Model-Based Certification of Automated Vehicles

by

Michael Sebastian Schmid

B. Eng., Augsburg University of Applied Sciences (2016)

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Master of Science in Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2020

© Massachusetts Institute of Technology 2020. All rights reserved.

Author .....
Department of Aeronautics and Astronautics
April 27, 2020

Certified by .....
Nancy Leveson
Professor of Aeronautics and Astronautics
Thesis Supervisor

Accepted by .....
Sertac Karaman
Chairman of the Graduate Program Committee

Model-Based Certification of Automated Vehicles

by

Michael Sebastian Schmid

Submitted to the Department of Aeronautics and Astronautics
on April 27, 2020, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

Automated vehicles (AVs) have long been predicted to disrupt the transportation industry any moment. Although numerous companies have shared that optimism and supported development, it now seems that the challenges of building automated vehicles are becoming apparent and are pushing the vision far into the future. While manufacturers are making technological progress there are concerns about the safety of AVs. For AVs new types of accidents such as accidents due to unsafe software behavior and interactions between vehicle hardware, software, and humans must be considered in addition to what has been done for the safety of conventional vehicles.

Many stakeholders in the AV industry are looking for ways to ensure and demonstrate the safety of AV designs. For one, regulators are responsible for assessing safety and granting certification based on whether a minimum level of confidence in safety was achieved. Similarly, manufacturers need to be able to argue for the safety of their AV design in order to receive certification and convince customers. Finally, suppliers such as software/hardware providers, AV app developers, etc., also depend on a way to demonstrate the safety of their product and convince Original Equipment Manufacturers (OEMs). Hence, safety is an important problem for the progress of the AV industry that calls for a solution.

This thesis presents a potential solution by suggesting a model-based certification. First, currently existing problems with AV safety and the currently taken approaches to address AV safety are identified. Then, an alternative model-based safety approach that addresses the identified problems is demonstrated and a conceptual architecture model and safety requirements are derived. Finally, it is shown how the model-based safety approach may be implemented through certification. The application to safety-related problems such as regulation and the insurance of AVs is outlined and an organizational structure and processes for model-based certification are derived.

Thesis Supervisor: Nancy Leveson
Title: Professor of Aeronautics and Astronautics

Acknowledgments

MIT has been an exciting journey that benefited both my professional and personal development. The great success of this journey was made possible by a number of people who gave me motivation, support, and were always there for me. The following is an attempt to acknowledge their contribution and express the loads of thanks and gratitude that I owe.

First of all, I want to thank my advisor Professor Nancy Leveson for the impact she has had on my personal and academic development. Nancy has helped me to see the world from a different perspective, understand that it is often not linear, and enhance my problem-solving skills. Often, accepting that mistakes and errors happen and finding adequate measures to control them is more effective than sanctions and pretending problems can be generally avoided. I am very grateful to have met Nancy and for the many lessons she has taught me.

I owe special thanks to Dr. John Thomas who was an invaluable supervisor for me not only in all the projects that I have worked on but also for advice on my research and career. Many skills such as how to communicate strategically in projects, problem-solving strategies, etc. I learned from John during the calls, meetings, and discussions in our projects together. I want to thank John for the many times he has reviewed my work, has provided structure and guidance, and was there to give advice. Thank you for the great collaboration and support.

I want to take this opportunity to record my sincere thanks also to the institutions that have made my studies in the US possible. Thank you to the German-American Fulbright Commission who have helped me plan this journey, provided important resources such as connections to people in the US, a strong community of Fulbright fellows, and funding, and were an important stepping stone on the way to my degree. My thanks also goes to the German Academic Scholarship Foundation who supported

me throughout my studies with resources for my personal and academic development such as seminars, academies, language courses, and funds to cover study expenses. Special thanks goes to the companies Ford, Zenuity, and the Renesas Electronics Corporation which have provided projects where I could apply what I learned in real-world projects, contribute to the knowledge transfer between research and industry, and generate the necessary funding for a degree at MIT. Thank you for the interesting projects and the great collaboration.

I am extremely grateful for my family and for how they helped me become who I am today. Most importantly, I want to thank my parents Martina Schmid and Gerhard Schmid and my sister Verena Schmid for the unconditional love and support that they have given me throughout my life. It was ultimately the constant feeling that they deeply believe in me that has made me achieve all the things that I can be proud of today. Further, I also want to express my deep and sincere gratitude to my aunt Bettina Deponte who has always been there for me to offer a different perspective and advice on difficult questions that arise as a graduate student more than 6000 kilometers away from home, friends, etc.

My appreciation also extends to Martin Otcinscki and Sandra Pjanic who have been my long-term best friends that always had my back whenever I needed it. In particular, I want to thank them for being great friends, being excited about my journey although it meant that we could not see each other much, and always being honest with me whenever I needed a second opinion.

I am also indebted to my long-loved ex-girlfriend Celina Bösl for the support she has given me during all the years before I came to MIT and helping me build the path to my dream. A large share of my development goes to having spent the many happy years with you.

I extend my thanks to those who I met during my time in the US and have made me

feel home here. I want to express special thanks to Evelyn Taylor-McGregor for the many great moments, the caring support during challenging phases of my program, and for inspiring me as a person. Sincere thanks also goes to my best friend Ingrid Mosquera who has been accompanying me on this journey, has always had my back, and who I have shared the most joyful moments with. My heartiest gratitude I owe to Calah Hanson for being a wonderful person that brought happiness into my life whenever she was around. A big thanks goes to my friends from the MetaMovements Dance Company who have introduced me to a new culture and have been a source of positivity and energy throughout my degree program.

I would like to thank the members of my research group for being great lab mates and friends and helping build a strong lab spirit and community. Thank you to Diogo Castilho, Lawrence Wong, Dylan Muramoto, and Elias Johnson. My sincere gratitude also goes to Beata Shuster who has grown from being a colleague to being a very good, trustworthy, and fun friend.

Finally, I am deeply grateful to Laura Rösch who has accompanied me on my way to MIT and kept my dream of attending MIT alive throughout the years by sending me postcards of our campus. Thank you also to the many other family members and friends who helped me stay motivated, hold on to my goals and principles, and gave me the feeling of being something special for them.

Contents

1 Introduction **17**

1.1 Motivation for a Model-Based Approach to Certification 17

1.2 Research Objectives 21

1.3 Overview of Chapters 21

2 Literature Review: Safety of Automated Vehicles **23**

2.1 Automation-Related Accidents in Automotive 23

2.1.1 Description of Accidents 24

2.1.2 Common Causal Factors 31

2.2 Currently Used Practices in the Development of Automated Vehicles 36

2.2.1 Description of Available Practices and Work-in-Progress 37

2.2.2 Common Problems 41

3 Conceptual Architecture and Model-Based Safety Analysis **47**

3.1 Benefits of a Safety Approach using a Conceptual Architecture Model 47

3.2 Conceptual Architecture Generation Process 50

3.3 Safety Analysis and Identification of Safety Requirements 66

3.3.1 STPA Step 3: Identify Unsafe Control Actions 66

3.3.2 STPA Step 4: Identify Scenarios and Safety Requirements . . 70

4 Implementation through Certification **77**

4.1 Certification & Relationship to Problems in the AV Field 77

4.1.1 Regulation of Automated Vehicles 78

4.1.2 Insurance of Automated Vehicles 83
4.2 Derivation of an Organizational Structure and Processes for Model-Based Certification 86

5 Summary **103**
5.1 Research Contribution 103
5.2 Future Work 105

A Unsafe Control Actions for the Generation of a Conceptual Architecture for AVs **107**

B Causal Scenarios & Design Constraints for the Generation of a Conceptual Architecture for AVs **115**

C Design Constraints for the Generation of an Organizational Structure for Model-Based Certification **119**

List of Figures

2-1 Vehicle trajectories before and after the collision [1]. 24
2-2 Test loop of the shuttle and accident location [2, p. 3]. 25
2-3 Paths of the pedestrian and the accident vehicle [3, p. 12]. 26
2-4 Path and speed of accident vehicle leading up to the accident [4, p. 11]. 26
2-5 Trajectory of the accident vehicle before the accident ([5], [6]). 27
2-6 Vehicle trajectory leading up to the collision (graphic elements from [6]). 28
2-7 Collision paths of the autonomous truck and the water cart [7, p. 1]. . 31

3-1 Process steps for generating a conceptual architecture. 51
3-2 The socio-technical control structure for AVs. 55
3-3 Initial operational control structure for AVs. 57
3-4 A conceptual architecture model for AVs. 64

4-1 Organizational structure for model-based certification. 89
4-2 Process steps for the certification of new developments. 94
4-3 Steps for the certification of design changes. 97
4-4 Certification steps for the regulation of changes in the operating environment of AVs. 99

List of Tables

3.1 Design constraints generated from UCAs. 58
3.2 Unsafe Control Actions – Automation. 67
3.3 Unsafe Control Actions – Safety Driver. 68

A.1 Unsafe Control Actions – Safety Driver. 108
A.2 Unsafe Control Actions – Automation. 112

Nomenclature

Acronyms

AV	Automated Vehicle
CAST	Causal Analysis based on System Theory
FMVSS	Federal Motor Vehicle Safety Standards
LKAS	Lane Keeping Assist System
NHTSA	National Highway Transportation Safety Administration
NTSB	National Transportation Safety Board
SAE	Society of Automotive Engineers
SAFAD	Safety First for Automated Driving
SOTIF	Safety of the Intended Functionality
STPA	System Theoretic Process Analysis
TCAS	Traffic Collision Avoidance System
UCA	Unsafe Control Action
V2V	Vehicle-to-Vehicle Communication

Chapter 1

Introduction

This chapter lays out the potential benefits of Automated Vehicles (AVs) as well as current challenges in their development and argues for a new approach to certification. Section 1.1 outlines the motivation for a new certification approach based on which the research objectives of this thesis are formulated in Section 1.2. An overview of the thesis chapters is given in Section 1.3.

1.1 Motivation for a Model-Based Approach to Certification

Automated Vehicles (AVs) have long been predicted to disrupt the transportation industry and bring greater quality to our daily lives. Optimists agree that AVs will become a reality shortly ([8], [9]) and even that a full transition to autonomous cars will be achieved within a few decades [10]. Many companies across different industries (e.g. Google, Apple, Tesla, Volkswagen, Uber, Toyota, Mercedes-Benz, etc.) have shown their interest and commitment to the development of autonomous vehicles ([10], [11]). Beyond offering increased mobility and comfort ([11],[12], [13, Ch. 1]), relaxing today‘s parking problem [11], and possibly reducing traffic congestion [11] AVs are expected to increase fuel economy ([11], [12], [13, Ch. 1]) and offer faster

transportation at the latest once communication between vehicles and the infrastructure around them is established [10].

However, the most popular argument is usually made about safety: Advocates expect AVs to reduce the number of traffic accidents and incidents and ultimately enhance road safety ([8], [12], [11], [14], [15], [16, p. 202]). Improved safety is seen as a result coming not only from the AV's potential future ability of vehicle-to-vehicle communication (V2V) [10] but also their innate capability to outperform humans as drivers [17]. Human error is often identified as the cause in accidents[1] [11] and many argue that removing the human operator will eliminate the events that lead to accidents ([10], [13, Ch. 1]). An estimate by the National Highway Transportation Safety Administration (NHTSA) even predicts that AVs together with V2V communication would help avoid about 500000 crashes and save about 930 lives per year [10].

Besides the optimism that AVs will benefit society there is also doubt on when and if at all AVs will fulfil the expectations placed on them [13, Ch. 1]. In fact, some companies claim that the arrival of AVs is overestimated and their design will take more effort and resources than originally anticipated [18]. While the debate about AVs is often centered around technical questions (e.g. how to validate, verify, and interpret the non-deterministic behavior of algorithms, if there is enough satellite coverage for AVs to work reliably, how much sensors will be needed, etc.) ([13, Ch. 1], [19], [20]), the challenges ahead are multi-disciplinary and reach far beyond technology (see [21]).

First, there are legal challenges pertaining to liability ([14], [13, Ch. 1]). Even today, where the driving task is still shared between the vehicle and the human operator who is still responsible for the vehicle's safety, it is often hard to argue if and when the human operator should have intervened and if the human operator can be held responsible. With an increasing level of automation in AVs courts, insurance compa-

[1] Many accident analyses only identify the proximate events leading to an accident. Often however, more extensive analysis can identify a variety of causal factors.

nies, and automobile companies will face additional challenges.

Second, AVs raise questions about social acceptance [14]. Humans have been driving vehicles for decades and convincing operators to invest in new technology, give up control, and trust a computer with the driving task will be a challenge even for the companies with the most advanced products.

Third, AVs introduce a new dimension of complexity that brings along a set of additional safety gaps and challenges. As computers take on more functions and are given more authority within systems their complexity is growing [22]. In aerospace and automotive in particular, systems of the latest generation are characterized by an increased amount of software, a higher level of software and system complexity[2] ([24], [25], [26]) and operation in complex environments. AVs, for example, are being deployed in a non-homogeneous environment to operate together with conventional vehicles and other road users whose behavior is hard to anticipate ([10], [18]). Vehicle operation is characterized by shared control between the human and the vehicle [27] and highly-frequent interaction with its environment that has led to monitoring systems for driver attentiveness, further adding complexity [20]. In addition, unlike in industries like aerospace, operator training in automotive is usually low and driver selection is less rigorous [22].

Ultimately, this complexity in automotive leads to challenges in system design ([28], [11]): Complexity introduces more unknowns and interactions and systems soon become intellectually unmanageable [22]. State spaces of such systems are enormous and cannot be fully anticipated anymore[3] [22]. As a result, the completeness and correctness of safety requirements and hence, overall system safety, can no longer be guaranteed ([29], [30], [25], [31], [32]).

[2]Modern cars have up to 150 million lines of code, compared to 7 million in a B 787 [23]

[3]The Traffic Collidance Avoidance System (TCAS), for example, has 10^{40} different states [22].

Fourth, the potential benefits of AVs have created a market with strong economic competition that may compromise safety [11]. Safety is often seen as a counterweight to performance and treated as a property that needs to be traded off against performance [28]. This notion is confirmed by recent research activities in automotive that have focused on preventing the over-design of AVs (that is, optimizing the design) while making sure minimum safety requirements are met (see [8]). In industry too, it has been shown that safety is viewed as a trade-off and finding the right balance between false-positives and false-negatives can challenge companies and has contributed to accidents (see [14]).

Lastly, security poses a key challenge for the success of AVs ([33], [14]). While vehicles become more computerized and allow more functionality it requires putting security measures in place that are dedicated to prevent or limit unauthorized control by third parties ([13, Ch. 1], [10]).

Progress with AVs and turning their potential into a benefit for society is currently obstructed by the many challenges they bring along. While it was shown that those challenges range across multiple sectors, their majority can be addressed by certification. The challenges arising from the AVs‘ high complexity and strong economic competition, for example, point to AV safety and call for an answer to the question of the certification of AVs [34]. Certification that ensures safety will help the public build trust and eventually make them benefit from AVs. Although regulation alone is likely not going to resolve all legal challenges completely, it may help industry provide better designs, rely less on the driver as a backup and an instance for liability, and ultimately relieve legislators. In addition, regulation may help the roles of humans and automation to become more distinct and (legal) responsibility can be assigned more clearly.

Hence, a new approach to certification may support engineers in managing complexity, provide an opportunity for companies to design AVs without having to trade off safety

with system performance, and ultimately help companies and society benefit, leading the way to large-scale deployment of AVs.

1.2 Research Objectives

The goal of this thesis is to show how the safety problem with AVs may be addressed more effectively by a certification approach that is based on a generic AV model. Hence, the objective of this thesis is broken down into the two research questions of 1) how a generic AV model may be created and 2) how the model can be used for certification to solve currently existing problems. Research question 1) is answered by demonstrating a systematic way to derive a generic AV model and safety requirements. An answer to research question 2) is provided by describing how the derived model may be used for certification and deriving an organizational structure and processes to implement model-based certification.

1.3 Overview of Chapters

Chapter 1 gives an introduction to the expected benefits of AVs and the challenges that currently delay progress in development and their large-scale deployment. Both optimistic views and doubts on the capabilities and safety of AVs are presented and the need for a new approach to certification is identified. The chapter concludes by formulating the research objectives of the thesis.

In Chapter 2 a literature review is performed and the state of the art in the safety of AVs is given. First, a selection of automation-related accidents in automotive is presented and common causal factors are identified (see Section 2.1). Then, Section 2.2 provides an overview on the currently used practices in the development of AVs and the limitations of the currently used practices are outlined.

In Chapter 3 a generic conceptual architecture model for AVs is derived and a model-

based safety analysis is performed. First, the benefits of using a conceptual architecture model for safety is described (see Section 3.1). Then, in Section 3.2, a methodology to derive a conceptual architecture model is demonstrated and the conceptual architecture model is used to perform a safety analysis and identify safety requirements (see Section 3.3).

Chapter 4 shows how the suggested model-based approach to safety may be implemented through certification. Section 4.1 outlines the role of certification and describes how it may be used to solve safety-related problems with AVs such as regulation and insurance. Section 4.2 derives an organizational structure and processes to implement the suggested model-based certification.

Chapter 5 provides a summary of the research contribution of this thesis and provides suggestions for future work.

Chapter 2

Literature Review: Safety of Automated Vehicles

This chapter reviews the state of the art in the safety of automated vehicles and shows that there is a safety problem. Section 2.1 presents a selection of automation-related accidents in automotive and determines common causal factors. In Section 2.2 an overview is provided on the currently used practices in the development of vehicle automation and gaps in these practices are identified.

2.1 Automation-Related Accidents in Automotive

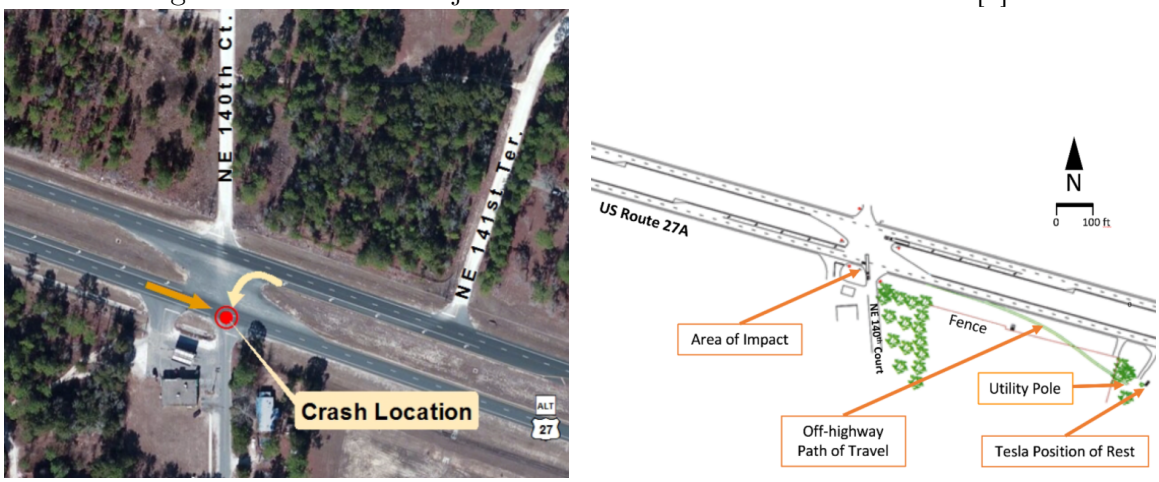
Many automation-related accidents have been blamed on human drivers that, until today, are legally responsible for the driving task, even when it is carried out by automation. While official accident investigations have often identified probable causes for accidents around the inadequate behavior of drivers the accidents point to gaps in the design of vehicle automation and therefore can be learned from. This section provides a selection of automation-related accidents in automotive and identifies a set of common causal factors. In Subsection 2.1.1 a brief description of each of the accidents is provided. Subsection 2.1.2 gives a summary of the identified common causal factors.

2.1.1 Description of Accidents

May 7, 2016 / Williston, Florida

A Tesla Model S was traveling on a highway using the vehicle's Traffic-Aware Cruise Control and Autosteer features. As a refrigeration semitrailer truck made a left turn across the Tesla's lane it struck the right side of the truck, crossed underneath it, and went off the right roadside. After leaving the roadway the Tesla continued through a drainage culvert and two wire fences, struck and broke a utility pole, rotated, and came to a rest. The impact had sheared sheared off the Tesla's roof and killed the driver. ([1], [19, p. 42]) A description of the accident is given in Figure 2-1.

Figure 2-1: Vehicle trajectories before and after the collision [1].



November 8, 2017 / Las Vegas, Nevada

As part of a pilot program Navya Arma ran one of its shuttles on a designated test loop around a shopping center in Las Vegas. On its first day of operation with passengers a truck backed into an alley in front of the shuttle and struck the shuttle with one of its front wheels [2]. The accident route of the shuttle is shown in Figure 2-2.

June, 2018 / Newfoundland

A 2016 Acura MDX equipped with a Lane Keeping Assist System was returned to its owner after a windshield replacement. When the vehicle owner used the LKAS

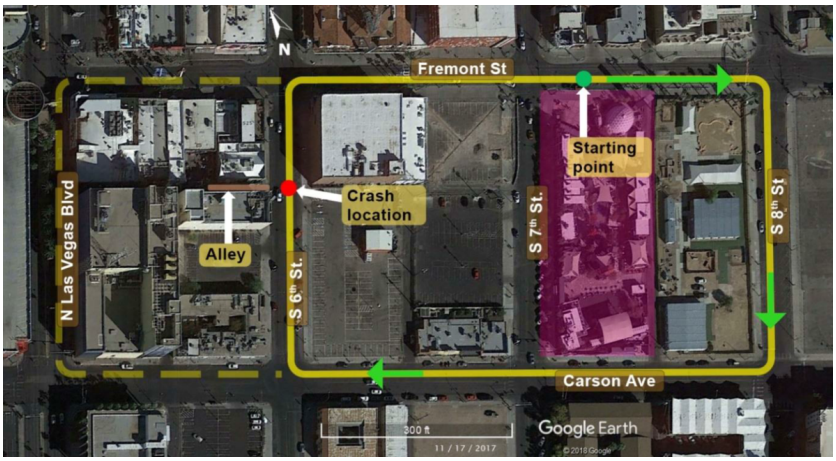


Figure 2-2: Test loop of the shuttle and accident location [2, p. 3].

after the windshield replacement on a road in Newfoundland it veered off its lane into oncoming traffic. The driver resumed control over the vehicle and prevented an accident ([19, p. 59f.], [35]).

March 18, 2018 / Tempe, Arizona

A modified 2017 Volvo XC90, operated as an Uber test vehicle with a self-driving system with one driver ([36], [3], [37]), was traveling in the right through lane of a two lane road in Tempe ([3], [37]) at approximately 43 mph in automated mode ([3], [19], [37]). As a pedestrian dressed in dark clothing pushed her bike across a non-illuminated, non-crosswalk region of the road, the Volvo did not slow down and struck the pedestrian ([19], [3], [37], [36]) with fatal consequences ([13, ch. 1], [36], [3], [37]). The accident is described in Figure 2-3 and a video showing the accident from the Volvo‘s inboard camera is provided in [38].

