

Computer Communications and Networks

Dietmar P. F. Möller
Roland E. Haas

Guide to Automotive Connectivity and Cybersecurity

Trends, Technologies,
Innovations and Applications



Computer Communications and Networks

Series editors

Jacek Rak, Department of Computer Communications, Faculty of Electronics, Telecommunications and Informatics, Gdansk University of Technology, Gdansk, Poland

A. J. Sammes, Cyber Security Centre, Faculty of Technology, De Montfort University, Leicester, UK

Editorial board members

Burak Kantarci, School of Electrical Engineering & Computer Science, University of Ottawa, Ottawa, Ontario, Canada

Eiji Oki, Graduate School of Informatics, Kyoto University, Kyoto, Japan

Adrian Popescu, Department of Computer Science and Engineering, Blekinge Institute of Technology, Karlskrona, Sweden

Gangxiang Shen, School of Electronic and Information Engineering, Soochow University, Suzhou, China

The **Computer Communications and Networks** series is a range of textbooks, monographs and handbooks. It sets out to provide students, researchers, and non-specialists alike with a sure grounding in current knowledge, together with comprehensible access to the latest developments in computer communications and networking.

Emphasis is placed on clear and explanatory styles that support a tutorial approach, so that even the most complex of topics is presented in a lucid and intelligible manner.

More information about this series at <http://www.springer.com/series/4198>

Dietmar P. F. Möller • Roland E. Haas

Guide to Automotive Connectivity and Cybersecurity

Trends, Technologies, Innovations and
Applications



Springer

Dietmar P. F. Möller
Clausthal University of Technology
Clausthal-Zellerfeld, Niedersachsen
Germany

Roland E. Haas
QSO Technologies
Bangalore, Karnataka
India

ISSN 1617-7975 ISSN 2197-8433 (electronic)
Computer Communications and Networks
ISBN 978-3-319-73511-5 ISBN 978-3-319-73512-2 (eBook)
<https://doi.org/10.1007/978-3-319-73512-2>

Library of Congress Control Number: 2018932982

© Springer International Publishing AG, part of Springer Nature 2019

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Printed on acid-free paper

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Foreword by Thomas Hanschke

The automotive industry, which encompasses a wide range of companies and organizations, is one of the most important worldwide industries today as it becomes more aware and responsive to its surroundings. Automakers are responsible for the design, development, manufacturing, marketing, and selling of automobiles and trucks, also called motor vehicles or, in short, vehicles. These vehicles provide promising intelligent functionality and get smarter at every international motor show (IAA), the world's leading trade show in the increasing complexity of vehicles such as digitization, electromobility, and smart mobility. Therefore, this book outlines research and experience gained about advances in technological innovations with regard to trends, technologies, innovations, and applications in the automotive industry. The technological advances in sensor and navigation technologies, the networked living space through the Internet of Things, and the advances of service in the form of an Internet of Data and Services and cloud-based ones will spur the visionary and feasible mobility of the future, the so-called connected and autonomous driving. The term connected car refers to the next generation of car technologies making use of the Internet, enabling the passengers of the vehicle to take advantage of numerous new services and features. The idea of fully autonomous driving cars seems to be too futuristic for much of the driving public to embrace right now. But for automakers, the path from current models to driverless cars is going to be an exciting period of transformation. These innovative developments represent enormous opportunities even as they augur a perilous, unsteady phase for the automotive industry. In this regard automotive cybersecurity is another great theme for the future of mobility embedding advanced digitization concepts under conditions of adequately adapted innovations. Furthermore, the automotive industry will be facing numerous sweeping and interlinked changes in the next decades. Not only are there many different potential changes facing the automotive industry but, unlike most other industries, the automotive industry, while incorporating modern Internet network-enabled technology, has been forced to completely and fundamentally reinvent itself as other industries have during the last decades. Thus, automotive cybersecurity is quickly becoming the most important factor when purchasing a modern vehicle, due to the increasing part of software and digital components and systems on board and the connective and surrounding digital infrastructure. Against this background, the book describes, in contrast to other books which focus more on

automotive E/E and software technology, the essential methodological and theoretical basics and extends them to the body of knowledge of future car characteristics in connection with the necessary technological trends, technologies, innovations, and applications related to the need of automotive connectivity and the associated cybersecurity.

