blocks of the system from vehicle level to component level. A block in SysML corresponds to a class in UML (or in software engineering) [12].
2. The IBD includes communication flows between the software and hardware blocks. A number of ports are defined for each block and the flows of data or other means (i.e. liquid or electricity) are introduced to connect these ports.

Behavioral diagrams
The behavior of the system is described in activity flow diagrams, sequence diagrams and state flow diagrams.
1. An activity flow diagram includes actions or operations and decisions that are defined in a system. If it is not specified which action will be performed in which functional block, then the diagram is called black-box activity diagram. Otherwise it is referred as white-box activity diagram.
2. The sequence diagram, as the name shows, describes a sequence of events or actions that were previously defined in an activity flow diagram. Therefore, a sequence diagram corresponds to a path of an activity flow diagram; namely a branch from the “initial flow” till the “activity final”. The additional value of this diagram is that it usually requires the allocation of each action or event to a specific block of the system.
3. The state flow diagram puts the information of activity and sequence diagrams into the context of system states. This diagram forms the decision unit of the system and it is often defined on the “supervisor” or “governor” level of the system.

Figure D-1 SysML Diagrams
Appendix E

**SUC 4: Act in case of GPS inaccuracy**

This section describes the role of the GPS in the CAD system and why an inaccuracy can be hazardous. The CAD system uses GPS data to position the host vehicle. Moreover, it receives through wireless communication the position of the surrounding vehicles obtained from their own GPS. As described in Appendix A, the communicated GPS positions are clustered with the camera and radar positions. These clusters are used as input for the classification of the surrounding traffic participants (see Figure A-2).

The CAD system shall distinguish from which sensor each object (surrounding vehicle) is detected. Therefore, the parameter “dataMode” is defined and assigned to each object. Table E-1 shows the different values of the parameter “dataMode”.

<table>
<thead>
<tr>
<th>dataMode</th>
<th>Sensor types and/or WiFi</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>No Object is detected</td>
</tr>
<tr>
<td>1</td>
<td>WiFi</td>
</tr>
<tr>
<td>2</td>
<td>Camera</td>
</tr>
<tr>
<td>3</td>
<td>WiFi + Camera</td>
</tr>
<tr>
<td>4</td>
<td>Radar</td>
</tr>
<tr>
<td>5</td>
<td>WiFi + Radar</td>
</tr>
<tr>
<td>6</td>
<td>Camera + Radar</td>
</tr>
<tr>
<td>7</td>
<td>WiFi + Radar + Camera</td>
</tr>
</tbody>
</table>

As described in Chapter 6, a wireless communication failure can be potentially hazardous. Such a failure results is realized from the CAD system a change of the lead vehicle dataMode from odd to even. However, a change in the dataMode does not necessarily mean that a wireless signal has been lost. It can also be caused by an inaccurate GPS position.

An inaccurate position affects the clustering of the detected objects and therefore the classification. Figure E-1 presents how an inaccurate GPS position of the lead vehicle is realized by the CAD system of the host vehicle. It is assumed that at time step “t” the lead vehicle is detected by the camera and the radar at the position of the blue truck (“Lead” in Figure E-1). However, the communicated position results in a wrong positioning of the lead as shown with the “Lead GPS” truck. Therefore, the host vehicle detects wrongly two vehicles even if there is only one. Furthermore, it assigns to the lead vehicle (Lead in Figure E-1) dataMode 7 as it detects it with radar, camera and wireless communication.

If an inaccurate GPS position is sent from the lead vehicle at time step “t+1” the classification in the host vehicle is affected. First, it can be assumed that the lead vehicle is still detected by the camera and the radar at the position of the blue truck (“Lead”) in Figure E-1. However, the communicated position results in a wrong positioning of the lead as shown with the “Lead GPS” truck. Therefore, the host vehicle detects wrongly two vehicles even if there is only one. Furthermore, it assigns to the lead vehicle (Lead in Figure E-1) dataMode 6, while it assigns dataMode 1 at the other object. By assigning dataMode 6 to the lead vehicle, the system reacts the same way like in case of communication failure (see Chapter 6). Such inaccurate GPS measurements occur often because a conventional GPS is used.
The solution for this case includes the development of the function “Monitor GPS” in the “Situation Awareness” block as described below.

**Monitor GPS**: The goal of this block is to distinguish whether a transition of the lead vehicle dataMode is caused by GPS inaccurate position. The block takes as input the detected objects, as obtained in the “Object Classification”. Then, it checks whether the lead vehicle is a communicating object (odd dataMode). In case of a sudden dataMode transition (from odd to even) of the lead vehicle, additional conditions are checked. The ID of each communicating vehicle is used in order to update the classification of the detected objects. If the ID of the lead vehicle is received and the communicated positions is within a predefined diameter (15m) from the previous position, then the lead is classified as communicating object. The updated position of the lead vehicle is obtained by the radar measurement. The block outputs the parameter “GPSerror” which is used in the “GPSModeSelection”.