January 22, 2018 / Culver City, California

A 2014 Tesla Model S was cruising in a high-occupancy vehicle lane behind another vehicle in Autopilot mode when its lead vehicle changed lanes to go around a parked fire truck[1]. The Tesla remained in the lane, accelerated, and struck the rear of the fire truck at about 31 mph. The accident is visualized in Figure 2-4. [4, p. 1]

[1]Because of an earlier collision a fire truck was parked diagonally across the lane.

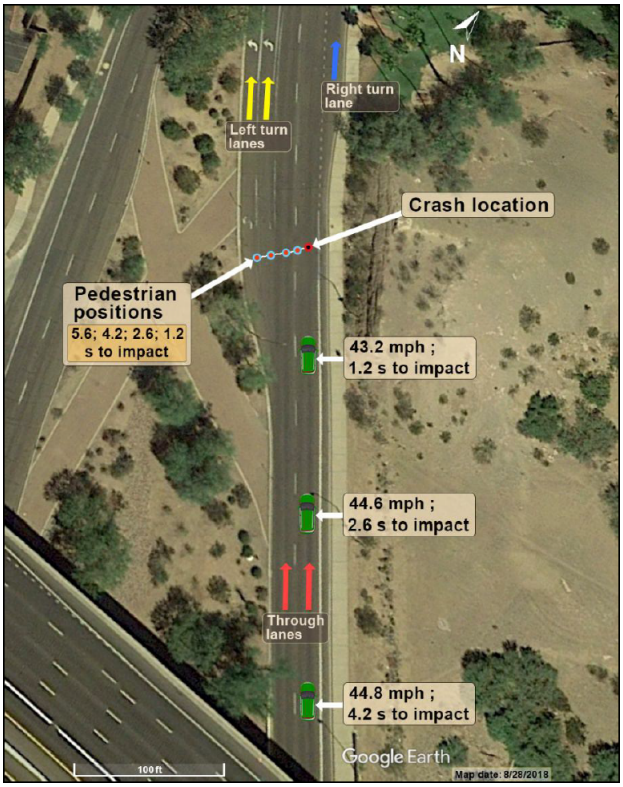


Figure 2-3: Paths of the pedestrian and the accident vehicle [3, p. 12].

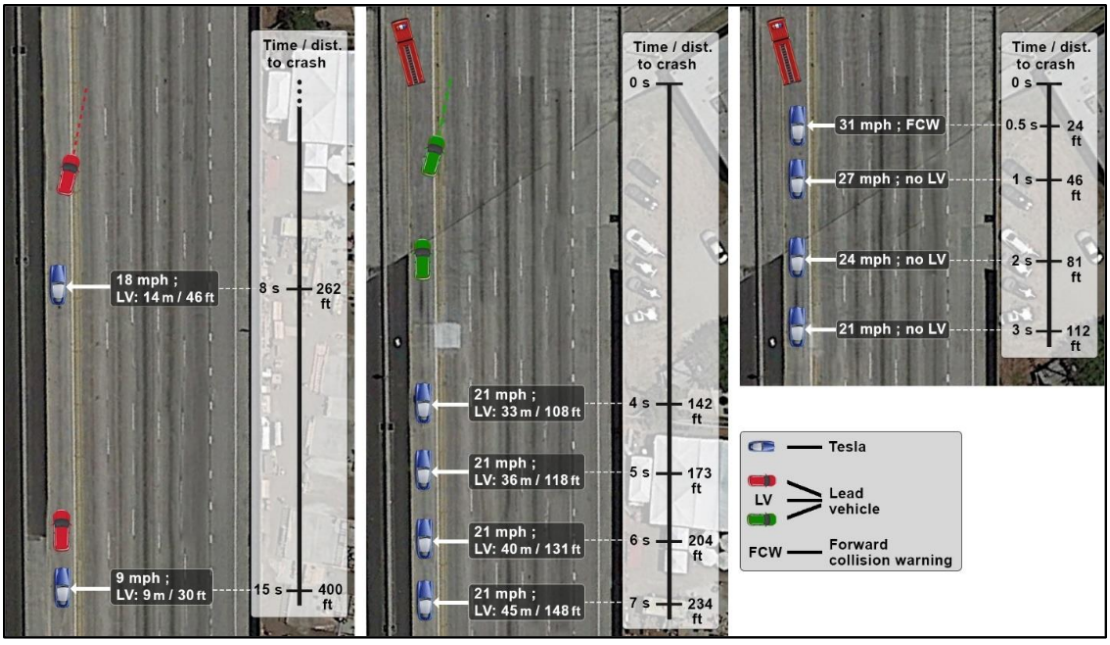


Figure 2-4: Path and speed of accident vehicle leading up to the accident [4, p. 11].

March 1, 2019 / Delray Beach, Florida

A 2019 Tesla Model 3 driving on a highway in Delray Beach, Florida in autonomous mode was approaching a private driveway as a combination vehicle pulled from the driveway. The combination vehicle crossed the highway and blocked the Tesla's path. The Tesla struck the left side of the semitrailer with about 68 mph, killing the Tesla's driver. [39]

March 23, 2018 / Mountain View, California

A 2017 Tesla Model X was moving on a highway in Mountain View using both the vehicle's Traffic-Aware Cruise Control and Autosteer features. As the vehicle approached an exit ramp it veered out of its lane, entered the gore area of the exit ramp, and crashed into a damaged crash attenuator at about 71 mph, killing its driver. A visualization of the accident is given in Figure 2-5. ([19], [5])

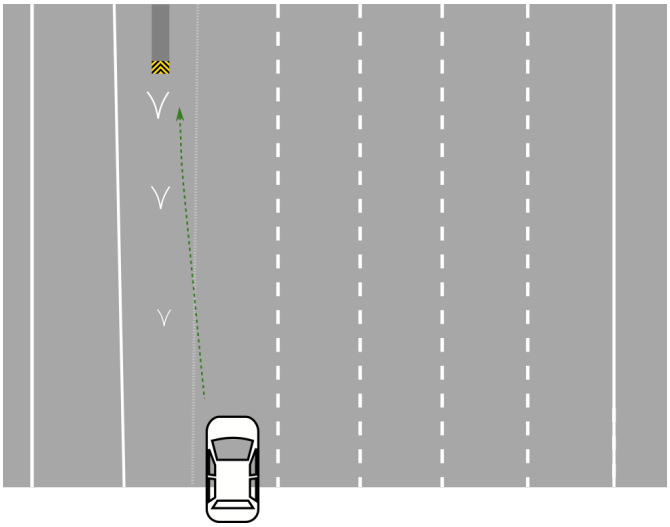


Figure 2-5: Trajectory of the accident vehicle before the accident ([5], [6]).

January 20, 2016 / China

A Tesla Model S was traveling in the left lane of a three-lane highway behind another vehicle in autonomous mode when the vehicle ahead moved into the center lane. The lane change of the lead vehicle revealed a slow-moving street sweeper truck but the Tesla did not slow down and crashed into the truck [19]. A description of the accident

is provided in Figure 2-6 and a video from the inboard camera of the Tesla is provided in [40].

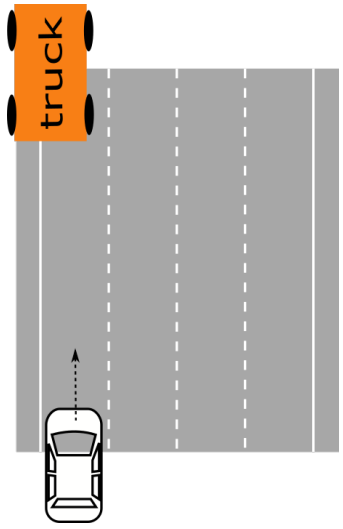


Figure 2-6: Vehicle trajectory leading up to the collision (graphic elements from [6]).

February, 2016 / Mountain View, California

A Lexus RX was driving in automated mode on a city road in Mountain View when it was about to do a right turn on an intersection. As the vehicle was programmed to go next to the curb whenever taking a right turn, the vehicle was in the rightmost lane. When the vehicle started to get to the right half of the lane it detected some sandbags that made the vehicle decide to come to a stop in its lane in order to avoid driving over them. When the traffic lights in front of the vehicle turned green the vehicle started moving and detected a bus that was travelling on its left behind it. The vehicle evaluated the space between itself and the approaching bus and concluded that there is not enough room for both of the vehicles. The Lexus assumed that the bus would slow down and the bus driver, too, assumed that the Lexus would stay put. The bus driver tried to pass the Lexus that kept advancing and hit the side of the bus. ([41], [42], [43])

December, 2017 / San Francisco, California

A General Motors Chevrolet Bolt was traveling on a multi-lane road in San Francisco when it initiated a lane change maneuver into a gap between two vehicles. As the gap closed rapidly because the vehicle ahead was slowing down the Chevrolet veered back into its original lane. However, a motorcyclist had already moved forward besides the Chevrolet and blocked the return maneuver. As a result, the Chevrolet was in a dilemma situation to either collide with the motorcycle or to crash into the cars in the adjacent lane and knocked over the motorcyclist. ([44], [45])

March, 2017 / Tempe, Arizona

A vehicle driving on a road in Tempe, Arizona was about to make a turn when an Uber vehicle in automated mode ran a red light [46]. The vehicle could not brake fast enough to avoid a collision and crashed with the Uber vehicle. As a result of the collision the Uber vehicle over-reacted and over-turned. The behavior of the autonomous driving system was traced to its dynamic model which did not consider disturbance forces from other vehicles acting on the car [44]. ([47], [48])

Tesla's Smart Summon Feature – Incidents and Issues

A new feature called *Smart Summon* allows Tesla vehicles to leave parking spaces and navigate around obstacles to their owners autonomously in eligible areas such as supermarket parking lots ([49], [50], [51]). In the widespread use of the feature various issues have been reported that cast doubt on whether the technology is ready for use ([51], [49]).

For one, customers have complained that the automation feels glitchy and sometimes only works intermittently without an actual benefit for customers [50]. The system is designed to work only in private parking lots but even then sometimes seems confused about its location. The vehicle would often stop suddenly for no obvious reason and erratically wander left and right [50].

Second, vehicles would sometimes violate traffic rules. For example, during use of the feature vehicles would not always stay on their side of the lane in parking lots and in one instance a Tesla drove itself the in the wrong direction on a one-way lane [50]. Another owner claims his Tesla had a near miss when it seemingly failed to understand its lack of right of way [51].

Third, vehicles have been caught running into obstacles while performing seemingly simple tasks. One owner, for example, reports that his vehicle ran straight into a wall when it was pulling straight out of his wide garage onto an empty driveway [51]. In addition to the many issues customers have deemed the feature useless in most situations also because the car would be too slow in driving and take too long to pick up a person [52].

August, 2015 / Newman, Australia

In a mine in Newman, Australia a control room operator had programmed an autonomous haul truck to do a right turn and perform a loop so the truck could be positioned underneath an excavator. However, while the right turn and the loop existed in the control system it was not physically signposted or marked to notify workers and manually-operated vehicles. When the autonomous truck was following its programmed procedure and was about to turn a manually-operated water cart that was travelling in the opposite direction entered its path. The autonomous truck detected the water cart in its path but could not prevent the collision anymore due to its speed of about 40 km/h and collided with the water cart. ([53], [7]) A description of the accident is given in Figure 2-7.

Other Accidents

Until October 16, 2019, the Department of Motor Vehicles of the State of California has registered 202 collisions involving autonomous vehicles. A complete list of the registered collisions can be found in [54].

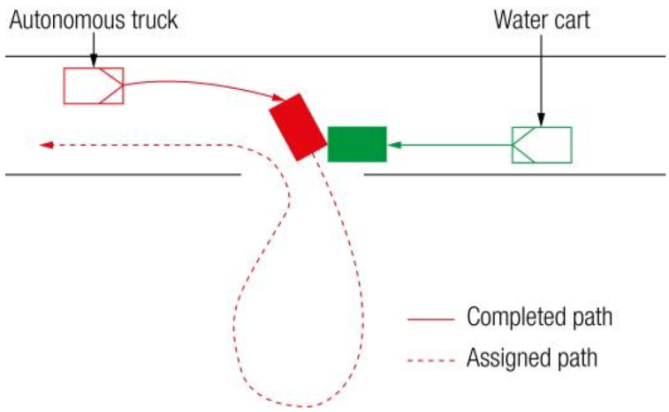


Figure 2-7: Collision paths of the autonomous truck and the water cart [7, p. 1].

2.1.2 Common Causal Factors

The described accidents show that safety is an important problem as vehicles become more and more automated. While automation has received much attention for its benefits an analysis of the presented accidents suggests that there is a set of common factors that has remained unaddressed and calls for a more effective approach to safety. The common causal factors among the presented accidents are given below.

Drivers of automated vehicles did not monitor the automated driving system sufficiently.

Driver attentiveness was an important causal factor in many of the presented accidents. Often, drivers were found overly reliant on the automation system and inattentive for most of the time when they were travelling in autonomous mode ([55], [19], [5], [56], [4, p. 7ff.], [1, p. 17], [19, p. 42ff.], [57], [58, p. 9], [59], [37, p. 3f.], [36]). In addition, accident analysis has revealed that feedback from owners of automated vehicles and other sources that suggests that the observed inattentiveness is common and not limited to the presented accidents (see [19, p. 40], [55]). Manufacturers have dedicated features and warning systems to keep drivers involved in the driving task but the provided warnings did not ensure sustained driver attention ([5, p. 2], [4], [1, p. 14]). As a result, drivers were often too distracted to realize, intervene, and avoid an imminent collision.

Manufacturers did not ensure that the functionality of their automation design is sufficient to perform the intended function.

The presented accidents have provided multiple examples in which automated vehicles did not detect a vehicle ahead or an imminent collision ([10], [19, p. 38, p. 44], [1, p. 1, p. 15, p. 30f.], [58, p. 1]). While there was no fault in the systems of the accident vehicles and their performance was in line with the industry state of the art ([1, p. 31], [58, p. 9, p. 11]) the used systems rely on rear-end collision avoidance technologies that are not designed to consider all possible crash modes. For example, potential collision scenarios such as cut-ins, cut-outs, and crossing path collisions are outside of the functional scope of state-of-the-art collision avoidance systems and may not always be identified ([58], [19, p. 44], [1, p. 25f.]).

Manufacturers did not ensure that the presentation of automation features to the public accurately represents and makes aware of system limitations.

Many automotive OEMs refer to their automation features as features for autonomous driving or autopilot features. However, to date most features still require human supervision and therefore some of the terms that manufacturers use to market their features have been found misleading ([60], [5], [4, p. 4]).

Tesla, for example, labels its vehicle‘s automation capabilities *Autopilot* which suggests that the vehicle has a self-sufficient driving system [19, p. 39, p. 41]. At the same time, Tesla‘s owner‘s manual points to the many limitations of the system, the need for human supervision, and indicates that the system is rather a driving assistance system. Drivers have frequently expressed that the name “Autopilot“ was misleading in understanding the system‘s limitations and that the capabilities of the feature have been oversold [55]. This misleading presentation of automation features has contributed to drivers‘ lack of understanding of system limitations, over-reliance, and is an important causal factor.

Manufacturers did not ensure sufficient maturity of automation designs before release into operation.

To date, most automation features in automotive still rely on human supervision as a backup for the limitations of automation technology. As many manufacturers argue that testing on public roads is necessary to identify problems and improve designs the release of systems with low maturity is intentional and common practice ([50], [52]). This approach, however, has lead to problems and has been an important factor in accidents and problems with automation in automotive ([19, p. 39, p. 41], [2, p. 7]). As an illustration, Tesla customers have complained about being used as beta testers for the Smart Summon feature that lulls users into a false sense of confidence [51]. Similarly, some accidents have involved scenarios in which the AVs did not detect obstacles or other vehicles (see [4], [19, p. 44ff.], [58]) because of low contrast with the background, complex or unusual vehicle shapes, etc.

Manufacturers did not ensure the validity of design assumptions.

Any system is designed to achieve a certain goal under specified conditions or assumptions about the environment. The design of automation in particular requires assumptions not only about the vehicle‘s operating environment but also about the expected behavior of other road users such as manually driven vehicles, pedestrians, bicyclists, etc. These assumptions have been challenging manufacturers of automated vehicles and causal factors related to inadequate assumptions can be identified in many of the presented accidents (see [44], [45], [44], [47], [48], [19, p. 59f.], [35]).

For example, in the Uber accident in Tempe the AV software did not predict the pedestrian‘s path accurately as a result of different inadequate assumptions in software design [3]. For one, it was assumed that pedestrians only cross roads at crosswalks and may not be expected elsewhere. Another flawed assumption included that bicyclists generally stay in their lane as they follow the road. In the accident, however, the pedestrian crossing the road was intermittently identified as a bicyclist (as she was pushing a bicycle) and hence, the software did not anticipate that the pedestrian

would enter the vehicle‘s lane.

Manufacturers did not sufficiently consider the safety implications of design decisions.

The field of automated vehicles is characterized by high competition, technological progress, and highly dynamic development. While manufacturers regularly integrate new or more improved components the high degree of coupling between system components frequently challenges manufacturers‘ capability to account for system properties such as safety. Often, safety is only considered on component level and the safety implications of decisions in system design are missed.

To illustrate, both Uber and Tesla agree in their design decision that objects must be identified by both camera and by radar in order to detect an emergency situation. However, the presented accidents have shown several scenarios in which this philosophy has contributed to accidents (see [19, p. 39], [4, p. 10ff.], [1, p. 30]). Further examples for inadequate design decisions that contributed to accidents include Uber‘s decision to disable Volvo‘s default Automatic Emergency Braking system ([19], [61], [37, p. 2]), Uber‘s accident suppression concept [3, p. 9] and the decision to use invisible travel paths for autonomous haul trucks [7].

Manufacturers and operator companies did not anticipate the safety impact of management decisions and did not devise sufficient measures to address currently existing safety issues.

To date automated vehicles still rely on human supervision that is, an attentive driver who will prevent an accident if the automated system no longer seems to operate safely. As a result, safety is ultimately in the hands of the human driver and the role of human drivers can no longer be reduced to supervision only. However, many of the presented accidents have demonstrated the disconnect between operations, design, and management ([62], [37], [3]) and drivers of automated vehicles were rarely consulted or involved in decisions that affect safety.

For example, when Uber decided to reduce the number of drivers in each vehicle to one person the impact on safety was not considered. While drivers complained that they were no longer able to monitor each other and were overwhelmed by the additional tasks they were now responsible for ([37], [3]), management did not identify the safety impact of their decision[2]. Similarly, also in other accidents the accidents' causal factors had been identified before and had been reported by drivers but no decision to make design changes or pause operation of the associated automation features was taken (see [62], [19]).

Manufacturers did not sufficiently account for interactions with manually-driven vehicles.

The transfer of the driving task from the human driver to automation is often justified by the potential safety benefits of replacing the error-prone human component. However, while automation may behave more consistently the accidents presented in this thesis have shown that unnatural behavior of automation may cause problems in the interaction with other road users such as manually-driven vehicles, motorcycles, etc. and can contribute to accidents.

In the shuttle accident in Las Vegas, for instance, the backwards driving truck driver assumed that the shuttle would stop at a reasonable distance from the truck. The shuttle however, continued slowly towards the truck and the next time the truck driver checked the shuttle's position the truck had already hit the shuttle (see [2]). Similar interaction problems had also caused the collision between an automated vehicle and a motorcyclist (see [44], [45]).

[2] A detailed analysis of the accident revealed that Uber's risk assessment procedures were inadequate, that Uber did not have a safety division or a dedicated safety manager, and that there were other safety-related lapses [59].

Manufacturers did not sufficiently enforce constraints such as limitations in the operational envelope of the system or local traffic rules in vehicle design.

The violation of rules by human drivers is a common problem that automated vehicles are expected to solve. However, to date AV manufacturers do not restrict vehicle capabilities based on design constraints or local traffic rules and drivers are able to misuse automated driving systems. For example, drivers have used automation to set excessive speeds that have contributed to multiple accidents ([56], [1, p. 11], [60], [5], [48]). Another example where misuse has caused problems is the use of automation features outside of the intended scope such as on city roads, public parking lots, etc. ([49], [50], [51], [52]).

2.2 Currently Used Practices in the Development of Automated Vehicles

Automotive systems have historically been characterized by rather simple mechanical, hydraulic, and electric systems and, as a result, development in automotive has long been dominated by only a few standards. As vehicle systems became more sophisticated and included more and more software and automation, the need for more guidance in vehicle development was realized (see, for example, [16], [63], [31], [64]) and organizations started to extend existing standards as well as create new guidelines. This section presents already existing standards along with work in progress and identifies their limitations. Subsection 2.2.1 provides a short description of the practices while Subsection 2.2.2 gives a summary of the identified limitations.

2.2.1 Description of Available Practices and Work-in-Progress

ISO 26262: Road vehicles – Functional safety

ISO 26262 is an international standard for the functional safety of electrical and electronic systems in production automobiles below a gross weight of 3500 kg. ISO 26262 was adapted from IEC 61508 to address the specific needs of electrical and electronic systems and focuses on malfunctioning behavior ([65], [66, p. 4f.], [67]). The standard prescribes a systems engineering process for safety engineering that covers all phases of the life-cycle of safety-related systems, including software components ([66, p. 9], [65]).

The process prescribed in ISO 26262 uses a top-down approach in which first, a hazard analysis is conducted to identify potential hazards and system level requirements([66, p. 12], [65]). The identified hazards are then assessed based on their likelihood of occurrence and severity. The risk assessment is then used to derive Automotive Safety Integrity Levels (ASILs) that are assigned to the respective each of the system level requirements [66, p. 19]. The assigned ASIL levels map to methods for verification and determine whether confirmation measures are required. These confirmation measures are then used to provide evidence to build a case that a system is acceptably safe ([67], [65]).

ISO 21448: Road vehicles – Safety of the intended functionality

ISO 21448 is an international standard currently in progress that was initiated to address safety gaps due to functional limitations and misuse [68] and is referred to as the standard for the Safety Of The Intended Functionality (SOTIF). ISO 21448 specifically targets unknown and hazardous behavior ([57], [68]) and provides a process and methods to identify and address SOTIF-related hazards. The process is structured in eight steps ([68], [57]) that are outlined in the following.

First, a specification of the system and its function is created (Clause 5). Then, SOTIF-related system hazards are identified, the associated risks are evaluated, and validation targets are specified (Clause 6). Subsequently, the system is analyzed for so-called *Triggering Events* (Clause 7)[3] and the probability of a hazardous system response is evaluated and compared to the validation targets (see Clause 6). Then, measures to reduce the identified risks are selected (Clause 8) and a validation and verification strategy is defined (Clause 9). The defined strategy is then used to demonstrate safety by verifying known use cases (Clause 10) and validating unknown use cases (Clause 11). The validation of unknown use cases is based on existing knowledge and statistics and targets at providing evidence that the validation targets have been met. The residual risk is evaluated and a documentation that there is no unreasonable risk in real-life scenarios is generated. Finally, the performed SOTIF activities are reviewed[4] and a recommendation with respect to acceptance is given.