I strongly recommend Prof. Dr. Dietmar P. F. Moeller's scholarly writing to students, academicians, and industrialists who are keen to learn about advanced methodologies in automotive connectivity and cybersecurity. His scientific expertise, he is a professor for stochastic models in engineering sciences of Clausthal University of Technology (TUC) and a member of the Simulation Science Center (SWZ) Clausthal-Göttingen, stands for this advanced and innovative book topic. The co-author, Dr. Roland Haas, is the founder and CEO of QSO Technologies, located in Bangalore, India, who is experienced in different sectors of the automotive industry, giving the book a detailed insight into its applications. I can say without reservation that this book, and more specifically the method it espouses, will change fundamental ideas for cutting-edge innovation and disruption in the automotive domain.

President Clausthal-University of Technology
Clausthal-Zellerfeld, Germany

Thomas Hanschke

Foreword by Jerry Hudgins

Automobiles have become one of the basic needs of humanity as global populations have become more and more mobile during the past century; however, cars and consumers have changed in different ways. Major automobile brands now implement new technologies based on what will improve the consumer driving experience, a strategy that has proven to be their key to success. Changing market dynamics are energizing the automotive industry, which has always been at the forefront of defining new paradigms for the new technologies embedded in their vehicles. Today, these new technologies can be summarized by two words: smart mobility. This book outlines the latest in smart mobility research, technological innovation, and applications within the automotive industry.

Automotive connectivity and cybersecurity are the overriding themes of smart mobility as advanced digital concepts are adapted for use in today's vehicles. The central focus of this book is the networking of the virtual computer world (cyber) with automotive (physical) components to create cyber-physical systems that incorporate the different intelligent assist systems used in today's vehicles. Cyber-physical systems, in this sense, consist of strong digital platforms that are well structured, well integrated, and only as complex as is absolutely necessary. These systems also guarantee that drivers and passengers are protected by innovative and intelligent safety systems whose connectivity enables them to access different kinds of information sources and services from within and outside of a vehicle. As this strong connectivity continues to evolve, it is imperative that the automotive industry examine the vulnerability of connected cars and determine what cybersecurity methods can best be used to defend against cybercriminal attacks on vehicles.

I strongly recommend Prof. Dr. Dietmar P.F. Möller's scholarly writing to students, academicians, and industry experts who seek to learn more about advanced methodologies in automotive connectivity and cybersecurity. As an adjunct professor in the Electrical and Computer Engineering Department of the University of Nebraska, Dr. Möller's research and expertise in cybersecurity is a valuable addition to our students and faculty. The co-author, Dr. Roland Haas, founder and CEO of QSO Technologies, located in Bangalore, India, provided expertise from different sectors of the automotive industry, giving the book a practical perspective and detailed insight into its real-world applications. This book, in contrast to other books that focus more on the basic theories

and methods associated with automotive electrical/electronic issues and software technology, extends beyond the basics to include future technological trends, innovations, and new applications in the automotive industry.

Head, Department of Electrical and
Computer Engineering, University of Nebraska
Lincoln, NE, USA

Jerry Hudgins

Foreword by Rayford Vaughn and Tommy Morris

Cybersecurity has emerged as one of the most important needs on the research front as evidenced by its nearly daily appearance in the news, loss of millions of files containing personally identifying information, allegations of electronic voting interference, cyberattacks against the electric grid, and many more such incidents. Industry has developed and implemented many new technologies in attempts to improve the cybersecurity protecting their mission critical IT infrastructure, but breaches still occur and the resulting penetrations cause reputational, economic, and physical harm. It appears that the adversary has the advantage and that the “penetrate and patch” philosophy is alive and well. While historically we have been concerned with traditional computer security and database security, advances in automation in other sectors bring with it the threat of cyberattack and new domains of cybersecurity research. Examples include industrial control system security, weapon systems security, securing the “Internet of Things” or IOT, and transportation security. It is these latter concerns that Dr. Dietmar P.F. Möller and Dr. Roland Haas have chosen to address.

Research topics involving automotive security are becoming increasingly important as rapid advances are being made in driverless vehicles, including the introduction of artificial intelligence into the operation of automobiles, over-the-air firmware updates, vehicle-to-vehicle communication, and the collection/storage of private information by the automobile. The case for enhanced proofs of correctness, stronger verification and validation techniques, formal models, and code proofs are stronger today than in the past.