The GPSModeSelection block is defined in the SafetyModeSelection of the SDU (see Chapter 6). Similar to the WiFiModeSelection block it consists of a stateflow diagram. The goal of this block is to output to the WiFiModeSelection the status of the GPS monitoring. Therefore, a conservative action of the WiFiModeSelection can be avoided in case it is caused by GPS inaccuracy. An overview of the GPSModeSelection block is shown in Figure E-2.

The GPSModeSelection consists of the following states:

1) **GPSstandby**: This state corresponds to the nominal functionality when no GPS inaccuracy is detected. The output GPSStatus is 0 and the SteadySafeState is 0.

2) **GPSInaccuracy**: In case an inaccuracy is detected there are two defined states
   a) **ClusterMIO**: This state is entered when clustering of an object based on the ID has been performed. The output GPSStatus is 1 and the SteadySafeState is 0 as no action is required.
   b) **Emergency**: This state corresponds to the case that the lead vehicle is classified as non-communicating and the additional check of the vehicle ID did not result in successful clustering. The output GPSStatus is 2 and the SteadySafeState is 2. When the GPSStatus is 2, the transition C (see Section 6.3.1) in WiFiModeSelection is enabled.

The transitions for the GPSModeSelection are listed below:

1) **Transition A**: GPS is initialized and inaccuracy detected.
2) **Transition B**: The dataMode of the lead vehicle does not change from odd to even (GPSerror in “MonitorGPS” is 0)
3) **Transition C**: Post-clustering occurred (GPSerror is 1)
4) **Transition D:** Default path  
5) **Transition E:** Post-clustering was not possible (GPSerror is 2)  
6) **Transition F:** Post-clustering occurred (GPSerror is 1)
# Appendix F

**Test cases for SUC 1**

The derived test cases for SUC 1 are shown in Table F-1.