UL 4600 – Standard for Safety for the Evaluation of Autonomous Products

UL 4600 is an automotive safety standard currently under way by Underwriters Laboratories and Edge Case Research. UL 4600 was created to address the safety of fully autonomous products and provides a framework to build a structured argument that an autonomous product is sufficiently safe. While it is stated that UL 4600 may be used to validate the safety of Artificial Intelligence (AI) components, Advanced Driving Assistance Systems, AV testing safety, and human factors are outside of its scope.

The safety argumentation in UL 4600 is based on the definition of a safety goal (a definition of the desired level of safety) and the argumentation of a safety case. The

[3]Triggering Events are conditions where functional insufficiencies and reasonably foreseeable misuse may lead to a potentially hazardous system response [68].

[4]This includes examining whether all the specified use cases in the intended scope of functionality were covered, whether a minimal fall-back solution is provided, and whether the acceptance criteria was met sufficiently to justify that the overall risk is unreasonable [68].

safety goal may be defined by companies or regulators. In contrast, the safety case is built using so-called prompts. Prompts are statements provided in UL 4600 that specify deviation categories for topics that are considered safety-relevant. These deviation categories define if and under which conditions a topic may be left out of a safety case. A safety case is built by addressing all required prompts with safety evidence that shows that the prompts have been addressed [69]. This safety evidence (e.g. generated by using a certain tool, technique, etc.) must match the integrity level defined for the safety goal [69] and can be generated using any standard or method. Once a safety case is built a self-audit is performed and a subsequent independent assessment is conducted to ensure that the safety case is well-formed and complete. ([69], [70], [71], [72], [73])

DO-178C – Software Considerations in Airborne Systems and Equipment Certification

DO-178C is an international standard for the development of commercial avionics software [74]. The standard provides guidelines for software certification and assigns assurance levels to software requirements based on their impact on system safety. Certification approval is given depending on whether satisfaction of the software requirements has been demonstrated with adequate rigor (depending on the software assurance levels) and documented [66, p. 23].

AUTOSAR (AUTomotive Open System ARchitecture)

AUTOSAR is an international collaboration between automotive manufacturers, suppliers, software companies, and other technology providers [75] that provides a standardized reference architecture for the software of Electronic Control Units (ECUs) [76]. The standard consists of a set of specifications to improve compatibility between electronic components within vehicles and enable software components to be developed independently [66].

MISRA C – Guidelines for the Use of the C Language in Critical Systems

MISRA-C is a standard by the Motor Industry Software Reliability Association (MISRA) that provides guidelines for the software development of automotive embedded systems in C [66, p. 8]. The standard specifically targets safety, security, portability, and reliability [77].

Safety First for Automated Driving (SAFAD)

Safety First for Automated Driving is a safety framework published by a group of automotive OEMs, automotive suppliers, software companies, and other technology providers [78] that provides a set of methods for the verification and validation of certain automated vehicles[5].

In order to design automated driving systems that outperform the average human driver [78] the framework defines twelve principles of Automated Driving which are used to determine necessary capabilities of any automated driving system [78, p. 6ff.]. Then, from the determined system capabilities a set of necessary system elements and required skills is identified and is used, together with some general requirements defined in the framework as a general specification of a safe functionality for automated driving systems [78, p. 29]. The identified safe functionality is then analyzed for potential risks and states of tolerable risk (so-called Minimal Risk Conditions) are defined for each of the system elements. The Minimal Risk Conditions are used to form a safety concept based on which verification and validation is performed to build confidence that the system is safe [78, p. 34].

References to Other Practices

The previous paragraphs have described some of the most relevant practices for the safety of automated vehicles. Beyond the practices described above there are several other resources that may also provide guidance. The references to these resources are provided below.

[5]The standard explicitly addresses automated vehicles of levels 3 and 4 only (as defined by the Society of Automotive Engineers (SAE)) [78, p. III].

- Measuring Automated Vehicle Safety (RAND Corporation, see [79])
- MIL-STD-882E - Standard Practice for System Safety (see [80])
- Federal Motor Vehicle Safety Standards (see [81])
- Voluntary System Self-Assessments (see [82])
- A Vision for Safety by the National Traffic Safety Administration (see [83])

2.2.2 Common Problems

The described practices present multiple approaches to the development of vehicle automation and AV safety. However, many of these approaches draw on the same principles and have similar limitations ([84, p. 488], [85]) . These generic limitations of the currently used practices are outlined in the following. The most important gaps of the currently available engineering practices are identified below.

For one, many currently used practices are inadequate to address AV safety because of their orientation towards less complex systems. Most established standards and methods were introduced when manually controlled system were prevalent. As a result, their focus often lies on component behavior. Today‘s systems however, are characterized by more software and automation and require a more systems-oriented approach ([63], [78, Ch. 2]).

ISO 26262, for example, was first released in 2011 and even today still builds on concepts like that are no longer applicable for autonomous systems ([8], [86])[6]. Even more recently developed standards such as ISO 21448 mainly target driving assistance systems and, as a result, may not be adequate for all types and levels of vehicle automation. Similarly, AVs also stretch the limits of the currently used safety techniques [22]. Several studies have identified limitations ([87, p. 3], [66, p. 13], [8]) that

[6]For example, controllability, which is no longer applicable with autonomous systems, is used to evaluate the risk of hazards

make the current safety techniques unfit for the application within AV development.

Second, many current resources consider safety too late in development [88, p. 13]. Safety needs to be integrated into early stages of development or otherwise problems can be found only late when design changes are difficult and expensive ([78, Ch. 2], [89], [26], [90]). In contrast, many of the existing practices consider safety only late in development ([78, p. 74], [91, Ch. 1]). For example, many practices use safety methods such as Fault Tree Analysis (FTA) which require a high level of design detail that is only available late in the design [91, Ch. 1, Ch. 2]. As a result, safety is often limited to analyzing designs that are already complete and hard to change [25].

Third, available processes do not provide sufficient guidance in AV development. For example, Feth [31] argues that there is not enough guidance for manufacturers on how to create a safe nominal behavior specification ([31], [63], [19]). ISO 26262 and MIL-STD-882E prescribe a process but do not provide specific techniques [64]. Similarly, DO-178C and the FMVSS‘s do not give specific methods [66, p. 20] and the techniques that are recommended in some practices do not provide guidance on how causal factors for hazards may be identified [91, Ch. 2].

Fourth, current practices provide insufficient traceability between requirements and design. Requirements must be traceable from their origin through different system levels to design artifacts in order to support system development. While many current practices account for requirements generation and flow-down they often leave gaps between requirements on different system levels [33]. For ISO 26262, for example, more refinement steps have been suggested to provide more traceability [13, Ch. 3].

Fifth, many of the currently used practices assess safety based on simulation, testing, and statistics ([57], [78, Ch. 2], [78, p. 86]). Safety is an emergent system property that can only be evaluated by considering the entire system. By definition, techniques such as testing, simulation, and statistics are projections of the operation of the system in

the real world and therefore are limited in their ability to account for all factors that may be relevant for safety [22]. This leads to a number of general limitations that are outlined in the following.

- Testing and simulation are discrete techniques and can only cover a small fraction of the large state space of today‘s systems ([22], [92]). Any technique to examine system behavior by experimentation is limited to providing a system response for a specific setting of inputs, environmental parameters, and other factors. However, many systems today such as in automotive involve a large amount of software that cannot be exhaustively tested or simulated.

- Testing and simulation can only reveal errors rather than prove that a system is flawless. As mentioned in the previous bullet, experimental techniques such as testing and simulation are discrete techniques that produce system responses for specific test cases. The system responses may either reveal specific problems or provide the expected results for the tested cases [92]. However, even if the experiments provide the expected results for all tested cases no conclusion may be made that the system is flawless[7]. Safety assessment requires considering all potential scenarios and evaluating if the hazards have been sufficiently addressed. Hence, experimental techniques may serve to find problems and gain confidence in a design (see [22]) but cannot be used to make conclusions about safety.

- Third, testing, simulation, and statistics are not applicable to software, humans, and their interactions. Software as well as human behavior are not stochastic and there is no effective way to test or simulate their interactions [92].

- The mentioned techniques all depend on assumptions that may not always hold and are therefore inadequate to evaluate safety [92]. Many examples in which vehicles were involved in accidents as soon as they were used on real roads after being tested in billions of simulation test cases confirm the limited success of testing, simulation, and statistics for safety assessment.

[7] unless exhaustive testing of all system states is possible.

Sixth, many current practices do not provide an approach to determine if vehicle automation is sufficiently safe. Certification and other related activities such as insurance rely on means to assess safety objectively and decide if a certain required level of safety has been met. Many international standards today such as ISO 26262, DO-178C, and UL 4600 follow an argumentation-based approach using safety cases to demonstrate that enough efforts were made to be confident in the safety of a design[66, p. 24]. However, an argumentation based on safety cases is subjective and does not provide an accurate estimate of safety but merely presents a structured documentation of the efforts to address subjectively chosen goals.

Seventh, current practices only address parts of safety and leave gaps for hazards. Safety is a system property that can only be achieved if all its parts including failures, component interactions, etc. are considered [78, Ch. 2]. Many standards are highly specialized and often address only a narrowly defined range of problems. As a result, no one standard alone provides sufficient guidance to achieve safety. Even together, the available standards do not cover all aspects of safety (see [31], [63, p. 62], [68]) and hence, are insufficient to address the safety problem with AVs. For example, while ISO 26262 mainly covers risks due to failures ([63], [93]) ISO 21448 addresses functional limitations and misuse [31], and UL 4600 is limited to fully autonomous systems [70]. By definition, ISO 21448 does not include unsafe interactions between system components [68] and UL 4600 does not consider driving assistance features or human factors [69].

Eighth, the practices that are currently used assume a correct and complete set of requirements at the start of development and ignore that many issues (e.g. safety issues, budget overruns, etc.) are caused by requirements flaws [94, p. 4]. Only rarely all requirements are defined at the beginning of design and often new requirements are identified along the way as design progresses. Nevertheless, the processes in many of the currently used practices rely on knowing all design requirements and keeping them mostly unchanged during development. In ISO 26262, for example, it is assumed

that all functions are specified in a safe way [14] and that focusing on deviations from the specification is sufficient to achieve safety [63].

Ninth, most of the currently used practices rely on traditional safety methods that are either inadequate for the application to AV safety or suffer general limitations ([95], [26], [96], [12], [13, Ch. 1]). A summary of the problems with many of the traditional safety methods is provided below.

- Many safety techniques rely on system decomposition into components for independent analysis (see [78, p. 90], for example). Safety, however, is an emergent system property and cannot be evaluated by looking at individual components.

- The majority of traditional techniques focuses on component failures and malfunctions and hence, cannot account for software ([25], [66, p. viii]).

- Human factors are considered only inadequately [66, p. viii]. Many traditional safety techniques consider operators an external element rather than a part of the system [97]. However, with more automation in vehicles come more human-machine interactions and operators can no longer be seen as separate ([91, Ch. 2], [78, p. 60]).

- Many safety methods such as FMEA, ETA, FTA, etc. are limited to random component failures and assume independence between components [92]. Complex systems however, fail in non-random ways and their components are highly interconnected. The behavior of software and humans, in particular, cannot be described with linear models or stochastics ([68, p. 12, p. 25], [91, Ch. 2]).

- Many safety techniques involve some kind of quantification ([66, p. 17], [68, p. 26]). Often, probabilities are used to rank hazards and identify the most effective use of resources [91, Ch. 2]. However, assessing risk based on probabilities is often inadequate ([13, Ch. 3], [90]). For one, probabilities are hard to obtain when only little design detail is available such as in early steps of

development ([91, Ch. 2], [98], [66, p. 17]). Further, probabilities are not applicable to all aspects of system behavior such as software and interaction with humans ([91, Ch. 2], [66, p. 14]). Finally, a risk quantification approach treats different risks, e.g. risks with high probability and low severity vs. risks with low probability and high severity, in the same way [91, Ch. 2] and ignores that depending on the problem different solutions may apply.

- Many of the available techniques, e.g. FMEA, FTA, etc. address reliability rather than safety ([63], [99], [68, p. 2], [78], [91, Ch. 2], [66, p. 13]). However, focusing on reliability alone is insufficient [91, Ch. 2] for addressing AV safety [25] as hazards may also arise from human behavior, software flaws, interactions between humans and software, etc. With their focus on reliability these safety methods are biased towards suggesting redundancy, fault tolerance, and over-design [91, Ch. 2] to address safety but will likely be ineffective for the safety of AVs ([33], [99]).

- Available safety techniques often draw on previous experience gained from the operation of similar systems [66, p. 14]. Such information however, is not available for new types of systems such as AV‘s for which similar designs have not been deployed yet ([66, p. 13f.], [95], [91, Ch. 2]). Even if the information was available the lessons learned may not always be applicable to new designs [22].

- Risk assessment is often done using some form of risk assessment matrix. Once the risks have been identified they are evaluated and measures to address the risks are identified. Current techniques however, do not consider the effectiveness of the chosen measures. As a result, whether risks have been addressed adequately is unclear and often depends on subjective judgement [91, Ch. 2].

Chapter 3

Conceptual Architecture and Model-Based Safety Analysis

This chapter presents a new approach to address the currently existing safety problems with AVs using a conceptual architecture model. Section 3.1 outlines the benefits of a model-based approach to safety and describes how this approach addresses the limitations of current attempts at AV safety. In Section 3.2 a systematic process to generate a generic AV model (a conceptual architecture) is demonstrated and a conceptual architecture for AVs is generated. The generated conceptual architecture model is then used in Section 3.3 to perform a safety analysis and identify safety requirements.

3.1 Benefits of a Safety Approach using a Conceptual Architecture Model

Many of the approaches currently used in the development of automated vehicles address safety only inadequately ([92], [91], [8], [66, p. 13]). As shown in Chapter 2 the approaches lack capabilities to account for important characteristics of today‘s systems such as complex software, human factors, and component interactions and

are often based on linear causality models that are too simplistic for state-of-the-art systems. In contrast, many of the identified limitations can be addressed by using a systems-theoretic, model-based approach to safety ([89], [100], [95], [91, Ch. 2]). STPA is a method based on systems theory that uses a control-abstraction model for analysis and is used in this thesis as an example to show how a model-based approach may help to address the limitations and safety-related problems with AVs that were identified in the previous chapter more effectively. In the following an overview of the benefits of using STPA and leveraging the control structure model that it relies on is provided.

Automated vehicles are characterized by the large amount of software and high complexity that their additional functionalities compared to conventional vehicles bring along. A common concept in engineering to address complexity is abstraction. However, many of the currently used approaches rely on a component abstraction that limits analysis to failure-based approaches and is inadequate for software ([63], [25], [91]).

In contrast, a model abstraction based on systems and control theory helps to manage complexity while software and problems beyond failures can still be considered. STPA was specifically designed for complex, software-intensive systems and the control structure model it uses unifies the theories of emergence, hierarchy, and control [87]. As a result, STPA has not only been shown to be scalable to very complex systems [101] and been successfully demonstrated in many automotive contexts ([33], [99], [24]) but hazard analysis is no longer limited to component failures.

With its control structure model STPA provides a systematic process to analyze interactive effects and identify unsafe control ([64], [101], [24], [26]). In this way, STPA helps to identify more causes of hazards than the currently used methods (e.g. unsafe control due to inadequate software behavior, human factors, human-machine interactions, etc.) ([87], [26], [91], [102]), can be applied beyond purely mechanical systems

to highly-complex and software-intense systems[1], and is fit for addressing the safety of automated vehicles.

The component abstraction that most of the current safety approaches rely on requires a fairly high level of design detail [91, Ch. 1, Ch. 2]. Such detailed design information is usually not available in early stages of development and as a result safety is often considered only late in development ([26], [89], [88, p. 13]). In contrast, STPA relies on a systems-control theoretic model abstraction and therefore requires only a control concept of the system rather than design details. This brings multiple advantages with respect to the safety of automated vehicles (see below).

First, STPA may be used when only limited design information is available ([91, Ch. 2], [89], [87], [26], [66, p. 15]) and therefore allows integrating safety early in the development. Second, STPA allows a generic approach to the safety problem with vehicle automation. The field of automation is characterized by high competition and hence each AV manufacturer makes their own choice of technology, depending on which will provide the most competitive advantage. In addition, as software assumes a more central role, a new market for customization of automation arises. Together with the already overwhelming amount of software in today‘s vehicles ([25], [29]) assuring adequate software behavior for each individual design becomes almost impossible.

However, as STPA requires only a generic control abstraction model of a system rather than design details the large variety of automation designs may be addressed. Hence, a safety approach based on a systems-theoretic model may provide the necessary engineering backbone to enable manufacturers to realize the potential of automation that is predicted for society.

Many safety methods today are used to identify potential hazards and quantify their

[1]STPA may also be applicable to non-deterministic systems such as in some future automotive concepts (see [103]).

associated risk only to make an argument if those hazards have to be addressed ([92], [66, p viii], [91, Ch. 2]). Even if risk were estimated accurately[2] current methods do not provide guidance on how the risks can be addressed ([63], [31], [19]). In contrast, with its control structure model STPA provides guidance on the identification of countermeasures to control hazards. As unsafe control is derived from a control structure, a control-theoretic model abstraction of the system, adequate measures may be found by either adapting the model or by adding additional control responsibilities to the controllers in the control structure model. The additional control responsibilities are documented as safety constraints and may be used to support engineers in selecting a system architecture ([99], [24]) and identifying component safety requirements.

Hence, an approach to safety that uses a control-abstraction model such as in STPA will help understand potential safety issues early in development, will provide guidance in identifying adequate countermeasures and will ultimately help addressing safety more effectively. A conceptual architecture model is a refined control structure that results from adapting a control structure using the design constraints and safety requirements generated through safety analysis. A detailed description of the process to generate a conceptual architecture is provided in the next section.

3.2 Conceptual Architecture Generation Process

Leveson [89] suggested a conceptual architecture model that is based on principles from systems- and control theory and hence, offers many benefits for addressing the safety problem with AVs (see previous section). The process starts with a highly-abstracted model of the system (the initial control structure) and uses safety analysis results to refine the model into a more detailed conceptual architecture (see examples provided in [89]). Any safety analysis method that is based on systems- and control

[2] An adequate estimation of risk is rare. Details about the problems with this approach are provided in Chapter 2.

theory and that is effective in early stages of design may be used to generate the safety analysis results that are needed to refine the initial control structure. For the many advantages that have been identified with it this thesis uses STPA for safety analysis. Hence, the conceptual architecture generation process that is described in the following follows the four basic STPA steps (see [100]). An overview of the steps to generate a conceptual architecture as outlined by Leveson [89] is given in Figure 3-1.

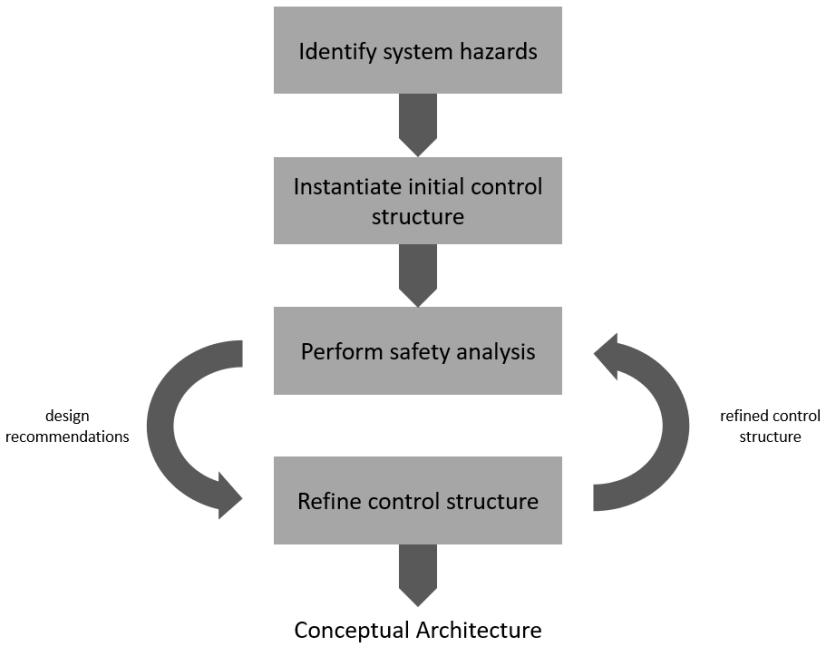


Figure 3-1: Process steps for generating a conceptual architecture.

In the first step, system hazards are identified. System hazards are system states that may lead to losses under some environmental condition(s) and hence, need to be prevented [95]. In the next step, an initial safety control structure is instantiated. The initial control structure is a highly-abstracted model of the control hierarchy in the system and is used to describe the control actions and feedback between entities in the system. Then, an STPA is performed on the initial control structure and design recommendations are generated from the UCAs and scenarios identified in the safety analysis. These design recommendations are then used to refine the initial control structure into a more detailed control structure. Multiple iterations may be needed to

arrive at a conceptual architecture that provides sufficient detail for manufacturers to start the design of a physical/logical architecture. An example demonstration of the conceptual architecture generation process is provided in the following paragraphs.

Identify system hazards

This analysis targets the safety of automated vehicles but also aims at enabling the success of automation technology. Therefore, the considered losses are not limited to causing harm to people or damage to objects but also include losses that may hurt the technology‘s reputation and ultimately success. An overview of the considered losses is provided below. Each loss is denoted by an identifier (L-) followed by a number.

- **L-1: Loss of Life or Injury of People**

  Any physical harm to one or more human bodies including minor injuries as well as death. For example, injuries such as a broken leg fall in this category.

- **L-2: Loss of or Damage to Vehicle**

  Any harm to the body of the AV including total loss of the vehicle. For example, this loss is used to describe damage to the AV‘s radar.

- **L-3: Loss of or Damage to Objects outside the Vehicle**

  Any harm to objects other than the vehicle or humans, for example, an AV that collides with a tree.

- **L-4: Loss of Transportation Mission**

  Any violation of the transportation task given to the vehicle. For example, this loss has occurred when the vehicle has missed a highway exit and can no longer satisfy timing constraints specified for the transportation task, e.g. bringing a person to the airport.

- **L-5: Loss of Customer Satisfaction**

  Any violation of the expectations of customers inside or outside the vehicle. This loss is used to describe, for example, when a vehicle makes a ride customer feel uncomfortable during transportation, e.g. by exhibiting erratic driving behavior.

- **L-6: Withdrawal of Regulatory Approval**

  Any reduction of vehicle permissions by any official regulatory institution. For example, losses such as a withdrawal of the right to drive on public roads are considered in this category.

The defined losses may occur when the automated vehicle enters a hazardous state that is matched by the environment in a way that a loss will result. The state of the environment cannot be controlled but only the vehicle states may be controlled by design. Hazardous vehicle states that may lead to the defined losses are shown below. Each hazard is denoted by an identifier (H-) followed by a number and a reference to the associated losses is provided in brackets.