A treatment of these subjects through case studies and narrative is needed, and we compliment Dr. Dietmar P.F. Möller and Dr. Roland Haas for taking up the challenge of writing this book and giving it a practical perspective with detailed insight into real-world applications. We are quite comfortable that this book will find

its way into academic settings as well as industrial R&D organizations and that it will promote the kind of thoughtful dialogue necessary before vulnerable transportation systems are deployed to the public.

Vice President Research, Research and
Economic Development, The University of Alabama
Huntsville, AL, USA

Rayford Vaughn

Associate Professor ECE, Director,
Center for Cybersecurity, Research and Education,
The University of Alabama
Huntsville, AL, USA

Tommy Morris

Preface

The goal of this book is to provide a comprehensive, in-depth, state-of-the-art summary of automotive connectivity and cybersecurity with regard to trends, technologies, innovations, and applications. It describes the challenges of the global automotive market, clearly showing where the multitude of innovative activities fit within the overall effort of cutting-edge automotive innovations and provides an ideal framework for understanding the complexity of automotive connectivity and cybersecurity. Given this complexity, we had to make some choices in selecting the material for this book. A top-down approach was taken that introduces the promising intelligent functionality of vehicles but also increases their complexity. Therefore, this book provides essential background on the issues facing the automotive industry as it attempts to balance consumers' mobility needs with their desire for convenience and safety.

Digitization will enable a quantum leap forward in the realization of sustainable, smart mobility services. This will require accessibility; embedment; tiny, powerful computers; sensors; wireless networks; smart devices; cloud services; etc. to support the vision of smart mobility and make it a reality. This book provides a framework within which the reader can analyze and integrate the associated requirements. Without such a reference, the practitioner is left to ponder the plethora of terms, standards, and practices that have been developed independently and that often lack cohesion, particularly in nomenclature and emphasis. Therefore, this book is intended to both cover a broad range of aspects of automotive connectivity and cybersecurity and provide a synopsis for the consideration of the many issues associated with these topics. The subjects discussed include the automotive market; automotive research and development; automotive E/E and software technology; connected cars and autonomous vehicles; and respective methodological approaches to cybersecurity, such as intrusion detection and prevention, to avoid cyberattacks against vehicles.

First, an overview of the automotive industry is presented that describes the technology wave that has changed the automotive industry and, ultimately, drives it toward futuristic concepts, such as smart mobility and autonomous driving.

Smart mobility characterizes the visionary, affordable mobility of the future, which will be applicable to and usable by everyone regardless of (1) location and

region, (2) periods of use and duration, and (3) individual ability and budget, resulting in a new business model called Mobility-as-a-Service.

In addition, the automotive industry is analyzed with regard to the global production and sale of vehicles. The analysis focuses on some automotive megatrends, such as tighter emission controls and the rise of electric vehicles with their need for an adequate charging infrastructure. Also, information on the connection between cars, the required road infrastructure, and the advanced driver assistance systems for accident avoidance is presented. The issues that automotive original equipment manufacturers (OEMs) and suppliers are facing due to digitization and autonomous driving are summarized, and new players and challenges in the automotive domain are identified.

Second, automotive research and development is outlined, as well as background on the complexity involved in developing new vehicle models. The automotive development process is described in detail, taking into account the three stages: conceptualization, project, and validation. A huge advantage in efficiency is realized through the deployment of specific tools, such as computer-aided design (CAD) systems for geometric design, digital mock-up (DMU), and computer-aided engineering (CAE) for analyzing, designing, and manufacturing products. This results in new product creation processes and better product quality by embedding the paradigm of requirements engineering. Furthermore, automotive modularization and product family-based platforms are introduced as they relate to virtual product creation and product life cycle management.

Against such a background, automotive mechatronic, electric, and electronic systems in cars, as well as automotive software technologies, are discussed, focusing on the different kinds of electronic control units (ECUs) required in today's premium cars.

Sensor technologies, which describe devices that generate a measurable signal in response to a stimulus, e.g., from car ECUs, are introduced, as well as bus systems and architectures. All of these systems have an impact on safety as part of the overall safety of a car. In modern cars, ECUs are functional and spatially distributed, which requires adequate test facilities, such as the hardware-in-the-loop test bed.