<table>
<thead>
<tr>
<th>Testnr.</th>
<th>Test description</th>
<th>Notes</th>
<th>Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>WiFi Failure for 0.1 sec and lead vehicle brakes at t=0 sec with alead = -6 m/s² without any preceding vehicle in front of it.</td>
<td>1A 2C 3B 4A</td>
<td>--V--</td>
</tr>
<tr>
<td>1.2</td>
<td>WiFi Failure for 0.1 sec and lead vehicle brakes at t=0 sec with alead = -6 m/s² with a preceding vehicle driving with d_MIO = 10 m and ddot_MIO = -0.1 m/s.</td>
<td>1A 2C 3B 4B</td>
<td>--VI--</td>
</tr>
<tr>
<td>1.3</td>
<td>WiFi Failure for 0.1 sec and lead vehicle does not brake (alead = 0 m/s²) without any preceding vehicle in front of it.</td>
<td>1A 2C 3A 4A</td>
<td>--I--</td>
</tr>
<tr>
<td>1.4</td>
<td>WiFi Failure for 1 sec and lead vehicle does not brake (alead = 0 m/s²) without any preceding vehicle in front of it.</td>
<td>1B 2(A/B/C) 3A 4A</td>
<td>--II--</td>
</tr>
<tr>
<td>1.5</td>
<td>WiFi Failure for 1 sec and lead vehicle brakes at t=0 sec with alead = -2 m/s² without any preceding vehicle in front of it.</td>
<td>1B 2C 3D 4A</td>
<td>--III--</td>
</tr>
<tr>
<td>1.6</td>
<td>WiFi Failure for 1 sec and lead vehicle brakes at t=0 sec with alead = -4 m/s² without any preceding vehicle in front of it.</td>
<td>1B 2C 3C 4A</td>
<td>--IV--</td>
</tr>
<tr>
<td>1.7</td>
<td>WiFi Failure for 1 sec and lead vehicle brakes at t=0 sec with alead = -6 m/s² without any preceding vehicle in front of it.</td>
<td>1B 2C 3B 4A</td>
<td>--VII--</td>
</tr>
<tr>
<td>1.8</td>
<td>WiFi Failure for 1 sec and lead vehicle brakes at t=0 sec with alead = -6 m/s² with d_MIO = 10 m and ddot_MIO = -0.1 m/s.</td>
<td>1B 2C 3B 4B</td>
<td>--VIII--</td>
</tr>
<tr>
<td>1.9</td>
<td>WiFi Failure for 1 sec and lead vehicle brakes at t=0.5 sec with alead = -6 m/s² without any preceding vehicle in front of it.</td>
<td>1B 2A 3B 4A</td>
<td>--IX--</td>
</tr>
<tr>
<td>1.10</td>
<td>WiFi Failure for 10 sec and lead vehicle does not brake (alead = 0 m/s²) without any preceding vehicle in front of it.</td>
<td>1B 2(A/B/C) 3A 4A</td>
<td>--X--</td>
</tr>
<tr>
<td>1.11</td>
<td>WiFi Failure for 10 sec and lead vehicle brakes at t=0 sec with alead = -2 m/s² without any preceding vehicle in front of it.</td>
<td>1B 2C 3D 4A</td>
<td>--XI--</td>
</tr>
<tr>
<td>1.12</td>
<td>WiFi Failure for 10 sec and lead vehicle brakes at t=0 sec with alead = -4 m/s² without any preceding vehicle in front.</td>
<td>1B 2C 3C 4A</td>
<td>--XII--</td>
</tr>
<tr>
<td>1.13</td>
<td>WiFi Failure for 10 sec and lead vehicle brakes at t=0 sec with alead = -6 m/s² without any preceding vehicle in front of it.</td>
<td>1B 2C 3B 4A</td>
<td>XIII</td>
</tr>
<tr>
<td>1.14</td>
<td>WiFi Failure for 10 sec and lead vehicle brakes at t=0 sec with alead = -6 m/s² with d_MIO = 10 m and ddot_MIO = -0.1 m/s.</td>
<td>1B 2C 3B 4B</td>
<td>--XIV--</td>
</tr>
<tr>
<td>1.15</td>
<td>WiFi Failure for 10 sec and lead vehicle brakes at t=0.5 sec with alead = -6 m/s² without any preceding vehicle in front of it.</td>
<td>1B 2A 3B 4A</td>
<td>--XV--</td>
</tr>
<tr>
<td>No.</td>
<td>Scenario Description</td>
<td>Reference Lines</td>
<td>Page</td>
</tr>
<tr>
<td>-----</td>
<td>-------------------------------------------------------------------------------------</td>
<td>-----------------</td>
<td>------</td>
</tr>
<tr>
<td>1.16</td>
<td>WiFi Failure for 0.4 sec and lead vehicle brakes at t=0 sec with alead = -4 m/s² without any preceding vehicle in front of it.</td>
<td>1A 2C 3C 4A</td>
<td>-XVI-</td>
</tr>
<tr>
<td>1.17</td>
<td>WiFi Failure for 0.4 sec and lead vehicle brakes at t=0 sec with alead = -6 m/s² without any preceding vehicle in front of it.</td>
<td>1A 2C 3B 4A</td>
<td>-XVII-</td>
</tr>
<tr>
<td>1.18</td>
<td>WiFi Failure for 0.4 sec and lead vehicle brakes at t=0.5 sec with alead = -6 m/s² without any preceding vehicle in front of it.</td>
<td>1A 2A 3B 4A</td>
<td>XVIII</td>
</tr>
<tr>
<td>1.19</td>
<td>WiFi Failure for 0.4 sec and lead vehicle brakes at t=0 sec with alead = -6 m/s² with (d_{MIO} = 10) m and (ddot_{MIO} = -0.1) m/s.</td>
<td>1A 2C 3B 4B</td>
<td>-XIX-</td>
</tr>
<tr>
<td>1.20</td>
<td>WiFi Failure for 0.7 sec and lead vehicle brakes at t=0 sec with alead = -4 m/s² without any preceding vehicle in front of it.</td>
<td>1B 2C 3C 4A</td>
<td>--XX--</td>
</tr>
<tr>
<td>1.21</td>
<td>WiFi Failure for 0.7 sec and lead vehicle brakes at t=0 sec with alead = -6 m/s² without any preceding vehicle in front of it.</td>
<td>1B 2C 3B 4A</td>
<td>-XXI-</td>
</tr>
<tr>
<td>1.22</td>
<td>WiFi Failure for 0.7 sec and lead vehicle brakes at t=0.5 sec with alead = -6 m/s² without any preceding vehicle in front of it.</td>
<td>1B 2A 3B 4A</td>
<td>-XXII-</td>
</tr>
<tr>
<td>1.23</td>
<td>WiFi Failure for 0.7 sec and lead vehicle brakes at t=0 sec with alead = -6 m/s² with (d_{MIO} = 10) m and (ddot_{MIO} = -0.1) m/s.</td>
<td>1B 2C 3B 4B</td>
<td>XXIII</td>
</tr>
</tbody>
</table>
About the Author

DIMITRIOS TZEMPETZIS

Date of Birth: 07/04/1987
Nationality: Greek

WORK EXPERIENCE

09/2013 – 09/2015  Postgraduate Design Engineer at Eindhoven University of Technology (TU/e) in the field of Automotive Systems Design

11/2011 – 09/2013  Quality and Systems Engineer at Inos Hellas S.A (Inos Gmbh, Member of Grenzebach)

EDUCATION


SKILLS

Languages  English: Michigan Proficiency - C2 level (based on CEF)
German: ZentraleMittelstufenPrüfung – B2 level (based on CEF)
Dutch: B1 level (based on CEF)
Greek: Native Speaker

Computer  SysML
MATLAB/ Simulink
C/ C++
OpenCV
Python
Microsoft Suite
MySQL

FIELDS OF INTEREST