- **H-1: Vehicle does not maintain a safe distance to other vehicles or objects [L-1,2,3,4,5,6]**

  Vehicle state that is reached when the AV‘s distance to other moving vehicles, static vehicles, or any objects falls below a minimum distance that is considered safe.

- **H-2: Vehicle leaves authorised or designated roadway. [L-1,2,3,4,5]**

  Vehicle state that describes when the vehicle is no longer on a roadway that complies with either its current configuration (e.g. in autonomous mode on a street for manual driving only) or planned route.

- **H-3: Vehicle exceeds its limits for controllability and stability. [L-2,5,6]**

  Vehicle state of reduced control such as lower traction on the road due to ice, excessive speed, etc.

- **H-4: Vehicle does not comply with driving laws (highway code). [L-5,6]**

  Vehicle state that is outside of the legal envelope for vehicle states, e.g. excessive speed.

Instantiate initial control structure

The initial control structure for AVs contains all the relevant social and technical parts related to the development, distribution, and operation of AVs. This includes regulators, OEMs, accident investigators, training institutions for AV operators, the automated vehicle, etc. The hierarchy and relationships between the elements in this socio-technical system are shown in the control structure in Figure 3-2.

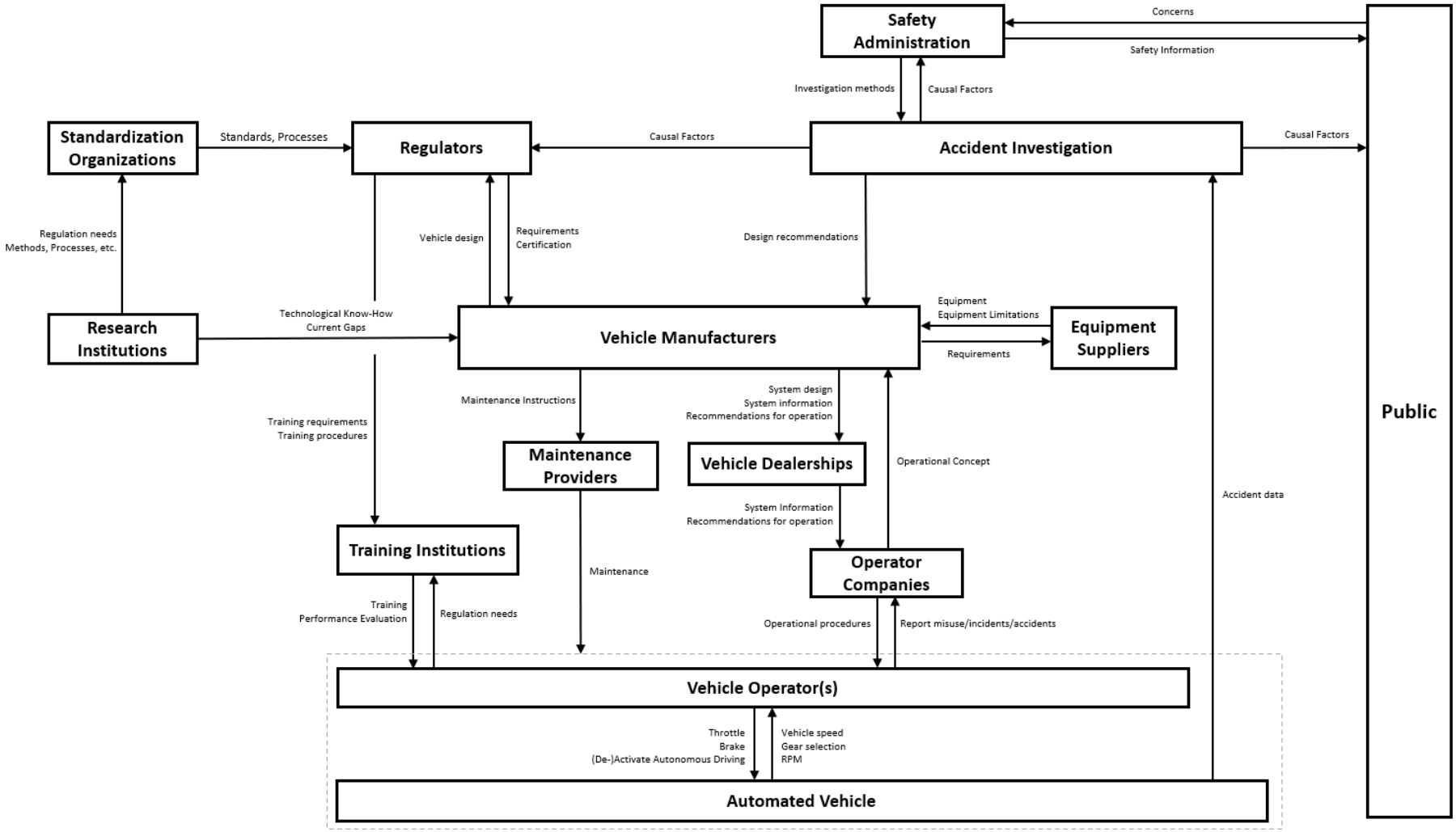


Figure 3-2: The socio-technical control structure for AVs.

The organizational structure around AVs may differ between states or countries due to local differences in federal structures, regulation, etc. However, the boxes in the control structure represent generic controllers which are associated with functions and control responsibilities. Hence, the initial control structure model is independent of specific organizational structures. The functions of each controller are described below.

Safety Administration	Overlook AV safety and provide adequate policy
Accident Investigation	Identify causal factors in accidents and provide recommendations to prevent future accidents
Regulation	Determine regulatory needs and provide frameworks to regulate the development and operation of AVs
Standardization	Identify standardization needs and provide guidelines for the development of AVs
Research	Identify current problems with AVs and communicate gaps, research needs, etc.; provide technological know-how
Vehicle Manufacturers	Create an economic benefit by developing and manufacturing AVs
Maintenance Providers	Provide maintenance service for AVs
Training Institutions	Teach operators on how to use AVs and provide feedback about their performance
Vehicle Dealerships	Create an economic benefit from purchasing and selling AVs; instruct customers on the operation of AVs
Operator Companies	Create an economic benefit from providing transportation services

Vehicle Operator(s)	Create an economic benefit from operating AVs
Equipment Suppliers	Create an economic benefit from the development and manufacturing of AV components
Public	Raise potential concerns with AVs (e.g. safety, impact on employment, etc.)

For the purpose of analyzing the safety of AVs the initial focus lies on the operational part of this control structure, see Figure 3-3.

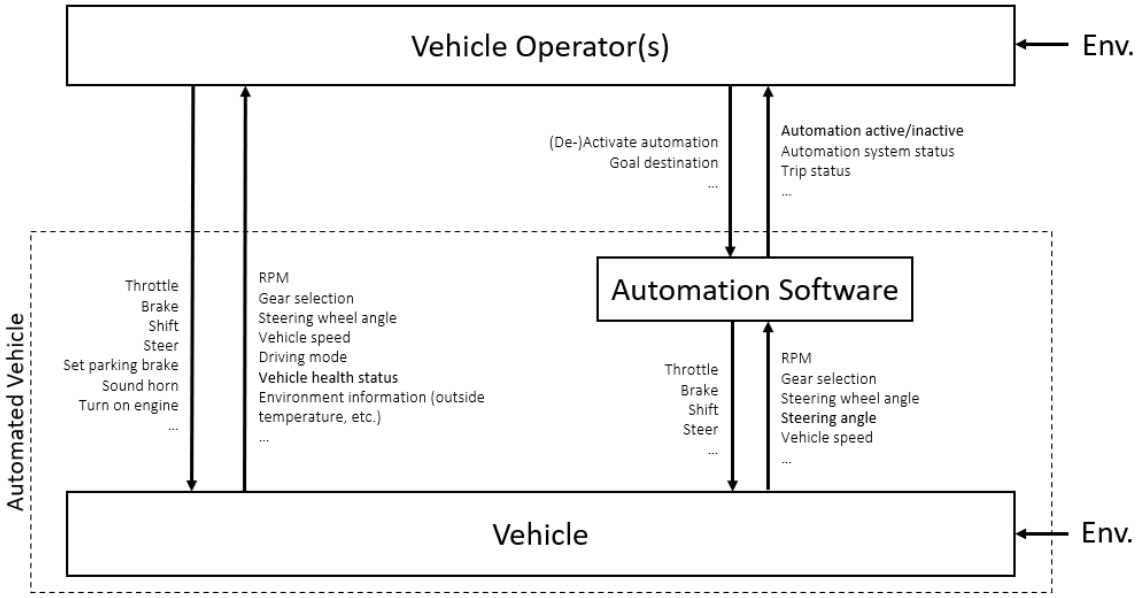


Figure 3-3: Initial operational control structure for AVs.

Perform safety analysis & refine control structure

The application of STPA to this fairly simple control structure leads to design constraints that drive the creation of the conceptual architecture. For example, the UCA *Safety Driver activates automation when the vehicle is outside the design envelope of the automation system (e.g. on icy road, city road, etc.).* identifies the problem of misuse that has been a causal factor in many accidents (see Section 2.1). A selection of the design constraints that were generated in this thesis in the application of

STPA to the initial control structure are given in Table 3.1[3]. Links to accidents from Section 2.1 that involved causal factors related to the identified design constraints are given in the footnotes of the design constraints.

Table 3.1: Design constraints generated from UCAs.

DC-1: The Safety Driver must not activate the automation when the vehicle controls have not been calibrated[4]. [UCA-40]
DC-2: The Safety Driver must provide braking when the vehicle‘s path is not clear[5]. [UCA-27]
DC-3: The Safety Driver must provide steering when automated steering is inconsistent with vehicle speed[6]. [UCA-35]
DC-4: The automation must not provide steering that exceeds the vehicle‘s limits for controllability and stability[7]. [UCA-66]
DC-5: The automation must provide braking to bring the vehicle to a safe stop whenever the vehicle has suffered a collision[8]. [UCA-69]
DC-6: The automation must not provide an excessive brake command that does not allow the following vehicle(s) to avert a collision[9]. [UCA-71]
DC-7: The Safety Driver must not activate any automation when the driver is not familiar with the system‘s configuration[10]. [UCA-43]
DC-8: The Safety Driver must not deactivate the automation when there are no means to control the vehicle[11]. [UCA-48]
Continued on next page

[3]The associated UCAs that the design constraints were generated from can be found in Appendix A. The UCAs may contain work from projects outside the scope of this thesis.
[4]see incident *June, 2018 / Newfoundland.*
[5]see, for example, accidents *May 7, 2016 / Williston, Florida* and *January 20, 2016 / China.*
[6]see incident *June, 2018 / Newfoundland.*
[7]see accident *March, 2017 / Tempe, Arizona.*
[8]see accident *May 7, 2016 / Williston, Florida.*
[9]see incidents under *Tesla: Smart Summon.*
[10]see accident *March 18, 2018 / Tempe, Arizona.*
[11]see accident *November 8, 2017 / Las Vegas, Nevada.*

Table 3.1 – continued from previous page

DC-9: The Safety Driver must not provide manual braking too late after a collision can no longer be avoided[12]. [UCA-29]
DC-10: The Safety Driver must not activate automation when the automated system is not fully functional. [UCA-45]
DC-11: The Safety Driver must not disable any required modules when automated mode is enabled. [UCA-51]
DC-12: The automation must not provide throttle too late when the vehicle‘s path is no longer clear (e.g. traffic light switched, junction blocked, etc.). [UCA-75]
DC-13: The automation must not provide throttle when the vehicle path is not clear[13]. [UCA-74]
DC-14: The automation must not provide throttle when the vehicle is unstable. [UCA-78]
DC-15: The automation must not provide excessive throttle when other vehicles or obstacles are close. [UCA-79]

While some of the above constraints may seem trivial many identify factors that have played a critical role in multiple accidents (see footnotes in Table 3.1) and may be addressed in future designs. For example, multiple accidents involved AV operators who did not identify the need for braking early enough or at all when the vehicles‘ path was not clear, e.g. due to another vehicle, pedestrians, etc. DC-2 provides the design constraint that needs to be enforced in order to prevent related accidents. Similarly, DC-7 points to a lack of driver awareness of the configuration of the automation and DC-9 to a lack of attention to the driving task, factors that have contributed to accidents such as the Uber accident in Tempe, Arizona (see accident *March 18, 2018 / Tempe, Arizona* in Section 2.1). Finally, DC-8 reveals the issue of deactivating automation when no other controls are available to control the

[12]see accident *March 18, 2018 / Tempe, Arizona*.
[13]see accident *February, 2016 / Mountain View, California*.

vehicle and points to an important causal factor in the accident of the Navya shuttle in Las Vegas (see accident *November 8, 2017 / Las Vegas, Nevada* in Section 2.1).

As can be seen, the design constraints derived from the UCAs provide useful information for refining the initial control structure. However, while the UCA-derived constraints cover many causal factors it is still unclear how the identified constraints can be enforced. The causal scenarios describe why the UCAs occurred and provide information how the identified constraints can be enforced and hence, help identify implications for the conceptual architecture. Some example causal scenarios along with the associated design/operational constraints are shown below[14]. The design constraints are denoted by an identifier *DC-* followed by a number.

UCA-74: Automation provides throttle when the vehicle‘s path is not clear (e.g. at a traffic light or junction). [H-1,4]

Scenario-1: The automation does not recognize that the vehicle‘s path is not clear. This flawed process model may be due to:

a) The automation does not receive information about obstacles around the vehicle.

b) The automation identifies a pedestrian that is about to enter the vehicle‘s path but interprets it as a dummy (e.g. because the person is not moving and is interpreted as a mock-up).

c) The feedback that the vehicle‘s path is not clear is provided only once the vehicle is close enough to the obstacles in the path (e.g. detection distance limitations of sensors).

[14]For more design constraints from causal scenarios refer to Appendix B.

DC-16: The automation must receive information about the vehicle‘s environment.

DC-17: The automation must always provide the most conservative estimate of its detections (obstacle type, speed, etc.).

Scenario-2: The automation does not account accurately for the movement of obstacles such as other vehicles, bicyclists, pedestrians, etc. This may be due to:

a) The automation design is based on inaccurate, flawed or missing prediction algorithms (e.g. assumption that pedestrians and bicyclists only cross roads on crosswalks).

b) The automation does not know its current location and hence, cannot tell if obstacles such as pedestrians or bicyclists may be expected.

DC-18: The automation must account for all potential future positions of obstacles around the vehicle‘s planned path.
DC-19: The automation must receive real-time information about its current location.

UCA-62: The automation provides steering that leads vehicle away from its destination. [H-2]

Scenario-3: The automation had to divert from its original route (e.g. due to a closed road) and kept executing the originally planned maneuvers.

a) The automation software does not have a function to re-calculate paths while driving.

b) The automation does not receive location information.

c) The automation does not have a function to derive maneuvers to follow an updated route.

DC-20: The automation must be able to re-calculate paths to its goal destination based on its current location and updates to the available road network

DC-21: The automation must receive real-time information about its current location.

DC-22: The automation must be able to detect closed roads and update the information to re-plan the route to the goal destination (e.g. update to the map of the available road network).

DC-23: The automation must have functionality to derive a maneuver sequence to follow an updated route.

Scenario-4: The automation did not know that it had diverted from its planned path and did not re-plan the route to its goal destination.

a) The automation does not receive information about its current location.

b) The automation receives inaccurate information about its current location and believes that it is still on its originally planned route.

DC-24: The automation must receive real-time information about its current location.

DC-25: The accuracy of location information must allow determining whether the vehicle has left or is still following its planned path.

These design constraints may be used simply as design requirements but may also be used for guidance in breaking down the high-level initial control structure into smaller logical components. For example, the constraint *The automation must receive real-time information about its current location* (*DC-19*, *DC-21*, and *DC-24*) indicates that the automation block in the initial control structure must have a localization component. Similarly, the constraints *The automation must receive information about the*

vehicle's environment (*DC-16*) and *The automation must account for all potential future positions of obstacles around the vehicle's planned path* (*DC-18*) identify a need for control structure elements dedicated to perception and prediction.

While the design constraints help identify required functionality through safety requirements they also provide guidance in the allocation of functions to control structure elements. For instance, if the requirements indicate that two processes need to perform their functions based on the same input data or simultaneously combining the processes into one logical component may be considered. Adapting the initial control structure based on the identified design constraints leads to a more refined control structure and ultimately, if performed iteratively, to a conceptual architecture. The conceptual architecture that results from the identified design constraints and refinements to the initial control structure is shown in Figure 3-4[15].

[15]This conceptual architecture contains work by Thomas [104].

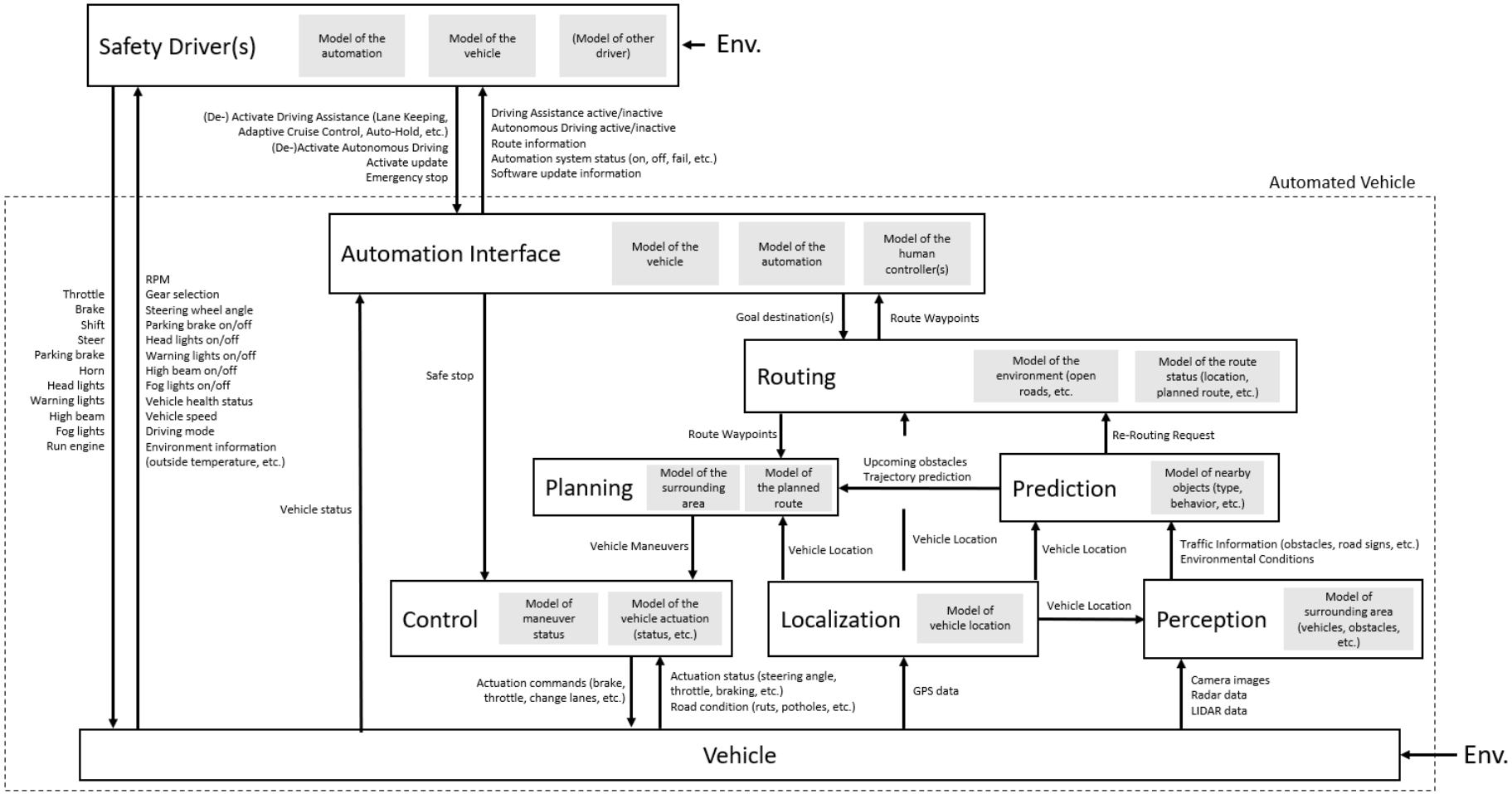


Figure 3-4: A conceptual architecture model for AVs.

The conceptual architecture contains an automation interface, modules for routing, planning, prediction, control, localization, and perception, and the vehicle. In addition, also the operator is considered a part of the AV. Hence, the conceptual architecture contains an operator crew that may have one or two safety drivers. The function of each element in the conceptual architecture is given below.

Safety Driver	Ensure safe operation of the vehicle; communicate destination, time constraints, etc.
Routing	Provide way points that define a unique route to the goal destination
Planning	Provide a sequence of vehicle maneuvers to follow the planned route
Control	Provide actuation command to execute the planned maneuvers
Localization	Identify and provide the vehicle‘s location
Prediction	Provide an estimate of the future position of nearby objects
Perception	Provide information about the vehicle‘s environment
Automation Interface	Provide operators/customers with ways to interact with the automation; communicate goals to automation
Vehicle	Provide physical space for transportation; provide vehicle hardware

3.3 Safety Analysis and Identification of Safety Requirements

In this section the conceptual architecture is used to derive safety requirements. The first two steps of STPA *Define Purpose of the Analysis* and *Model the Control Structure* have already been performed when the conceptual architecture was derived and remain the same. Following the remaining two steps of STPA first Unsafe Control Actions (UCAs) are identified and safety requirements are derived. Then, the identified UCAs are used to derive causal scenarios and detailed safety requirements.

3.3.1 STPA Step 3: Identify Unsafe Control Actions

This subsection presents some examples of the UCAs that were identified in the scope of this thesis (see Table 3.2 and Table 3.3)[16]. The UCAs are organized in tables with a table for each controller.

[16]The UCAs presented in this thesis contain work from research projects with Zenuity and the Renesas Electronics Corporation.

Table 3.2: Unsafe Control Actions – Automation.

Control Action	**Not Providing**	**Providing**	**Provided Too Early / Too Late**	**Provided For Too Long / Stopped Too Soon**
brake	Automation does not provide a brake command after the vehicle has suffered a collision and the vehicle must stop. (UCA-1) [H-1,2,3]	Automation provides excessive braking when a following vehicle cannot decelerate in time. (UCA-2) [H-1]	Automation stops providing brake command too early when vehicle speed is still above speed limits (e.g traffic flow limit). (UCA-3) [H-1,2,3,4]	Automation stops providing brake command too soon when a collision is imminent. (UCA-4) [H-1]
throttle	Automation does not provide throttle when a speed increase is required to avoid a rear or side collision. (UCA-5) [H-1]	Automation provides throttle when vehicle is under maintenance. (UCA-6) [H-1,3]	Automation provides throttle too early before the vehicle's path has become clear (e.g. dynamic obstacles slower than anticipated, etc.). (UCA-7) [H-1,4]	Automation provides throttle for too long until forward collision becomes imminent. (UCA-8) [H-1]
steering	Automation does not provide steering when the vehicle‘s lane is about to end. (UCA-9) [H-1]	Automation provides steering when it steers the vehicle into a collision (e.g. lane keeping feature). (UCA-10) [H-1]	Automation provides steering too late after merging is no longer possible (e.g. gap closed by another vehicle). (UCA-11) [H-1,4]	Automation stops providing steering too early before the maneuver is finished. (UCA-12) [H-1,4]

Table 3.3: Unsafe Control Actions – Safety Driver.