Finally, the AUTomotive Open System ARchitecture (AUTOSAR), a worldwide, joint initiative of several major industries to create and establish an open standardized software architecture for automotive ECUs, is described.

However, a textbook cannot describe all of the innovative aspects in the automotive domain. For this reason, the reader is referred to specific supplemental material, such as textbooks, reference guides, user manuals, etc., as well as Internet-based information that addresses several of the topics selected for the book.

Third, the technologies essential for the evolution of connected cars, such as cyber-physical systems that integrate computing and networking technologies and the Internet of Things that offers communication capabilities anytime, anywhere, are discussed. They represent the enabling technologies and driving factors of connected cars.

New business models related to services and applications are being developed. Furthermore, an interoperable, scalable platform is essential for the connected car paradigm and infrastructure. The required network is based on three sets of IEEE standards. Besides AUTOSAR, GENIVI is a development standard for setting requirements and implementation standards and providing certification programs. Beyond connected cars, autonomous vehicles have become the most advanced technological development in smart mobility. The advances in wireless networks of connected cars and autonomous vehicles, however, can have a negative impact due to the emergence of new types of cyberattacks. Therefore, cybersecurity is becoming a key issue with the main objectives of detecting, deterring, and averting vulnerabilities. Thus, intrusion detection and prevention represents the most important concern for overcoming weaknesses in the attack value chain. Thus, the most important methods are introduced, as well as advances in the direction of deep learning. Moreover, the innovative field of mobility apps for connected cars is included.

Fourth, due to the methodological background of connected cars and cybersecurity, their practical implementation is another important subject of the book, showing the reader how to develop and implement new services and technologies, such as carsharing, car hailing and ridesharing, connected parking, and automated valet parking, as well as advanced driver assistance systems.

The material in the book can be difficult to comprehend if the reader is new to such an approach. Automotive connectivity and cybersecurity is a multidisciplinary domain, founded in computer science, systems and software engineering, mechanical engineering, simulation science, and communications engineering as well as electronics. Readers may find the material challenging. Therefore, specific case studies have been included with related topics to help the reader master the material. It is assumed that the reader has some knowledge of basic electric and electronic components and some experience with systems and software engineering.

The book can be used in various ways as the primary text for a course. It contains more material than can be covered in detail in a quarter-long (30-h) or semester-long (45-h) course. Instructors may elect to choose their own topics and add their own case studies. The book can also be used for self-study as a reference for engineers, scientists, and computer scientists for on-the-job training; for students in graduate schools; and for automotive connectivity and cybersecurity practitioners and researchers.

For instructors who have adopted the book for use in a course, a variety of teaching support materials are available for download from <http://www.springer.com/book/978-3-319-73511-5>. These include a comprehensive set of PowerPoint slides to be used for lectures and all video-recorded classes.

The book is divided into 12 chapters, which can be read independently or consecutively.

Chapter 1 gives a brief overview of the specific topics covered in the book. Compared to other industries, the automotive industry has taken advantage of many

efficiency improvements driven by Internet-based technology but has also maintained the same structure, as opposed to reorganizing its whole ecosystem. A number of factors could push the automotive industry into new configurations, perhaps ultimately toward futuristic concepts, such as smart mobility. Smart mobility characterizes the visionary, feasible mobility of the future: applicable and usable for everyone regardless of location and region, regardless of periods of use and duration, and regardless of individual ability and budget.

Chapter 2 gives an overview of the global production and sales of the automotive industry and reports on the industry's megatrends, such as tighter emission controls and the rise of electric vehicles, car ownership versus mobility, connectivity, advanced driving assistance systems, and autonomous driving. It also covers background on the digital transformation in the automotive industry.

Chapter 3 focuses on the automotive development process, specifically the complexity involved in developing new vehicle models and the modularization approach and platforms used in the automotive industry, which will allow the efficient handling of an ever-increasing, multibrand vehicle model line. Moreover, the idea of product life cycle management is introduced, an approach that facilitates collaborative work processes for various phases of the product or system life cycle represented by a number of phases and activities spread out across the automaker's organization and suppliers. The sum of these activities is called the product or system life cycle, which can be described using a model that contains the conceptualization, utilization, evolution, and ultimate disposal phases.