Control Action	**Not Providing**	**Providing**	**Provided Too Early / Too Late**	**Provided For Too Long / Stopped Too Soon**
activate automated driving	Safety Driver does not (re-)activate automation after an override has occurred. (UCA-13) [H-1,2,3,4]	Safety Driver activates automation when vehicle is in an inadequate configuration (e.g. component just replaced but not re-calibrated). (UCA-14) [H-1,2,3]	Safety Driver activates automation too early before driver has given up manual control. (UCA-15) [H-1,2,3]	
deactivate automated driving	Safety Driver does not deactivate automation when the driver is about to leave the vehicle (e.g. to provide first aid in an accident ahead). (UCA-16) [H-1,2,3]	Safety Driver deactivates automation when the vehicle is performing a maneuver (e.g. passing another vehicle). (UCA-17) [H-1,2,4]	Safety Driver deactivates automation too early before the driver is prepared to resume manual control. (UCA-18) [H-1,2,3]	

Continued on next page

Table 3.3: Unsafe Control Actions – Safety Driver (continued).

Control Action	**Not Providing**	**Providing**	**Provided Too Early / Too Late**	**Provided For Too Long / Stopped Too Soon**
activate lane keeping		Safety Driver activates lane keeping when lane markers are no longer valid (e.g. building site with new lanes in different color). (UCA-19) [H-1,2,3,4]	Safety Driver activates lane keeping assistance too late when the vehicle has already entered another lane. (UCA-20) [H-1,2]	
deactivate lane keeping	Safety Driver does not deactivate lane keeping when the lane markers are worn out. (UCA-21) [H-1,2,3,4]	Safety Driver deactivates lane keeping when the driver does not control steering. (UCA-22) [H-1,2,4]	Safety Driver deactivates lane keeping too late after the driver has stopped to steer the vehicle. (UCA-23) [H-1,2,4]	
emergency stop	Safety Driver does not provide emergency stop when a manual override is not effective. (UCA-24) [H-1,2,3]	Safety Driver provides emergency stop when the driver is not prepared to resume manual control. (UCA-25) [H-1,2,3]	Safety Driver provides emergency stop too late after the vehicle does not detect an obstacle in or near the vehicle's path. (UCA-26) [H-1,2,3]	

3.3.2 STPA Step 4: Identify Scenarios and Safety Requirements

This subsection analyzes why the identified UCAs may occur by determining causal scenarios and derives safety requirements to prevent the identified UCAs. In the following a few examples are given of the causal scenarios and safety requirements that were found in this thesis.

UCA-1: Automation does not provide a brake command after the vehicle has suffered a collision and the vehicle must stop. [H-1,2,3]

Scenario-5: The automation does not recognize that the vehicle was involved in a collision. This flawed process model may result from one or more of the following:

a) The automation receives incorrect feedback about the vehicle‘s state (e.g. that the vehicle has not suffered a crash or that the vehicle‘s integrity has not been compromised).

b) The automation receives feedback that the vehicle has been involved in a collision but discounts the feedback because it assumes that the associated sensors have been compromised during the collision.

c) The automation does not receive any feedback about whether the vehicle has suffered a collision (e.g. not considered in design).

d) The feedback that the vehicle has suffered a collision is received by the automation after parts of the automation system have been shut down due to the damages they have suffered during the crash.

R-1: The automation must always receive feedback when the vehicle has suffered a collision.
R-2: The automation must never discount feedback about a collision of the vehicle.
R-3: Feedback about a collision of the vehicle must be robust against collisions and before parts of the automation are shut down due to their damages.

Scenario-6: The automation recognizes the collision but decides that it is safer to keep travelling. This flawed control algorithm may occur because:

a) The automation software is programmed to evaluate risk and to transition to a minimal risk condition after a collision. As a result, the vehicle decides that it is safer to travel to the next safe stopping point off the highway rather than stopping on the highway and continues to the next exit.

b) The automation software was programmed to forego providing brake commands after detecting a collision in order to prevent false positives, e.g. on highways when there is no collision, but the vehicle detects a collision. However, as collision detection systems mature and achieve higher dependability, this behavior is no longer justified.

c) The automation software was programmed incorrectly and there is an error in the automation software.

R-4: The automation must never make decisions based on estimated risk after the vehicle has suffered a collision.
R-5: The vehicle must never continue moving after it has suffered a collision except to the side of the road.
R-6: The automation must never forego brake commands to slow down after a collision has been detected.
R-7: The certification of automation software must ensure that adequate development practices were used to minimize software errors.

Scenario-7: The automation was re-activated after the collision by the driver, e.g. because the driver was unaware that the vehicle is no longer in automated mode and wanted to switch it off.

R-8: The driver must be informed whenever the vehicle is in automated mode.
R-9: The automation must not allow re-activation after a collision until the vehicle has been checked by maintenance.

Scenario-8: The automation provides a brake command, but the brakes do not apply. This inadequate control execution may result from the following:

a) The communication channel from the automation system to the braking system is broken (e.g. damaged during collision).

b) The braking system is compromised and does not apply the braking actuators (e.g. disconnected from power during the collision).

c) The brake actuators are compromised (e.g. jammed, disconnected from actuation power source, etc.)

R-10: The communication channel between the automation and the braking system must never prevent braking.
R-11: Braking must never be prevented by a compromised braking system (including brake actuators).

Scenario-9: The automation provides a brake command that is executed by the brakes, but the vehicle speed does not decrease. This inadequate process behavior may occur because:

a) The vehicle is on slippery ground that makes braking ineffective (e.g. wet or icy road).

b) The vehicle‘s engine is stuck on full thrust and counteracts the braking action.

R-12: The vehicle must be able to brake on slippery underground.

R-13: The vehicle‘s braking force must exceed its propulsive power.

Scenario-10: The automation cannot provide a brake command because the required vehicle systems have been compromised during the collision.

R-14: The vehicle must come to a safe stop once its braking system becomes impaired.

UCA-19: Safety Driver does not deactivate lane keeping when the lane markers are worn out. [H-1,2,3,4]

Scenario-11: The Safety Driver is not aware that lane keeping is still activated. The driver‘s flawed process model may be due to the following:

a) The Safety Driver receives incorrect feedback about the lane keeping feature, e.g. that the system is no longer activated.

b) The Safety Driver receives feedback from the automation to take over control but assumes that the system has shut itself off after the take-over request.

c) The Safety Driver does not receive feedback on whether the lane keeping feature is activated (e.g. no cockpit indication, no steering wheel movement, etc.)

d) Feedback is provided to the Safety Driver through a centralized system and it takes the driver too much time to locate feedback about the lane keeping system (e.g. for navigating to the right page on the screen of the automation interface).

R-15: The Safety Driver must always be informed when the lane keeping feature is activated.
R-16: The vehicle must never provide feedback that the lane keeping feature is deactivated when it is activated.
R-17: The vehicle must provide drivers fast access to feedback on whether the lane keeping feature is activated at all times.

Scenario-12: The lane keeping feature does not turn off when the Safety Driver deactivates the lane keeping feature. This inadequate control execution may be due to the following factors:

a) The communication between the driver interface and the automation is compromised (e.g. activation button is broken).

b) The automation does not react to the deactivation command (e.g. in an ignore mode, software bug, etc.)

c) The automation system sends a command to switch off the lane keeping feature but the feature stays activated (e.g. stuck in a certain mode, etc.)

R-18: The lane keeping feature must automatically turn off when the communication channel between automation and the driver interface becomes impaired.
R-19: The automation must never ignore deactivation commands in any mode.
R-20: The automation must automatically shut down once it is no longer cooperative (e.g. when stuck in certain mode, etc.)

Scenario-13: The Safety Driver deactivates the lane keeping features but the vehicle keeps following the worn out lane markers. This inadequate process behavior may occur because:

a) The vehicle‘s wheels are travelling in ruts in the lane that guide it along the worn out lane markers.

R-21: The vehicle must not rely on the driver to inhibit following ruts in the road.

Scenario-14: The AV operator company exchanged the Safety Drivers between vehicles (e.g. as part of job rotation) and did not account for vehicle differences such as different levels of system maturity in different vehicles.

R-22: Commercial operators of AV fleets must ensure that driver training matches their vehicle.

Scenario-15: The Safety Driver is aware of the worn out lanes and that the lane keeping feature is activated but decides to keep the feature activated. This flawed control algorithm may result because:

a) The Safety Driver is not aware of the limitations of the system (e.g. did not receive training or training only covered full automation).

b) The Safety Driver was taught to keep the lane keeping feature activated when there are worn out lane markers, e.g. in order to help the company collect miles with automation features activated.

c) The Safety Driver was taught that the feature is backed up by road information (e.g. on-board map, geo-fencing, etc.) and that worn out lane markers will not impact the feature when the map is no longer available/accurate (e.g. road has changed, feature no longer relies on a map, etc.).

R-23: The drivers of AVs must receive training on AV limitations and on how to manage the limitations in operation.
R-24: Safety Drivers must be informed whenever an update is performed on their AV.

Chapter 4

Implementation through Certification

This chapter describes how the suggested safety approach based on a conceptual architecture model may be implemented through certification. Section 4.1 describes the role of certification and outlines how it can help solve the currently existing safety-related problems in the AV field. Subsections 4.1.1 and 4.1.2 detail how the conceptual architecture model from Chapter 3 can be applied to regulation and insurance and illustrate the associated benefits. An organizational structure and processes to implement model-based certification are derived in Section 4.2.

4.1 Certification & Relationship to Problems in the AV Field

The goal of using a model-based approach to safety is to address the problems that are currently holding back progress and the large-scale deployment of AVs. As mentioned earlier, many of the currently existing problems are safety-related and may be addressed using with the conceptual architecture model from Chapter 3. However, the safety problem with AVs may not be solved by prescribing a conceptual architecture and model-based safety analysis alone but must also be enforced on an organizational level (see [95, Ch. 12, Ch. 13], [92], and organizational part of the initial control structure in Figure 3-2).

Certification ties together the operational and organizational parts of the control structure and hence, may help solve many of today‘s problems with AVs (see Section 1). For example, certification may help reassure that AVs are safe despite their high complexity that challenges manufacturers, help the public build trust in AVs, and resolve issues with respect to their social acceptance. Further, certification may be useful in delineating the responsibilities of drivers and manufacturers and provide guidance in liability questions. The following subsections describe how certification based on the conceptual architecture model from Chapter 3 may be used for and benefit regulation and the insurance of AVs.

4.1.1 Regulation of Automated Vehicles

The many expected benefits from the increased use of automation technology have aroused the interest of consumers and manufacturers and have created a very competitive and dynamic market. While technological progress continues without restraint it creates a challenging problem for regulation ([105], [15]).

Today already vehicles contain between 100 and 150 million lines of code, albeit not fully autonomous yet, and make it increasingly difficult to understand the software in its entirety and assess the overall safety of the vehicle. Further, in the competition on the AV market each manufacturer is pushing its own technology and the market is flooded with a variety of AV- and software-designs that create an even bigger burden on regulation (see [106, p. 10, p. 24, p. 53], [107, p. 21], [108]).

With the great amount of flexibility that software allows additional variations are possible not only between vehicle models but even individual vehicles can be customized depending on the demands of customers. In addition, vehicle capabilities may be tuned, customized, upgraded during a vehicle‘s life-cycle, e.g. by so-called “over the air“ updates (see [109]) or through the rising app market for AVs. Finally, system adaptations are no longer limited to official changes by manufacturers or maintenance

facilities. Vehicle owners may have their own ideas and may be able to tweak their vehicles, e.g. to increase the vehicle‘s maximum speed, etc.

Hence, the challenges of regulation are an overwhelming (but still increasing) amount of software in vehicles, an unmanageable amount of product and software variations, and software changes that may occur over time and may be hard to trace when done by private persons. These challenges are stretching the limits of regulatory approaches that are currently used for conventional vehicles and make regulation of AVs a separate important problem. The following paragraphs show how these regulation problems may be addressed by using an approach based on a conceptual architecture.

The example safety analysis in Chapter 3 has demonstrated a conceptual architecture can be derived from a generic control structure for AVs and how a new regulatory process may adopt this strategy of separating safety analysis from development and use safety requirements as an input for manufacturers and as a baseline for certification. Manufacturers can then develop their design against the conceptual architecture and safety requirements and seek certification by demonstrating compliance. Such a requirements-driven approach offers multiple advantages that are described in the following.

First, regulation based on a conceptual architecture allows a separation of concerns. By definition, manufacturers focus on design and have expertise in creating designs based on a given set of requirements. Once a design has been created manufacturers apply their experience in testing, simulation, etc. to verify the design against a given set of requirements (see [110], [91, p. 24]). However, manufacturers often lack the resources (e.g. workforce, methods, expertise, or experience) to ensure the validity and completeness of the requirements that drive their designs ([30], [91, Ch. 4]). As a result, many projects suffer delays or budget overruns due to requirement-related issues ([30], [91, Ch. 5, Ch. 6], [96], [95]). Often, emergent system properties such as safety find their way into the requirements no earlier than in the final stages of

development when manufacturers start to consider safety and problems are difficult and expensive to resolve ([89], [26], [90], [88]).

In contrast, regulators often do not have the detailed technical knowledge or expertise in creating and verifying designs but bear responsibility for safety. These issues can be addressed with regulation based on a conceptual architecture. Using a conceptual architecture along with a set of safety requirements manufacturers can no longer miss to include safety early in their development process but can leverage the provided safety requirements to drive their development. Manufacturers can focus on their strengths of creating and verifying designs and are free in their technical solution as long as they can verify that their design meets the conceptual architecture and safety requirements.

Similarly, regulators can also focus on their responsibility for safety and do not have to understand individual designs in order to make a safety assessment. Often, regulators do not have the resources (e.g. personnel and time) to understand every design in detail within the time constraints of the market (see challenges mentioned in [111, p. 3, p. 7, p. 23]). With a conceptual architecture-driven approach to certification the task of safety analysis, identification of safety requirements, and the derivation of a conceptual architecture may be "outsourced" to third party experts that may have better resources than manufacturers or regulators. In addition, it is possible to include expertise from multiple sources into the identification of safety requirements. For example, safety experts may bring in their expertise in identifying hazards for highly-complex software-intense systems (e.g. by using methods such as STPA) manufacturers may provide help to derive a conceptual architecture, while operators, the public, consumer unions, accident investigators, etc., may raise problems observed in AV operation, accidents, or general concerns.

Hence, regulation based on a conceptual architecture model allows all involved parties to focus on their individual concerns and strengths while, as a by-product, regulation

enforces that safety is considered early in development.

Second, regulation based on a (generic) conceptual architecture reduces the effort for regulators to a more manageable amount. Today already, the level of complexity, product variation, and flexibility to introduce changes simply by a software update in automotive systems creates an overwhelming burden for regulators (see description of regulatory challenges in [112], [113], [114]). With more automation to come, up to fully autonomous vehicles, there is little confidence that regulation will be able to keep up with the current rate of technological progress. At the same time, as automation technology spreads across industries and products become more and more interconnected (e.g. the IPhone and Tesla‘s Smart Summon feature ([49], [51])) the impact on our daily lives is rising and creates an even bigger need for regulation.

The suggested generic, model-based approach to regulation via a conceptual architecture reliefs this bottleneck in regulation by providing a generic solution to the safety problem that significantly reduces effort. While regulators can focus on one generic safety analysis (that, in turn, may be more thorough too) rather than certifying each AV design individually manufacturers also benefit. For manufacturers, certification will no longer create an extra effort but will be integrated as safety requirements that flow through the development process along with other design requirements from the beginning.

Hence, the suggested conceptual architecture-driven approach to regulation significantly reduces regulation effort, uses the resources of manufacturers and regulators more effectively, and provides an alternative to address the challenges of regulation to come.

Third, a regulation approach based on a conceptual architecture will decrease development cost for AV manufacturers. AVs contain some of the most advanced technology today and the future of AVs rests on further technological progress. As a result, man-

ufacturers have made significant investments in research for AVs ([115], [10], [11]) and AV development is currently characterized by trial and error and testing. The iterative nature of this development approach is cost-intensive as manufacturers have to assess safety and seek certification for each of their designs ([116], [117], [118]).

In contrast, regulation based on a conceptual architecture and safety requirements saves manufacturers the cost of performing safety analyses for each design individually. Safety analysis is centralized and the results of the analysis are given to manufacturers before the start of development. This not only frees manufacturers up from performing safety analyses and the associated cost but also protects manufacturers from late and costly design changes as a result of identifying safety problems only in late stages of development. In addition, cost benefits for manufacturers may arise from being able to use the conceptual architecture as a baseline concept for a series of their products. For example, different vehicle models may be designed based on the same conceptual architecture and may be certified using the same or similar verification techniques.

Further, another cost advantage for manufacturers is better longevity of system designs. While today previous system designs can often no longer be used due to the missing traceability between design artifacts and requirements introduced by the late safety "add-on's" they have often received (see [91], [89], [90], [26]), a conceptual architecture-driven approach to regulation and development will provide more traceability and guidance during early design. As a result, previous designs will no longer be difficult to understand and too complex to build on and may be re-used for future designs.

Finally, manufacturers will also benefit from the increased safety of their vehicles as their expenses for liability cases are likely to decrease. The development of AVs based on a set of safety requirements from regulators will lead to safer AVs, less losses[1], and

[1]It should be noted that the definition of losses may include even more than just accidents. For

hence, less customer complaints and lawsuits.

In summary, using a conceptual architecture model for regulation offers manufacturers a variety of cost benefits in AV development.

4.1.2 Insurance of Automated Vehicles

The trend to more automation in automotive vehicles comes with a shift in the characteristics of the insurance problem that is predicted to revolutionize the automotive insurance sector [119]. For one, cyber-security will play a more central role in the insurance of AVs than with less automated vehicles. Increased automation has lead to higher degrees of connectivity, both within individual vehicles as well as between vehicles and between vehicles and infrastructure. Cyber-security is no longer a problem that is limited to revealing sensitive user information but also includes potential hacking and remote control of AVs.

Another important issue for insurance providers is liability assignment. With more and more automation it is difficult to blame drivers alone for accidents and so far there are no arrangements that delineate the responsibilities of OEMs, software suppliers, and operators.

Further insurance challenges arise with insurance coverage. With AVs it is often difficult to assign clear responsibility in accidents, especially when official regulations are missing. As a result, claims may take longer to follow through with and insurance companies are already considering alternatives. In the UK, for example, insurance providers have adopted a single cover approach: Victims of collisions are compensated regardless of fault and insurance companies are taking the task of following up with faulty parties to recover funds [120].

Ultimately, insurance challenges arise with insurance pricing. As a result of higher

example, customer satisfaction may also be considered.

vehicle complexity insurance providers expect that more specialized skills will be needed in the repair of AVs that, in turn, will result in higher costs. In addition, while insurance companies often draw on driving data to calculate insurance pricing to date enough data is not available for AVs[2]. With insufficient data it is difficult for insurers to estimate the frequency and severity of accidents and conventional models of insurance pricing will be challenged. In conclusion, the many challenges with the insurance of AVs indicate a need for a new insurance strategy.

Many insurance companies are currently in search for new insurance concepts to overcome the mentioned problems ([120], [112], [119], [121]). The conceptual architecture model suggested in this thesis can serve as a basis for such new insurance concepts and provides an opportunity to solve insurance-related problems with AVs. As OEMs create their AV designs based on a common conceptual architecture and safety requirements it gives insurance providers a reference for their business model. For instance, while the conceptual architecture may be used as a reference in legal cases, the losses defined in the generic safety analysis offer a way for insurers to estimate the severity of potential accidents. In addition, with the conceptual architecture and safety requirements determined ahead of design there is an opportunity for insurers to influence the design of AVs and integrate insurance-relevant features into vehicles. The key advantages of a conceptual architecture-driven approach to insurance are outlined in the following paragraphs.

For one, an insurance approach based on a conceptual architecture and safety requirements provides better guidance in liability questions. As the level of automation in vehicles increases, assigning fault in accidents has become more difficult (see [61], [36]). Complex interactions between drivers and vehicles obscure the boundaries between the responsibility of drivers and vehicle manufacturers. Further, only few AV OEMs write their own AV software and often additional parties such as software suppliers are involved. To date there is no guidance on how to determine responsibility in

[2]Insurances need at least 100 to 150 billion miles of driving data as a statistical basis [120].

automation accidents and past cases have shown that assignment of liability currently tends to be incomprehensible (see [61]). A conceptual architecture provides better guidance on the responsibilities of manufacturers and operators, and may even help determine the role of software suppliers. For example, liability may be assigned based on whether an AV satisfies the conceptual architecture and safety requirements. As a result, the suggested approach may resolve problems related to liability and help overcome one of the major blocking points for the insurance of AVs.

Second, with the ability to estimate the severity of potential accidents insurance companies may be able to develop a model for insurance pricing and coverage. Before a conceptual architecture is derived and safety requirements are identified a generic safety analysis has to be performed (see Chapter 3). The losses defined at the beginning of this safety analysis can be used by insurance companies to estimate the severity of potential accidents. These estimates can then be used to calculate the cost that insurances will have to bear and determine insurance pricing as well as coverage. In addition, insurance companies may be able to contribute in the process of identifying a conceptual architecture and safety requirements to avoid accidents of high severity or cost. For example, insurance companies could provide additional requirements to protect high cost vehicle components or define losses for the safety analysis to limit accident severity, reduce the financial loss in potential accidents, and ultimately reduce insurance cost. Further, different pricing models/policies based on the fulfillment of safety requirements may be possible. Therefore, an insurance strategy based on a conceptual architecture model and safety requirements will help insurance providers to identify effective options for the insurance of AVs.

Finally, with the opportunity to contribute to requirements insurance providers may be able to offer insurance options before billions of miles of automated driving data are available. Today already, insurance companies are offering discounts to customers who agree to vehicle tracking, driver monitoring, and capturing other insurance-relevant data ([120], [119]). If insurance companies can now contribute their own requirements

to ensure vehicles provide the information that insurances need they may be able to collect insurance-relevant vehicle data more effectively. This, in turn will not only help insurance providers with claims [119], e.g. by providing data from accidents, but will also enable insurances to provide insurance options for AVs before automated vehicles have built up enough statistical data for an insurance concept. As a result, new insurance models that do not rely on statistics may be possible. To summarize, a conceptual architecture model may help get insurances more involved in the development of AVs and may offer alternative ways for insurance providers to form an insurance concept.

4.2 Derivation of an Organizational Structure and Processes for Model-Based Certification

This section provides an example implementation of the suggested model-based certification. First, the methodology from Section 3.2 is applied to the organizational part of the initial control structure (see Figure 3-2 in Section 3.2) and a suitable organizational structure for a model-based approach to certification is derived. Then, the derived organizational control structure is used to identify processes to involve certification throughout the AV life-cycle.

The suggested model-based approach to certification brings along many changes in the responsibilities and interactions between stakeholders in the AV field (see Section 4.1.1 and 4.1.2). For example, as manufacturers are no longer responsible for safety analysis and presenting evidence for the safety of their design, certification is no longer limited to interactions between manufacturers and regulators. Rather, with a model-based certification approach the certification effort is split between multiple entities such as regulators, safety analysis providers, and verification institutions. These and other changes have implications on the organizational part of the control structure. The

conceptual architecture generation process demonstrated in Section 3.2 provides a systematic way to identify unsafe control in the organizational structure and is used in this thesis to derive a control structure that can facilitate a model-based certification. Some examples of the organizational constraints that were identified in the scope of this thesis and their implications on the organizational control structure are given below[3].

DC-26: The results of the safety analysis must not impact the analyst‘s economic benefit. [Scenario-21]
DC-27: AV manufacturers must receive information about causal factors in previous accidents with AVs. [Scenario-22]
DC-28: Problems in the operation of AVs must lead to a re-evaluation of the operational concept and procedures. [Scenario-23,24]
DC-29: Operator companies must receive direct feedback about their companies‘ accidents, incidents, and issues in operation. [Scenario-25]

Design constraint DC-26 points to the need for independence between safety analysis and the economic benefits of the conductor of the analysis and calls for a separate, manufacturer-independent safety analysis element in the control structure. Constraint DC-27 implies that accidents with AVs must be followed up by an accident analysis and the identified causal factors must be distributed in order to prevent the re-occurrence of causal factors in AV accidents. Hence, a centralized control structure component is necessary to supervise the analysis of AV accidents, keep track of the identified causal factors, and make them available to manufacturers and others.

Recommendation DC-28 and DC-29 both emphasize the need for feedback from operations. While recommendation DC-28 refers to feedback from operations to a centralized, higher-level control structure element that can order a re-evaluation of the operational concepts and procedures of AVs, recommendation DC-29 suggests that

[3]The UCAs and scenarios to the constraints are given in Appendix C.

operator companies have their own, direct feedback channel from the operation of their vehicles to prevent companies from dismissing feedback (e.g. because they assume the feedback comes from other operator companies and does not apply to their operations). The organizational control structure that results from refining the organizational part of the initial control structure is shown in Figure 4-1.

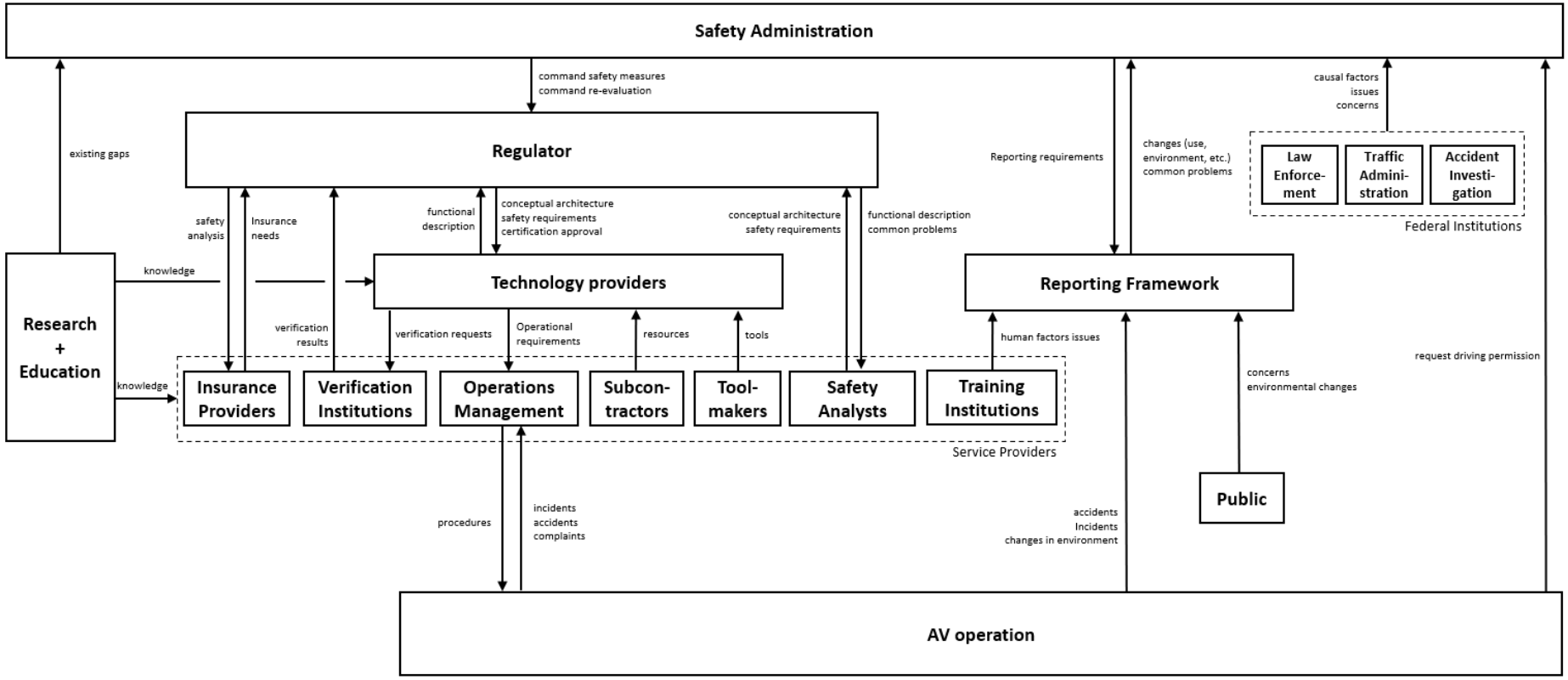


Figure 4-1: Organizational structure for model-based certification.

Regulators are the central element of the organizational structure for model-based certification. However, regulators are only responsible for managing the certification process and no longer have to review the detailed designs of manufacturers. Similarly, technology providers such as OEMs, suppliers, and other innovators also benefit from the shift in responsibilities that comes with a model-based approach to certification and can focus on development rather than being responsible for safety analysis and preparing documentation for certification. The certification effort is carried out by safety analysts and verification institutions. Safety analysts are third parties that derive a conceptual architecture and safety requirements based on a description of a system's functionality[4]. Similarly, verification institutions are third parties that provide services such as testing, simulation, etc. to verify the compliance of a product with safety requirements.

Verification institutions may be supported by research institutions with knowledge about common gaps in state-of-the-art automation and suitable verification methods while safety analysts may also be supplied with information from the regulator. For example, regulators may provide safety analysts with information about currently observed issues and help analysts include these issues in the analysis. Regulators may receive this information from the safety administration that collects information about causal factors in accidents, issues observed in operation, and concerns from federal institutions such as accident investigation, law enforcement (e.g. police, traffic wardens, etc.), and traffic administration (e.g. highway administration).

In addition to the information from federal institutions the safety administration may receive feedback about the use of AVs, their operating environment, observed issues, and any changes in the operation of AVs through a reporting framework. The reporting framework is an interface for the public, training institutions, and stakeholders involved in the operation of AVs to report concerns, human factors issues, accidents,

[4]For example, institutions such as NCAP may be considered (see `https://www.euroncap.com/en`).

incidents, or any changes that may affect the safe operation of AVs. Based on the information that the safety administration receives the safety administration may require the regulator to take safety measures or require a safety re-evaluation to identify potentially necessary re-certification efforts.

Finally, insurance providers are responsible for providing insurance-relevant requirements to the regulator and building insurance models based on safety analysis data (safety requirements and associated hazards) received from the regulator. Subcontractors and toolmakers may support technology providers with the resources and tools needed for development while operations management implements the operational requirements from technology providers as operational procedures. In turn, operations management receives direct feedback from AV operations about accidents, incidents, and complaints. An overview of the responsibilities in the organizational control structure for model-based certification is given below.

Regulators	Manage the certification process
Technology Providers	Develop technical solutions based on safety requirements
Safety Analysts	Provide conceptual architecture and safety requirements
Verification Institutions	Devise verification procedures to requirements; conduct and document verification activities
Safety Administration	Coordinate safety efforts to protect the public
Law Enforcement	Report observed issues and potential concerns
Traffic Administration	Report issues observed in traffic
Accident Investigation	Identify causal factors in accidents

Insurance Providers	Identify insurance-relevant requirements; provide insurance based on satisfaction of safety requirements
Research + Education	Raise awareness of technical gaps and safety issues; generate knowledge and provide education
Training Institutions	Report issues observed in operator training
Reporting Framework	Provide means for feedback from operations
Public	Raise concerns of the public
Operations Management	Develop operational procedures; monitor operations and adapt procedures
Subcontractors	Provide resources to technology providers
Toolmakers	Provide tools for technology providers

The goal of regulation is to protect the public from hazards and ensure that a minimum level of safety is met [79]. Safety is an emergent system property that needs to be controlled throughout the system life-cycle ([99], [95]). Any changes to a system, technical, social, or changes in the operating environment may impact safety and need to be controlled ([79], [78]). AVs in particular, are subject to a dynamic operating environment and provide a large amount of flexibility with respect to their functionality. For instance, AVs may be modified by software updates offered through manufacturer upgrades or AV apps. Further, while the operation of AVs is currently limited to a few locations [112], if progress is made, AVs may spread quickly around the globe and may soon be operated in environments beyond their originally intended scope.

Finally, due to their ability to receive software updates, AVs may have longer life-cycles and may experience more change in their operating environment. For instance,

owners may decide to keep their vehicle longer because the software update to bring a vehicle up-to-date may be cheaper than buying a new vehicle. As a result, AVs that were originally designed for one environment are then operated in a different environment that has evolved over time.

Hence, certification must not cover initial development only but must monitor and control any changes that occur throughout the life-cycle of all deployed AVs. In the following the organizational structure from Figure 4-1 is used to derive the required processes to involve certification throughout the AV life-cycle and ensure AV safety.

The first task of certification is to regulate new developments. Before manufacturers can release their AVs on public roads they need to demonstrate that their design is acceptably safe. In contrast, regulators are responsible for ensuring a minimum level of safety by granting certification based on an evaluation of the safety of AV designs. These responsibilities are currently addressed by assigning the responsibility for safety analysis and safety documentation to manufacturers and the responsibility for review of the documentation and approval for certification to regulators. As a result, manufacturers are usually following the V-model for development and submit documentation of development artifacts such as requirements, verification results, etc. at the end of the process (see [110]). In contrast, a model-based certification approach leads to a shift not only in responsibilities but also the steps in the certification process. Figure 4-2 highlights the process steps for a model-based certification in the control structure from Figure 4-1.

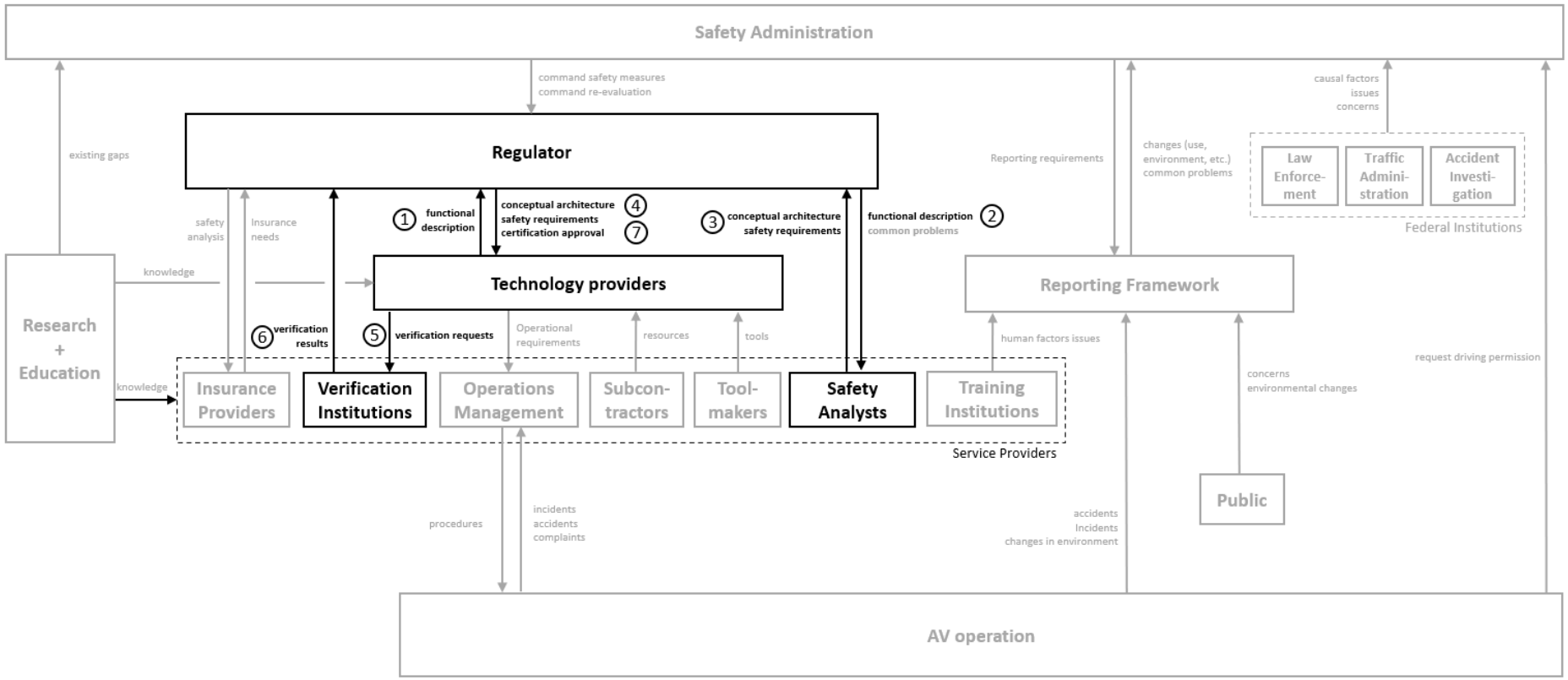


Figure 4-2: Process steps for the certification of new developments.

In the first step, manufacturers provide the regulator with a functional description of their system (see ① in Figure 4-2). The functional description lays out the basic purpose and system and may be derived from a Concept of Operations that the manufacturer may already have prepared. In the second step (see step ②), the regulator provides the functional description to a safety analyst to create a conceptual architecture and identify safety requirements. Then, the safety analyst follows the methodology outlined in Section 3.2 to derive a conceptual architecture, identifies safety requirements (see methodology in Section 3.3), and provides those to the regulator (see step ③).

The regulator shares the identified safety requirements and conceptual architecture with the manufacturer as a basis for development (step ④). The manufacturer, then, designs the product based on the conceptual architecture and safety requirements and requests a verification from a verification institution (step ⑤). The verification institution conducts the necessary tests, simulations, etc., and provides the regulator with the verification results (step ⑥). Based on the verification results the regulator decides if certification approval is granted and provides manufacturers with the results of the approval decision (step ⑦).

Another important task of certification is the regulation of design changes such as upgrades, updates, etc ([105], [95]). Design changes may be initiated by OEMs, e.g. to offer additional functionality or keep vehicles up-to-date, by suppliers that offer new or improved technology such as more reliable cameras, software, etc. or by other innovators such as AV app developers, etc. While it is obvious that an update to the automated driving software may manipulate vehicle behavior and impact safety AV apps that do not change safety-critical functions themselves may also have an impact on safety in interaction with other apps on the vehicle. However, verification of all possible combinations of AV designs and apps is practically infeasible.

Nevertheless, STPA can help identify requirements and point to critical test cases.

The organizational structure shown in Figure 4-1 provides a framework to leverage STPA through the suggested model-based certification approach and introduce changes in a systematic way with regulatory oversight. The process steps for the certification of design changes are outlined in Figure 4-3.

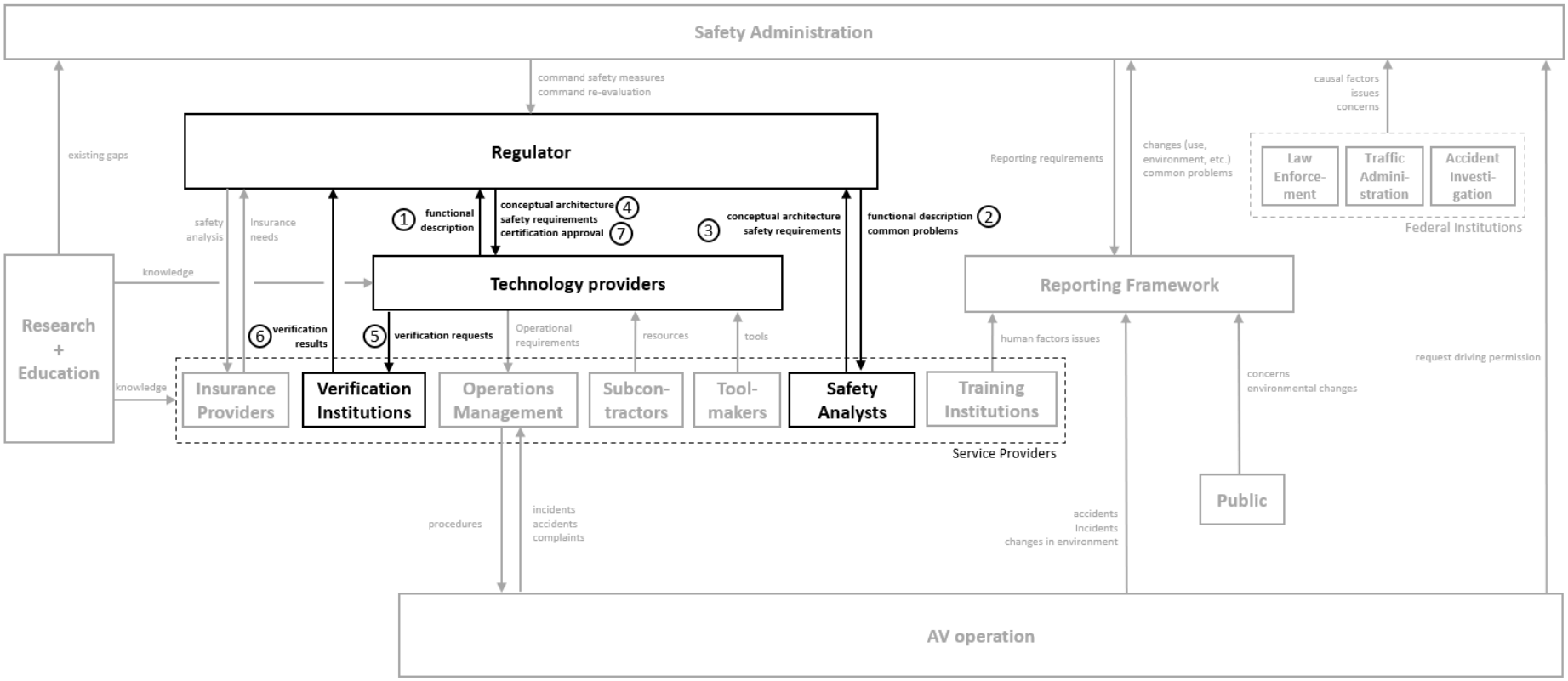


Figure 4-3: Steps for the certification of design changes.

In the first step the technology provider that intends to introduce a change provides a functional description of the planned changes to the regulator (see step ① in Figure 4-3). The regulator then adds the conceptual architecture and safety requirements that were used for the original development of the system and provides this information to a safety analyst (step ②). Based on the provided information the safety analyst performs a safety analysis and identifies safety requirements. Most of the safety requirements will remain unchanged but a need for some new requirements or removing some of the existing requirements may be identified.

The identified safety requirements are then shared with the regulator (see step ③) which informs the technology provider about the updated safety requirements (step ④). The technology provider uses the safety requirements to develop a design solution and provides the finalized design to a verification institution with a request for re-verification (step ⑤). The verification institution conducts the verification of the change and informs the regulator of the results (step ⑥). The regulator uses the verification results to decide on approval and either grants certification or demands that the design is changed and re-verified against the safety requirements (step ⑦).

Finally, a task of certification is to ensure that changes in the operating environment of AVs do not negatively impact safety ([105, p. 464], [95]). As more progress is made in the AV field AVs may spread quickly to environments for which they were originally not designed for and safety has not been considered. Further, environments may change over time and challenge the assumptions that were made in the original design. The organizational structure from Figure 4-1 may be used to control these changes in the operating environment of AVs through certification and ensure safety throughout the life-cycle of AVs. The certification step to regulate changes in the operational environment of AVs are shown in Figure 4-4.

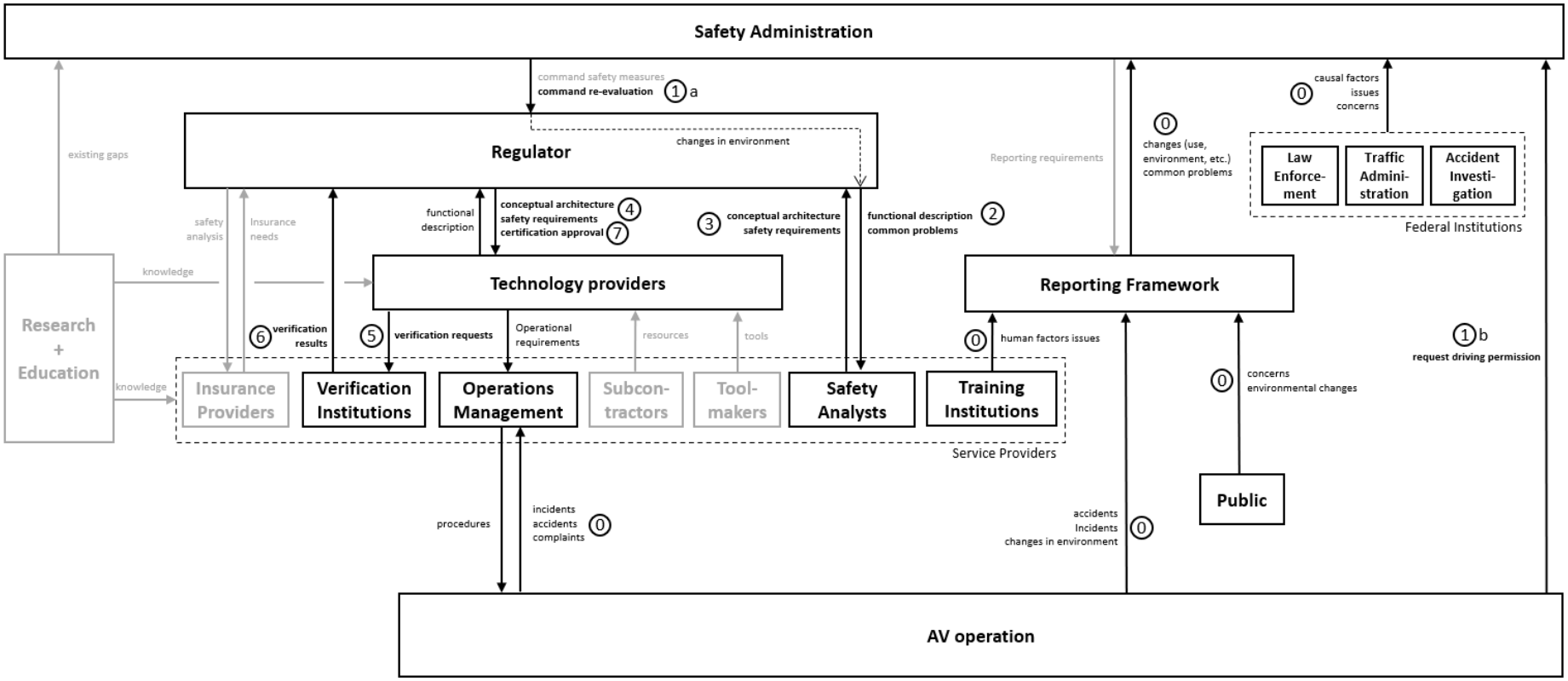


Figure 4-4: Certification steps for the regulation of changes in the operating environment of AVs.