Chapter 4 gives an overview of mechatronic, electric, and electronic systems in the automotive domain, as well as architectures and bus system requirements, with an emphasis on disciplined approaches to their design. The increasing role of software content and product complexity requires more adequate development tools, such as model-based software development and hardware-in-the-loop (HIL) tests. Furthermore, AUTOSAR is introduced, as well as the GENIVI platform, essential for telematic and infotainment components, and future trends. As a practical example, advanced driver assistance systems, which support vehicle drivers by enhancing safety and by improving driving conditions, are discussed, as well as the required sensor suite.

Chapter 5 provides a detailed treatment of the key technologies essential for the evolution of connected cars. Cyber-physical systems are engineered systems that integrate computing and networking technologies and the Internet of Things, which offer communication capabilities anytime, from anywhere, with everything. It also refers to telematic and infotainment concepts and refers to platforms and architectures for connected cars, as well as the connected car in the cloud and autonomous vehicles. Several case studies that are essential for the evolution of the connected car, such as BMW's ConnectedDrive; Mercedes's COMAND® Online; and HERE, which provides digital mapping for fully autonomous driving.

Chapter 6 introduces cybersecurity as a body of technologies, processes, and practices designed to protect computers, data, networks, and programs against

intrusion, damage, or unauthorized access by cyberattacks. It focuses on the scale and complexity of vehicles' cyber and physical components and their vulnerability to a variety of security challenges, intrusions, threats, and malicious cyberattacks, whose intent is to disrupt communication, steal sensitive information or records, and damage the functioning of the system, as well as the risk level as a function of likelihood and consequences. Hence, a solid theoretical foundation for cybersecurity of vehicle cyber-physical systems is introduced based on the concepts of artificial intelligence; deep neural networks (DNN) and deep learning (DL); control theory; epidemic theory; game theory; graph theory; and the importance of cybersecurity with regard to different kinds of attack scenarios, e.g., spear phishing attack. Furthermore, the characteristics of attack taxonomies, as well as automotive attack surfaces and vulnerabilities, are presented along with the anatomy of attack surface intrusion points in vehicles and the associated risks. However, vehicle security depends on a variety of different tools and methods that systematically perform security testing. Intrusion detection, in this regard, describes the detection of any set of actions that attempts to compromise the integrity, confidentiality, or availability of a system, whereas intrusion prevention refers to actions that attempt to prevent a detected intrusion from succeeding. Different detection methods for different kinds of intrusion types are described, including numerous static, dynamic, and hybrid methods for prevention, as well as several examples of car hacking.

Chapter 7 begins by motivating the innovative topic of mobile apps for connected cars and focuses on the current trends in Car IT and agile software development. The two major operating systems, iOS and Android, and the corresponding app markets are briefly discussed, as well as the features of application programming interfaces for mobile app development and how car manufacturers are embracing smartphone technologies by integrating Apple's and Google's hardware and software into cars' infotainment systems. Important programming languages for app development, such as Objective-C®, Swift®, and Java® App, are briefly described, followed by a detailed treatment of the design and implementation of such apps for ridesharing, carpooling, and cab sharing.

Chapter 8 discusses carsharing, analyzing the carsharing concept and the different variants of it, as well as carsharing services offered to date. It also describes significant modifications to the hardware/software infrastructure of a car required for use in the carsharing business model. The impact of electric vehicles in carsharing applications is discussed, as well as their specific system architecture, which is highlighted by a block diagram of a standard electric vehicle. Also, carsharing activities by OEMs and their brands are surveyed. Since the whole use case of carsharing relies on the constant connectivity between the car and the backend system, proper security of the used vehicles is a major concern that can be realized by intrusion detection and prevention to avert vulnerabilities through cyberattacks.

Chapter 9 presents car hailing and ridesharing services as a promising approach for reducing personal car usage in a city, thereby cutting down on the need for parking spaces, reducing traffic jams, and helping to reduce pollution. It covers online transportation network companies offering cab services/car hailing and ride hailing, which provide cab services through their respective apps for smartphones. As a case study, we analyze the mobility-on-demand services in the metropolitan area of Bangalore, India, with regard to cab types and prices, as well as services offered. The problem of safety and initiatives to increase ridesharing safety and prevent crime, both for customers and for drivers, is described in detail.