Control structure elements such as Law Enforcement, Traffic Administration, training institutions, AV operators, the public, etc., continuously provide feedback about changes in the operating environment of AVs to the Safety Administration[5] (see step ⓪ in Figure 4-4). The Safety Administration evaluates the reported changes and commands a re-evaluation whenever a change has been identified (see step ①a). The regulator provides the identified environmental changes to a safety analyst along with a description of AV functionality (see step ②). The safety analyst performs a safety analysis, determines if there are new safety requirements that have to be met, and shares the results of the analysis with the regulator (see step ③). If new safety requirements were identified the regulator provides the new requirements to OEMs which can then consider design changes together with their suppliers (see step ④).

Once design changes are implemented the OEMs provide their design to a verification institution and request a re-verification (see step ⑤). The verification institution performs the verification and provides the regulator with the verification results (see step ⑥). Then, based on the verification results, the regulator grants re-certification for operation in the changed environment or requests that further design changes be made to satisfy the safety requirements (see step ⑦).

While this process addresses environmental changes over time it can also be applied when vehicles are used in new types of environments. For example, if an owner imports an AV to a new country for which the AV was originally not intended AV owners or AV operator companies may start this process by simply requesting driving permission with the respective safety administration (see step ①b).

The derived certification processes are mainly intended to solve certification problems specific to AVs. However, the processes may lend themselves to addressing a number of other currently existing problems in automotive, too. For example, as countries

[5]The federal institutions provide direct feedback while training institutions, the public, and AV operators provide their feedback through a reporting framework.

put more effort into reducing the environmental impact of the automotive industry, a systematic process is needed for managing driving permissions for vehicles of different age and technology. To date, rather radical approaches are taken with limited success (see regulation in Mexico City [122] or the development with diesel vehicles in Germany ([123], [124], [125], [126])). A more continuous approach such as the one described above may help to find more effective solutions.

Chapter 5

Summary

This thesis has suggested a model-based approach to the certification of automated vehicles. The motivation for a such a new approach was given in Chapter 1. Chapter 2 presented a literature review that highlighted the currently existing safety problem with AVs and identified gaps in the approaches that are currently taken to address safety. In Chapter 3 a systematic process to derive a generic conceptual architecture model for AVs was demonstrated and safety requirements were identified for the conceptual architecture. Chapter 4 described how the suggested model-based approach to safety may be implemented through certification and how safety-related problems with AVs such as regulation and insurance may be addressed using the derived conceptual architecture model and safety requirements. An organizational structure for model-based certification was derived and processes to involve certification and address safety throughout the AV life-cycle were identified. In this chapter the research contribution of this thesis is summarized (see Section 5.1) and an outlook on potential future work is provided (see Section 5.2).

5.1 Research Contribution

This thesis has identified some of the problems that are currently holding AVs back in generating the expected societal benefit. Most of the identified problems center

around the topic of safety and therefore the following two research questions were identified for this thesis (see Chapter 1):

1. How can a generic model for AVs be systematically derived?

2. How can the derived model be used to solve safety-related problems such as the regulation and insurance of AVs?

The contribution of this thesis can be summarized by the answers it has provided to these research questions.

Research question 1 was answered by demonstrating a systematic way to derive a conceptual architecture and safety requirements. Section 3.1 pointed to the limitations of current approaches to safety and argued that a model-based approach to safety can address these limitations and may help with solving the safety-related problems with AVs. Section 3.2 demonstrated a methodology to systematically derive a conceptual architecture model from a simple control-abstraction of AVs. Then, in Section 3.3 the derived conceptual architecture model was used to perform a safety analysis and identify safety requirements for the conceptual architecture model.

Research question 2 was answered by showing how certification can help implement the suggested model-based approach to safety and how the conceptual architecture model and safety requirements can be used for safety-related problems with AVs such as regulation and insurance. To date, there is no dedicated certification for AVs in most countries. Section 4.2 derived an organizational structure and processes to implement model-based certification that covers the entire life-cycle of AVs. With the proposed model-based certification regulators as well as manufacturers may be relieved as the efforts for safety analysis, the identification of safety requirements, and certification is distributed across regulators, manufacturers, and third parties for safety analysis, verification, etc.

5.2 Future Work

The goal of this thesis was to provide a solution to the safety problem with AVs that is currently a blocking point for their progress and large-scale deployment. It was shown how a generic AV model may systematically derived and how the derived model and safety requirements can be implemented through certification to address the currently existing safety problem. Future work may build on this work to solve problems beyond certification and facilitate progress in the AV industry. Possible extensions of this thesis and topics for future work are outlined in the following.

One future research topic is the creation of a process to develop the conceptual architecture into a detailed architecture design. The model-based certification suggested in this thesis uses the conceptual architecture model and safety requirements as a design input to AV manufacturers. In order for manufacturers to use the conceptual architecture and safety requirements as a starting point for design a process is needed to develop these inputs into a detailed, physical architecture. Such a process may allow manufacturers to develop their individual, likely confidential AV designs while complying with the conceptual architecture that provides guidance with safety and certification. In this way, sensitive design information can be protected while the risk for manufacturers to violate safety standards and fail certification is minimized.

Another topic for future research is the identification of a verification strategy to ensure that manufacturers‘ designs comply with the conceptual architecture and satisfy the safety requirements. If verification is delegated to verification institutions (as suggested in this thesis) verifiers must be able to check if the designs they verify can actually be described with the conceptual architecture that the safety requirements are based on or otherwise safety gaps will remain. A verification strategy may devise verification methods such as theoretical analysis, visual inspection, testing, simulation, etc. to determine compliance with the conceptual architecture and safety requirements.

Finally, future research may be conducted on identifying requirements for the organizational structure for model-based certification. The organizational part of the control structure plays an important role in improving AV safety and the progress of the AV field. This thesis has derived an organizational structure to implement model-based certification and address currently existing safety-related problems with AVs. However, this thesis stopped at the creation of the organizational structure and more requirements could be found if a safety analysis is performed on the organizational structure. This may help identify more constraints for the controllers in the organizational control structure and help prevent unsafe interactions.

Appendix A

Unsafe Control Actions for the Generation of a Conceptual Architecture for AVs

This section provides some examples of the UCAs that were identified for the generation of a conceptual architecture model for AVs in the scope of this thesis.

Table A.1: Unsafe Control Actions – Safety Driver.

Control Action	**Not Providing**	**Providing**	**Provided Too Early / Too Late**	**Provided For Too Long / Stopped Too Soon**
brake	The Safety Driver does not provide braking when the vehicle‘s path is not clear. (UCA-27) [H-1,4]	The Safety Driver provides manual braking with an excessive amount of braking when there is following traffic that cannot decelerate in time. (UCA-28) [H-1]	The Safety Driver provides braking too late after a collision can no longer be avoided. (UCA-29) [H-1]	The Safety Driver provides manual braking for too long when collision has already been averted (potentially causing another collision). (UCA-30) [H-1]
brake	The Safety Driver does not provide braking when the vehicle speed exceeds speed limits. (UCA-31) [H-2,3,4]	The Safety Driver provides manual braking with insufficient amount for an override when an emergency vehicle is present. (UCA-32) [H-4]	The Safety Driver provides braking too early before vehicle is in safe area to decelerate. (UCA-33) [H-1,4]	The Safety Driver provides braking for too long when over-speed has already been corrected (potentially causing another collision). (UCA-34) [H-1]
steering	The Safety Driver does not provide steering when automated steering is inconsistent with vehicle speed. (UCA-35) [H-1,2,3,4]	The Safety Driver provides steering that is excessive beyond the vehicle‘s limits for controllability and stability. (UCA-36) [H-1,2,3,4]	The Safety Driver provides steering target too late after vehicle has exited automated mode. (UCA-37) [H-1,2,4]	The Safety Driver stops providing steering too early before a collision is averted. (UCA-38) [H-1]

Continued on next page

Table A.1: Unsafe Control Actions – Safety Driver (continued).

Control Action	**Not Providing**	**Providing**	**Provided Too Early / Too Late**	**Provided For Too Long / Stopped Too Soon**
activate automated driving	The Safety Driver does not activate automated driving when the driver is about to give up control over the vehicle. (UCA-39) [H-1,2,4]	The Safety Driver activates the automated driving when the vehicle controls have not been calibrated. (UCA-40) [H-1,2,3,4]	The Safety Driver activates automated driving too early before the driver has given up control over the vehicle (e.g. causes override). (UCA-41) [H-1,2,4]	
activate automated driving	The Safety Driver does not activate automated driving when the vehicle enters an area that is automated driving only. (UCA-42)][H-1,4]	The Safety Driver activates the automation when the driver is not familiar with the current system configuration. (UCA-43) [H-1,2,3,4]	The Safety Driver activates automated driving too late after giving up control over the vehicle. (UCA-44) [H-1,2,4]	
activate automated driving		The Safety Driver activates the automation when the automated system is not fully functional. (UCA-45)] [H-1,2,3,4]	The Safety Driver activates automated driving too early before the vehicle has entered an area that is eligible for automated driving. (UCA-46)] [H-1,2,4]	

Continued on next page

Table A.1: Unsafe Control Actions – Safety Driver (continued).

Control Action	**Not Providing**	**Providing**	**Provided Too Early / Too Late**	**Provided For Too Long / Stopped Too Soon**
deactivate automated driving	The Safety Driver does not deactivate the automation when automated driving behavior is inconsistent (e.g. steering leads vehicle off its path, etc.). (UCA-47) [H-1,2,3,4]	The Safety Driver deactivates the automation when there are no means to control the vehicle. (UCA-48) [H-1,4]	The Safety Driver deactivates automated driving too late after the automation has maneuvered the vehicle into an unsafe situation (e.g. an imminent collision). (UCA-49) [H-1,2,3,4]	
disable automation module	The Safety Driver does not disable a module when the function of the module is impaired. (UCA-50) [H-1,2,3,4]	The Safety Driver disables any required modules when automated driving is active. (UCA-51) [H-1,2,3,4]	The Safety Driver disables a module too late after problems with a module have started to impact automated driving. (UCA-52) [H-1,2,3,4]	

Continued on next page

Table A.1: Unsafe Control Actions – Safety Driver (continued).

Control Action	**Not Providing**	**Providing**	**Provided Too Early / Too Late**	**Provided For Too Long / Stopped Too Soon**
throttle	The Safety Driver does not provide throttle when vehicle speed is too slow for the current location. (UCA-53) [H-1,4]	The Safety Driver provides throttle for too long beyond speed limits (e.g. limit for controllability, speed limit, etc.). (UCA-54) [H-2,3,4]	The Safety Driver provides throttle too early when the vehicle‘s path is not clear yet. (UCA-55) [H-1,4]	The Safety Driver stops providing throttle too early before automated driving is re-activated. (UCA-56) [H-1,4]

Table A.2: Unsafe Control Actions – Automation.

Control Action	**Not Providing**	**Providing**	**Provided Too Early / Too Late**	**Provided For Too Long / Stopped Too Soon**
steering	The automation does not provide steering when the vehicle is about to enter a curve. (UCA-57) [H-1,2,4]	The automation provides steering to keep the vehicle in its lane when the driver is trying to pass a bicyclist. (UCA-58) [H-1,4,3]	The automation provides a steering command too early when an upcoming maneuver does not yet require steering input. (UCA-59) [H-1,2,4]	The automation stops providing a steering rate command too soon when a sufficient steering angle (for the planned maneuver, trajectory, etc.) has not been reached yet. (UCA-60) [H-1,2,4]
steering	The automation does not provide a steering command when the vehicle is at an intersection where the traffic is managed by a police officer. (UCA-61) [H-1,4]	The automation provides steering that leads vehicle away from its destination. (UCA-62) [H-2]	The automation provides a steering rate command too late when an upcoming maneuver has already started. (UCA-63) [H-1,2,4]	The automation provides a steering rate command for too long when the vehicle has already completed a maneuver. (UCA-64) [H-1,2,3,4]
steering	The automation does not provide steering when an obstacle enters the vehicle‘s path. (UCA-65) [H-1]	The automation provides steering that exceeds the vehicle‘s limits for controllability and stability. (UCA-66) [H-1,2,3,4]	The automation provides steering too late after an obstacle has entered the vehicle‘s path. (UCA-67) [H-1]	The automation stops providing steering too early before a collision with an obstacle in the vehicle‘s path has been averted. (UCA-68) [H-1]

Continued on next page

Table A.2: Unsafe Control Actions – Automation (continued).

Control Action	**Not Providing**	**Providing**	**Provided Too Early / Too Late**	**Provided For Too Long / Stopped Too Soon**
braking	The automation does not provide braking to bring the vehicle to a safe stop when the vehicle has suffered a collision. (UCA-69) [H-1,2,3,4]	The automation provides a brake command when the driver is providing throttle. (UCA-70) [H-1,3]	The automation provides braking that is excessive and does not allow the following vehicle(s) to avert a collision. (UCA-71) [H-1]	The automation stops providing a brake command too early when the vehicle speed is still above speed limits. (UCA-72) [H-2,3,4]
throttle	The automation does not provide throttle when the vehicle is merging into a faster lane (e.g. to pass a vehicle in front). (UCA-73) [H-1,4]	The automation provides throttle when the vehicle path is not clear. (UCA-74) [H-1,4]	The automation provides throttle too late when the vehicle‘s path is no longer clear (e.g. traffic light switched, junction blocked, etc.). (UCA-75) [H-1,4]	The automation stops providing throttle too soon before the vehicle has cleared an intersection. (UCA-76) [H-1,4]
throttle	The automation does not provide throttle when the vehicle comes to a stop at a railroad crossing. (UCA-77) [H-1,4]	The automation provides throttle when the vehicle is unstable. (UCA-78) [H-1,2,3,4]	The automation provides excessive throttle when other vehicles or obstacles are close. (UCA-79) [H-1,4]	The automation provides throttle for too long when vehicle has already reached target speed in its planned trajectory. (UCA-80) [H-1,3,4]

Continued on next page

Table A.2: Unsafe Control Actions – Automation (continued).

Control Action	**Not Providing**	**Providing**	**Provided Too Early / Too Late**	**Provided For Too Long / Stopped Too Soon**
activate headlights	The automation does not provide headlights when the light conditions are low. (UCA-81) [H-1,2,4]		The automation provides headlights too late after switching them off for an oncoming vehicle. (UCA-82) [H-1,4]	The automation stops providing headlights too early before light conditions are sufficient. (UCA-83) [H-1,4]

Appendix B

Causal Scenarios & Design Constraints for the Generation of a Conceptual Architecture for AVs

This section provides some examples of the design constraints that were identified for the generation of a conceptual architecture model for AVs in the scope of this thesis.

UCA-27: Safety Driver does not provide manual braking when the vehicle's path is not clear (e.g. red light, stop sign, etc.). [H-1,2,3,4]

Scenario-16: The Safety Driver does not know that the automation is not active and assumes that the automation will provide braking. This flawed process model may result from one or more of the following:

a) The Safety Driver does not receive feedback on whether the automation is active.

b) The automation indicates that it is not active but the Safety Driver is unable to interpret the feedback correctly, e.g. because the Safety Driver cannot see

the feedback (feedback light in direct sunlight, etc.).

DC-30: The Safety Driver must receive continuous feedback on the tasks that are currently carried out by automation (e.g. steering, throttle, fully automated driving, etc.).
DC-31: The feedback on whether the automation is active must be robust against interferences in the cockpit such as sun light, different viewing angles, etc.

Scenario-17: The Safety Driver does not know that the automation does not recognize obstacles in its path. This flawed process model may result from one or more of the following:

a) The vehicle appears to slow down and the Safety Driver assumes that the vehicle has detected the obstacles in its path.

b) The Safety Driver receives feedback that the vehicle has detected obstacles but the feedback refers to other obstacles than the ones in the vehicle‘s path.

c) The Safety Driver does not receive information on which obstacles have been detected by automation.

DC-32: The vehicle must continuously provide the operator with feedback about its detected obstacles.
DC-33: Safety Drivers must receive training in identifying obstacles that were missed by the automation.
DC-34: Procedures must be implemented to help Safety Drivers to verify continuously if the automation has missed any obstacles.

Scenario-18: The Safety Driver does not know that the vehicle is about to set off and hence, does not pay attention to whether the vehicle‘s path is clear. This flawed process model may result from one or more of the following:

a) The automation does not inform the Safety Driver before it starts driving in automated mode.

b) The automation informs the Safety Driver that it is about to start driving in automated mode but the Safety Driver is unaware of the intended path (e.g. Safety Driver may assume the vehicle will just re-position itself such as on a parking space).

DC-35: The automation must request permission from the Safety Driver before it starts to drive in automated mode.
DC-36: The automation must show its intended path as part of requests to start automated driving.

Scenario-19: The Safety Driver is not aware of the limitations of driving automation and assumes that closely monitoring traffic is not necessary. This flawed process model may result from one or more of the following:

a) The Safety Driver has not received training about the limitations of automated driving systems.

b) The training on handling the limitations of automated driving systems has led the Safety Drivers to over-reliance, e.g. because the situations encountered in the simulator training made the flaws of automation seem superficial and predictable.

DC-37: Safety Drivers must receive training in understanding and handling the limitations of automated driving systems.
DC-38: The training for Safety Drivers must incorporate accident scenarios from the real world and must not rely exclusively on theory and simulator training.

Scenario-20: The Safety Driver is overwhelmed due to the assigned responsibilities during AV operation. This may occur due to the following:

a) The AV operator company has reduced the number of drivers in a vehicle and hence, increased the workload per Safety Driver.

b) AV operator companies have ordered Safety Drivers to document observed flaws of the automation during operation (e.g. for system improvements).

DC-39: The vehicle must provide support for Safety Drivers to document flaws of the automated driving system without demanding much of their attention.

DC-40: Human factors experts and Safety Drivers must be involved in the allocation of operational tasks to Safety Drivers.

Appendix C

Design Constraints for the Generation of an Organizational Structure for Model-Based Certification

This section provides some examples of the UCAs and design constraints that were identified in the scope of the generation of an organizational structure for model-based certification in this thesis.

The vehicle manufacturer identifies safety gaps only insufficiently. (UCA-84) [H-1,2,3,4]

Scenario-21: The vehicle manufacturer is under economic pressure and cannot afford slowing down development due to gaps found in the safety analysis.

DC-26: The results of the safety analysis must not impact the analyst‘s economic benefit. [Scenario-21]

The vehicle manufacturer provides an AV design that addresses common causal factors in AV accidents only insufficiently. (UCA-85) [H-1,2,3,4]

Scenario-22: The vehicle manufacturer did not know the causal factors that have contributed to previous accidents with AVs.

DC-27: AV manufacturers must receive information about causal factors in previous accidents with AVs. [Scenario-22]

The operator company provided operational procedures that were ineffective for the safe operation of AVs. (UCA-86) [H-1,2,3,4]

Scenario-23: The operator company did not know about problems in operation.
Scenario-24: The operator company knew about problems with the current operational procedures but did not consider changes necessary.

DC-28: Problems in the operation of AVs must lead to a re-evaluation of the operational concept and procedures. [Scenario-23,24]

Scenario-25: The operator company received feedback about accidents, incidents, and issues in operation but assumed the data was collected from fleets of other operator companies and does not apply to their fleet.

DC-29: Operator companies must receive direct feedback about their companies‘ accidents, incidents, and issues in operation. [Scenario-25]

Bibliography

[1] National Transportation Safety Board. Collision Between a Car Operating With Automated Vehicle Control Systems and a Tractor-Semitrailer Truck Near Williston, Florida, May 7, 2016, 2017.

[2] National Transportation Safety Board. Low-Speed Collision Between Truck-Tractor and Autonomous Shuttle, Las Vegas, Nevada, November 8, 2017, 2017.

[3] National Transportation Safety Board. Vehicle Automation Report Tempe, AZ, HWY18MH010. Technical report, National Transportation Safety Board, 2019.

[4] National Transportation Safety Board. Rear-End Collision Between a Car Operating with Advanced Driver Assistance Systems and a Stationary Fire Truck, Culver City, California, January 22, 2018, 2019.

[5] National Transportation Safety Board. Preliminary Report Highway HWY18FH011.

[6] OnlineWebFonts. Car Top View Free Icon. URL `https://www.onlinewebfonts.com/icon/8780`. Accessed: March 3, 2020.

[7] Government of Western Australia. Collision between an autonomous haul truck and manned water cart. Incident Report 226, Government of Western Australia, Department of Mines and Petroleum, Resources Safety, 2015.

[8] Akshay Kumar Shastry. *Functional Safety Assessment in Autonomous Vehicles*. PhD thesis, Virginia Tech, 2018.

[9] Torben Stolte, Gerrit Bagschik, and Markus Maurer. Safety goals and functional safety requirements for actuation systems of automated vehicles. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 2191–2198. IEEE, 2016.

[10] Vasant Raval and Michael Dentlinger. Risk Landscape of Autonomous Cars. *EDPACS*, 56(3):1–18, 2017.

[11] Jason Borenstein, Joseph Herkert, and Keith Miller. Self-Driving Cars: Ethical Responsibilities of Design Engineers. *IEEE Technology and Society Magazine*, 36(2):67–75, 2017.

[12] Asim Abdulkhaleq, Markus Baumeister, Hagen Böhmert, and Stefan Wagner. Missing no Interaction - Using STPA for Identifying Hazardous Interactions of Automated Driving Systems. *International Journal of Safety*, 2(01):115–124, 2018.

[13] Yrvann Emzivat. *Safety System Architecture for the Design of Dependable and Adaptable Autonomous Vehicles*. PhD thesis, Loire Bretagne University, 2018.

[14] Philip Koopman and Michael Wagner. Autonomous Vehicle Safety: An Interdisciplinary Challenge. *IEEE Intelligent Transportation Systems Magazine*, 9 (1):90–96, 2017.

[15] Nidhi Kalra and Susan Paddock. Driving to Safety: How Many Miles of Driving Would It Take to Demonstrate Autonomous Vehicle Reliability? *Transportation Research Part A: Policy and Practice*, 94:182–193, 2016.

[16] Milos Vasic and Aude Billard. Safety issues in human-robot interactions. In *2013 IEEE International Conference on Robotics and Automation*, pages 197–204. IEEE, 2013.

[17] Vinayak Dixit, Sai Chand, and Divya Nair. Autonomous Vehicles: Disengagements, Accidents and Reaction Times. *PLoS one*, 11(12), 2016.

[18] Neal Boudette. Despite High Hopes, Self-Driving Cars Are "Way in the Future". *The New York Times*, 2019. URL `https://www.nytimes.com/2019/07/17/business/self-driving-autonomous-cars.html?action=click&module=inline&pgtype=Homepage`. Accessed: February 23, 2020.

[19] Syed Asim Shah. *Safe-AV: A Fault Tolerant Safety Architecture for Autonomous Vehicles*. PhD thesis, 2019.

[20] Lex Fridman. Human-Centered Autonomous Vehicle Systems: Principles of Effective Shared Autonomy. *Computing Research Repository*, 2018.

[21] Chris Schwarz, Geb Thomas, Kory Nelson, Michael McCrary, Nick Sclarmann, Matthew Powell, et al. Towards Autonomous Vehicles. Technical report, Mid-America Transportation Center, 2013.

[22] Nancy Leveson. System Safety in Computer-Controlled Automotive Systems. In *SAE 2000 World Congress*, 2000.

[23] Roberto Saracco. Guess what requires 150 million lines of code... *IEEE*, 2016. URL `https://cmte.ieee.org/futuredirections/2016/01/13/guess-what-requires-150-million-lines-of-code/`. Accessed: October 13, 2020.

[24] Asim Abdulkhaleq and Stefan Wagner. Experiences with Applying STPA to Software-Intensive Systems in the Automotive Domain. In *2013 STAMP Conference*, 2013.

[25] Asim Abdulkhaleq, Sebastian Vöst, Stefan Wagner, and John Thomas. An Industrial Case Study on the Evaluation of a Safety Engineering Approach for Software-Intensive Systemsin the Automotive Domain. 2016.

[26] Margaret Stringfellow, Nancy Leveson, and Brandon Owens. Safety-Driven Design for Software-Intensive Aerospace and Automotive Systems. *Proceedings of the IEEE*, 98(4):515–525, 2010.

[27] John McDermid, Yan Jia, and Ibrahim Habli. Towards a Framework for Safety Assurance of Autonomous Systems. 2019.

[28] Robert Alexander, Martin Hall-May, and Tim Kelly. Certification of autonomous systems under UK military safety standards. In *Proceedings of the 2nd Systems Engineering for Autonomous Systems (SEAS) Defence Technology Centre (DTC) Annual Technical Conference*. Citeseer, 2007.

[29] Yunhan Jack Jia, Ding Zhao, Qi Alfred Chen, and Z. Morley Mao. Towards Secure and Safe Appified Automated Vehicles. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 705–711. IEEE, 2017.

[30] Carl Bergenhem, Rolf Johansson, Andreas Söderberg, Jonas Nilsson, Jörgen Tryggvesson, Martin Törngren, and Stig Ursing. How to Reach Complete Safety Requirement Refinement for Autonomous Vehicles. 2015.

[31] Patrik Feth, Rasmus Adler, Takeshi Fukuda, Tasuku Ishigooka, Satoshi Otsuka, Daniel Schneider, Denis Uecker, and Kentaro Yoshimura. Multi-aspect Safety Engineering for Highly Automated Driving. In *International Conference on Computer Safety, Reliability, and Security*, pages 59–72. Springer, 2018.

[32] Tyler Gaskill. Pumping the Brakes. *Quality Progress*, 49(11):12, 2016.

[33] Giedre Sabaliauskaite, Lin Shen Liew, and Jin Cui. Integrating Autonomous Vehicle Safety and Security Analysis Using STPA Method and the Six-Step Model. *International Journal on Advances in Security*, 11(1&2):160–169, 2018.

[34] Nancy Leveson. White Paper on the Use of Safety Cases in Certification and Regulation. White paper, Massachusetts Institute of Technology, 2019.

[35] Yvonne Colbert. Camera directed car toward oncoming traffic after windshield replacement, man says. *CBC*, 2018. URL `https://www.cbc.ca/news/canada/nova-scotia/camera-directed-car-toward-oncoming-traffic-after-windshield-replacement-man-says-1.4715817`. Accessed: March 18, 2020.

[36] Michael Laris. Pedestrian in self-driving Uber crash probably would have lived if braking feature hadn't been shut off, NTSB documents show. *The Washington Post*, 2019. URL `https://www.washingtonpost.com/local/trafficandcommuting/pedestrian-in-self-driving-uber-`

collision-probably-would-have-lived-if-braking-feature-hadnt-been-shut-off-ntsb-finds/2019/11/05/7ec83b9c-ffeb-11e9-9518-1e76abc088b6_story.html. Accessed: November 12, 2019.

[37] National Transportation Safety Board. Preliminary Report Highway: HWY18MH010, 2018.

[38] YouTube. Uber self-driving car crash in Tempe, Arizona, 2018. URL https://www.youtube.com/watch?v=XtTB8hTgHbM. Accessed: March 4, 2020.

[39] National Transportation Safety Board. Preliminary Report Highway HWY19FH008, 2019.

[40] YouTube. Tesla autopilot crash in China, 2016. URL https://www.youtube.com/watch?v=fc0yYJ8-Dyo. Accessed: March 4, 2020.

[41] Adrian Padeanu. Google explains how self-driving Lexus RX 450h hit a bus. *motor1.com*, 2019. URL https://www.motor1.com/news/61614/google-explains-how-self-driving-lexus-rx-450h-hit-a-bus-video/. Accessed: December 5, 2019.

[42] Aaron Tilley. Google's Self-Driving Car Caused Its First Accident. *Forbes*, 2016. URL https://www.forbes.com/sites/aarontilley/2016/02/29/googles-self-driving-car-caused-its-first-accident/#43513c4d538d. Accessed: November 12, 2019.

[43] Phil LeBeau. Google's self-driving car caused an accident, so what now? *CNBC*, 2016. URL https://www.cnbc.com/2016/02/29/googles-self-driving-car-caused-an-accident-so-what-now.html. Accessed: October 10, 2019.

[44] University of Toronto. Introduction to Self-Driving Cars, 2019. URL https://de.coursera.org/learn/intro-self-driving-cars. Accessed: March 4, 2020.

[45] Jeffrey Cook. GM sued by motorcyclist injured in crash involving self-driving car. *ABC News*, 2018. URL https://abcnews.go.com/US/gm-sued-motorcyclist-injured-crash-involving-driving-car/story?id=52630445. Accessed: August 29, 2019.

[46] Alison Griswold. Uber has taken its self-driving cars off the road after one flipped over in Arizona. *Quartz*, 2017. URL https://qz.com/942199/uber-suspended-its-self-driving-cars-after-one-flipped-over-in-tempe-arizona/. Accessed: December 11, 2019.

[47] Andrew Maynard and Elizabeth Garbee. Column: Fender-benders tell us more about self-driving cars than the trolley problem ever could. *Tampa Bay Times*, 2017. URL https://www.tampabay.com/opinion/columns/column-fender-benders-tell-us-more-about-self-driving-cars-than-the/2320278/. Accessed: February 8, 2020.

[48] Ryan Randazzo. Self-driving Uber involved in Tempe crash. *AZ Central*, 2017. URL https://www.azcentral.com/story/money/business/tech/2017/09/08/self-driving-uber-involved-tempe-crash/647843001/. Accessed: August 12, 2020.

[49] Andrew J. Hawkins. Tesla's Smart Summon parking feature has been used over a million times. *The Verge*, 2019. URL https://www.theverge.com/2019/10/23/20929438/tesla-smart-summon-use-one-million-q3-2019. Accessed: March 10, 2020.

[50] Jeff Plungis. CR's evaluation of Tesla's highly anticipated feature shows there's a long way to go to get to "Full Self-Driving". *Consumer Reports*, 2019. URL https://www.consumerreports.org/automotive-technology/teslas-smart-summon-performance-doesnt-match-marketing-hype/. Accessed: January 25, 2020.

[51] Lewin Day. Tesla's Smart Summon - Gimmick or Greatness? *Hackaday*, 2019. URL https://hackaday.com/2019/11/07/teslas-smart-summon-gimmick-or-greatness/. Accessed: January 12, 2020.

[52] Brad Templeton. Tesla Rolls Over Own Feet With Pointless Smart Summon Release. *Forbes*, 2019. URL https://www.forbes.com/sites/bradtempleton/2019/10/11/tesla-rolls-over-own-feet-with-pointless-smart-summon-release/#214e83352db4. Accessed: November 11, 2019.

[53] Cole Latimer. Autonomous truck and water cart collide on site UPDATE. *Australian Mining*, 2015. URL https://www.australianmining.com.au/news/autonomous-truck-and-water-cart-collide-on-site-update/. Accessed: January 7, 2020.

[54] Department of Motor Vehicles. Report of Traffic Collision Involving an Autonomous Vehicle (OL 316). Technical report, State of California, 2019. URL https://www.dmv.ca.gov/portal/dmv/detail/vr/autonomous/autonomousveh_ol316. Accessed: April 22, 2020.

[55] Neal F. Boudette. Autopilot Cited in Death of Chinese Tesla Driver. *The New York Times*, 2016. URL https://www.nytimes.com/2016/09/15/business/fatal-tesla-crash-in-china-involved-autopilot-government-tv-says.html. Accessed: January 21, 2020.

[56] Ron Hurtibise. Tesla's autopilot was engaged during fatal crash, NTSB says. *South Florida Sun Sentinel*, 2019. URL https://www.sun-sentinel.com/local/palm-beach/fl-bz-ntsb-tesla-report-autopilot-engaged-20190516-dlrrt646lzadvlgk4eoxk5nehi-story.html. Accessed: February 12, 2020.

[57] Lars Schnieder and René S Hosse. *Leitfaden Safety of the Intended Functionality*. Springer, 2018. ISBN 978-3-658-25022-5.

[58] Kareem Habib. ODI Resume on the subject of automated vehicle control systems. Technical report, National Highway Traffic Safety Administration, 2017.

[59] Andrew J. Hawkins. Uber is at fault for fatal self-driving crash, but it's not alone. *The Verge*, 2019. URL https://www.theverge.com/2019/11/19/20972584/uber-fault-self-driving-crash-ntsb-probable-cause. Accessed: December 5, 2020.

[60] National Transportation Safety Board. Addressing Systemic Problems Related to the Timely Repair of Traffic Safety Hardware in California, 2019.

[61] Sean Hollister. Uber won't be charged with fatal self-driving crash, says prosecutor. *The Verge*, 2019. URL https://www.theverge.com/2019/3/5/18252423/uber-wont-be-charged-with-fatal-self-driving-crash-says-prosecutor. Accessed: October 11, 2019.

[62] Alan Ohnsman. Fatal Tesla Crash Exposes Gap In Automaker's Use Of Car Data. *Forbes*, 16, 2018. URL https://www.forbes.com/sites/alanohnsman/2018/04/16/tesla-autopilot-fatal-crash-waze-hazard-alerts/#6a3755245572. Accessed: September 26, 2019.

[63] Satoshi Otsuka, Daniel Schneider, Denis Uecker, and Kentaro Yoshimura. Multi-aspect Safety Engineering for Highly Automated Driving. In *Computer Safety, Reliability, and Security: 37th International Conference, SAFECOMP 2018, Västerås, Sweden, September 19-21, 2018, Proceedings*. Springer, 2018.

[64] Asim Abdulkhaleq, Stefan Wagner, and Nancy Leveson. A comprehensive safety engineering approach for software-intensive systems based on STPA. *Procedia Engineering*, 128:2–11, 2015.

[65] Rami Debouk Kelly Stashko Barbara J. Czerny, Joseph D'Ambrosio. ISO 26262 Functional Safety Draft International Standard for Road Vehicles: Background, Status, and Overview. Technical report, 2018.

[66] National Highway Traffic Safety Administration. Assessment of Safety Standards for Automotive Electronic Control Systems. Technical report, U.S. Department of Transportation, 2016.

[67] Wikipedia. ISO 26262. URL https://en.wikipedia.org/wiki/ISO_26262. Accessed: March 3, 2020.

[68] International Organization for Standardization. ISO/PAS 21448:2019-Road vehicles - Safety of the intended functionality. Standard draft, International Organization for Standardization, 2019.

[69] Underwriters Laboratory and Edge Case Research. UL 4600 Technical Overview. Webinar, 2019. URL https://www.youtube.com/watch?v=pct1vkGEQNI. Accessed: March 4, 2020.

[70] Phil Koopman. An Overview of Draft UL 4600 "Standard for Safety for the Evaluation of Autonomous Products". *Medium*, 2019. URL https://medium.com/@pr_97195/an-overview-of-draft-ul-4600-standard-for-safety-for-the-evaluation-of-autonomous-products-a50083762591. Accessed: April 22, 2020.

[71] Semiconductor Engineering. UL 4600 - Standard For Safety For the Evaluation of Autonomous Products. *Semiconductor Engineering*. URL https://semiengineering.com/knowledge_centers/automotive/ul-4600-standard-for-safety-for-the-evaluation-of-autonomous-products/. Accessed: December 12, 2020.

[72] Junko Yoshida. UL Takes Autonomy Standards Plunge. *EE Times*, 2019. URL https://www.eetimes.com/document.asp?doc_id=1334569#. Accessed: September 21, 2019.

[73] Underwriters Laboratory and Edge Case Research. UL 4600 General Stakeholder Overview, 2019. URL https://www.youtube.com/watch?v=yg1sfBeJTRI. Accessed: March 4, 2020.

[74] SAE International. Introduction to DO-178C. *SAE International*, 2020. URL https://www.sae.org/learn/content/C1410/. Accessed: September 15, 2020.

[75] AUTOSAR. The standardized software framework for intelligent mobility. URL https://www.autosar.org/. Accessed: March 4, 2020.

[76] Vector Informatik GmbH. Automotive Open System Architecture. URL https://www.vector.com/int/en/know-how/technologies/autosar/. Accessed: March 4, 2020.

[77] Wikipedia. MISRA C, 2019. URL https://en.wikipedia.org/wiki/MISRA_C. Accessed: November 5, 2020.

[78] Matthew Wood, Philipp Robbel, and Michael Maass. Safety First For Automated Driving. Technical report, Aptiv Services et al., 2019.

[79] Laura Fraade-Blanar, Marjory Blumenthal, James Anderson, and Nidhi Kalra. Measuring Automated Vehicle Safety. Technical report, Rand Corporation, 2018.

[80] Department of Defense. MIL-STD-882E: System Safety. Standard practice, Department of Defense, 2012.

[81] National Highway Traffic Safety Administration. Regulations. URL https://www.nhtsa.gov/laws-regulations/fmvss. Accessed: March 3, 2020.

[82] National Highway Traffic Safety Administration. Voluntary Safety Self-Assessment. URL `https://www.nhtsa.gov/automated-driving-systems/voluntary-safety-self-assessment`. Accessed: March 2, 2020.

[83] National Highway Traffic Safety Administration. Automated Driving Systems 2.0: A Vision for Safety. Statement, U.S Department of Transportation, 2017. 442 pp.

[84] Andreas Reschka. Safety Concept for Autonomous Vehicles. In *Autonomous Driving*, pages 473–496. Springer, 2016.

[85] Giedre Sabaliauskaite, Lin Shen Liew, and Fengjun Zhou. AVES - Automated Vehicle Safety and Security Analysis Framework. In *ACM Computer Science in Cars Symposium*, 2019.

[86] Tobias Schmid. Safety Analysis for Highly Automated Driving. In *2018 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, pages 154–157. IEEE, 2018.

[87] Sara Abidi Nasri. *Application of the STPA methodology to an automotive system in compliance with ISO 26262*. B.S. thesis, University of Stuttgart, 2018.

[88] Max Stoltz-Sundnes. *STPA-Inspired Safety Analysis of Driver-Vehicle Interaction in Cooperative Driving Automation*. M.S. thesis, KTH Royal Institute of Technology, 2019.

[89] Nancy Leveson. An Improved Design Process for Complex, Control-Based Systems Using STPA and a Conceptual Architecture. White paper, Massachusetts Institute of Technology, 2019.

[90] Nancy Leveson and Nicholas Dulac. Incorporating Safety in Early System Architecture Trade Studies. *Journal of Spacecraft and Rockets*, 46(2):430–437, 2009.

[91] Cody Harrison Fleming. *Safety-driven Early Concept Analysisand Development*. PhD thesis, Massachusetts Institute of Technology, 2015.

[92] Nancy Leveson. Are You Sure your Software Will Not Kill Anyone? *Communications of the ACM*, 63(2), 2020. ISSN 0001-0782.

[93] Lars Schnieder and Rene Hosse. Ein ganzheitliches Sicherheitsverständnis für die Entwicklung sicherheitsrelevanter elektronischer Steuerungssysteme für Kraftfahrzeuge. In *20th Symposium Automatisiertes und Vernetztes Fahren (AAET)*, 2019.

[94] Martin Hillenbrand. *Funktionale Sicherheit nach ISO 26262 in der Konzeptphase der Entwicklung von Elektrik/Elektronik Architekturen von Fahrzeugen*, volume 4. KIT Scientific Publishing, 2012.

[95] Nancy Leveson. *Engineering a Safer World: Systems Thinking Applied to Safety*. MIT Press, 2011.

[96] Nancy Leveson. *Safeware: System Safety and Computers*. Addison-Wesley, 1995.

[97] Markus Hörwick. *Sicherheitskonzept für hochautomatisierte Fahrerassistenzsysteme*. PhD thesis, Technical University of Munich, 2011.

[98] Nicolas Dulac and Nancy Leveson. An Approach to Incorporating Safety in Early Concept Formation and System Architecture Evaluations. In *Proceedings of the First IAASS Conference: Space Safety, a New Beginning*, volume 599, 2005.

[99] Asim Abdulkhaleq, Stefan Wagner, Daniel Lammering, Hagen Boehmert, and Pierre Blueher. Using STPA in Compliance with ISO 26262 for Developing a Safe Architecture for Fully Automated Vehicles. *Automotive - Safety and Security 2017*, 2017.

[100] Nancy Leveson and John Thomas. STPA Handbook. Handbook, Massachusetts Institute of Technology, 2018.

[101] John Thomas and Seth Placke. Analyzing Feature Interaction in Automobiles. In *2014 STAMP Conference*, 2014.

[102] Harkleroad, Vela, Kuchar, and Barnett Merchant-Bennett. Risk-based Modeling to Support NextGen Concept Assessment and Validation. Technical Report ATC-405, Lincoln Laboratory, Massachusetts Institute of Technology, 2013.

[103] Ioannis Georgilas. Explainable AI, Model Reconciliation and System-Level Analysis for Safe-Control of Medical Devices. *Intelligence, Robots and ethics*, page 137, 2019.

[104] John Thomas. STPA: A Systems Approach for Autonomous Vehicles. Presentation, Massachusetts Institute of Technology, 2019.

[105] Jesse Krompier. Safety First: The Case for Mandatory Data Sharing as a Federal Safety Standard for Self-Driving Cars. *U. Ill. JL Tech. & Pol'y*, page 439, 2017.

[106] Bryan Gibson. Analysis of Autonomous Vehicle Policies. Report, University of Kentucky, 2017.

[107] Nikolay Nyashin. The State and Future of Autonomous Vehicle Regulation in the United States. Report, City University of New York, 2018.

[108] ITU News. Practical lessons for regulating autonomous vehicles. 2018. URL `https://news.itu.int/autonomous-vehicles-regulation-lessons/`. Accessed: October 10, 2019.

[109] Ford. Quick, seamless over-the-air software updates help keep vehicle tech current. *Antelope Valley Ford Lincoln*, 2020. URL https://www.avford.com/blogs/3003/quick-seamless-over-the-air-software-updates-help-keep-vehicle-tech-current/. Accessed: November 23, 2019.

[110] International Organization for Standardization. ISO 26262-1:2018 - Road vehicles - Functional safety. Standard, 2018. URL https://www.iso.org/standard/43464.html. Accessed: December 3, 2019.

[111] Todd Litman. Autonomous Vehicle Implementation Predictions. Report, Victoria Transport Policy Institute, 2020.

[112] Lynne McChristian and Richard Corbett. Regulatory Issues Related to Autonomous Vehicles. *Journal of Insurance Regulation*, 35(7), 2016.

[113] Marc Scribner. Self-Driving Regulation. Report, Competitive Enterprise Institute, 2014.

[114] Bill Canis. Issues in Autonomous Vehicle Testing and Deployment. Report, Congressional Research Service, 2020.

[115] John Leonard. The Reality and Future of Automated Driving. Presentation, MIT Club of Northern California, 2019.

[116] Ding Zhao and Huei Peng. From the Lab to the Street: Solving the Challenge of Accelerating Automated Vehicle Testing, 2017.

[117] Clifford Atiyeh. NHTSA Mildly Redefines Safety Rules for Autonomous Cars. *Car and Driver*, 2020. URL https://www.caranddriver.com/news/a31785858/nhtsa-new-safety-rules-autonomous-cars/. Accessed: September 14, 2019.

[118] Cliff Saran. Autonomous vehicle tech is too costly and unreliable for production. *Computer Weekly*, 2019. URL https://www.computerweekly.com/news/252461838/Autonomous-vehicle-tech-is-too-costly-and-unreliable-for-production. Accessed: October 23, 2019.

[119] Eric Schneider. Insurance for autonomous vehicles & self-driving cars. *Foundershield*, 2019. URL https://foundershield.com/insurance-for-autonomous-vehicles-self-driving-cars/. Accessed: November 11, 2019.

[120] Mohsen Khalkhali. Safety of Autonomous Driving Systems. Email message to Michael Schmid, February 6, 2020.

[121] Insurance Information Institute. Background on: Self-driving cars and insurance. 2018. URL https://www.iii.org/article/background-on-self-driving-cars-and-insurance. Accessed: December 11, 2019.

[122] Olivier De Weck, Daniel Roos, and Christopher Magee. *Engineering systems: Meeting human needs in a complex technological world*. MIT Press, 2011.

[123] Deutsche Welle. German court bans diesel vehicles on key autobahn. 2018. URL `https://www.dw.com/en/german-court-bans-diesel-vehicles-on-key-autobahn/a-46317300`. Accessed: December 2, 2019.

[124] Stella Braun. Battling air pollution through driving bans. *Deutsche Welle*, 2018. URL `https://www.dw.com/overlay/media/en/battling-air-pollution-through-driving-bans/42807736/46317300`. Accessed: August 29, 2019.

[125] Arthur Sullivan. Older German diesels face reckoning. *Deutsche Welle*, 2018. URL `https://www.dw.com/en/older-german-diesels-face-reckoning/a-45671281`. Accessed: March 19, 2019.

[126] Peter Herzog, Michael Weissbaeck, Holger Herrmuth, and Martin Schüssler. The Diesel SUV-Pushing Back Emission Frontiers. Technical report, SAE Technical Paper, 2007.