Chapter 10 deals with one of the most relevant and straightforward applications of connected cars—connected parking, including the main challenges and opportunities for connected parking. A multitude of new apps provide information, often in real-time, about available parking spaces; manage the booking, often allowing for cashless billing; and integrate with OEMs' connectivity services. The most sophisticated version to automate the complete parking process is the automated valet parking (AVP). AVP systems turn the vehicle into a robotic car that automatically finds a parking space and maneuvers the car into a free slot. The first commercial systems will be available in high-end cars soon and will also be deployed for carsharing. However, cybersecurity will have a huge impact on connected parking and AVP. Thus, the major cyber threats are analyzed, and potential solutions for increasing cybersecurity, such as intrusion detection and prevention, are discussed.

Chapter 11 gives an introduction to advanced driver assistance systems (ADAS), presenting examples of commercial ADAS. Main topics are image processing and object tracking algorithms, as well as the detection of moving objects and the respective optical flow algorithm. The implementation of an ADAS using MATLAB® is shown as a use case for rapid prototyping using MATLAB's image processing toolbox. The chapter also presents an introductory treatment of software architectures for higher-level ADAS functions and autonomous driving. The chapter concludes with a discussion of cybersecurity and functional safety and presents a comprehensive list of further reading material.

Chapter 12 summarizes the investigation conducted by the authors of this book and gives an outlook on future trends, technologies, innovations, and applications.

Besides the methodological and technical content, all of the chapters in the book contain chapter-specific, comprehensive questions to help readers determine if they have gained the required knowledge, identify possible knowledge gaps, and conquer those gaps. Moreover, all chapters include references and suggestions for further reading.

We would like to express special thanks to Patricia Worster, University of Nebraska-Lincoln, for her excellent assistance in proofreading. We would also like to thank Wayne Wheeler and Simon Rees, Springer Publ., for their help with the organizational procedures between the publishing house and the authors.

Furthermore, we would like to thank Sainath Suni and Kavitha S.K., QSO Technologies, for supporting the proofreading process and drawing many of the illustrations for this book from sketches we drafted. The authors also would like to thank Prof. Asoke Talukder (IIIT-B emeritus), Prof. Dinesha K.V. (IIIT-B), Prof. K.L.S. Sharma (IIIT-B emeritus), Tobias Kuipers (MBRDI), Shambo Bhattacharjee (University of Leeds), and Lynn Degrande for their valuable inputs and feedback.

Moreover, we sincerely thank our students from TUC and IIIT-B and all of the authors who have published Car IT, car hacking, cybersecurity, or smart mobility material and have directly and/or indirectly contributed to this book through citations. Finally, we would like to deeply thank our wives, Angelika and Kavitha, for their encouragement, patience, and understanding during the writing of this book.

Clausthal-Zellerfeld, Germany
Bangalore, India

Dietmar P. F. Möller
Roland E. Haas

Contents

1	Introduction	1
1.1	The Automotive Industry	3
1.2	Scope of This Book	7
1.3	Overview of Topics	9
	References and Further Reading	11
2	The Automotive Industry	13
2.1	The Automotive Market	13
2.2	The Automotive Megatrends	19
2.2.1	Tighter Emission Controls and the Rise of Electric Vehicles	20
2.2.2	Car Ownership Versus Mobility	23
2.2.3	Connectivity	25
2.2.4	Safety and Advanced Driver Assistance Systems	26
2.2.5	Autonomous Driving	28
2.2.6	Digitalization	29
2.3	Automotive OEMs and Suppliers	30
2.4	New Players and Challenges	33
2.5	The Digital Transformation of the Automotive Industry	34
2.6	Exercises	37
	References and Further Reading	39
3	Automotive Research and Development	45
3.1	The Automotive Development Process	45
3.1.1	Requirements Engineering	59
3.1.2	Design as a Multiparameter Optimization Problem	60
3.2	Automotive Modularization and Platforms	63
3.3	Virtual Product Creation	64
3.4	Product Life Cycle Management	69
3.4.1	Loss of Control in Life Cycle Management	70
3.4.2	Systems Engineering Approach	71
3.4.3	Product Life Cycle Stages	73
3.4.4	Software Life Cycle Processes	75

3.5 Exercises	77
References and Further Reading	79
4 Automotive E/E and Automotive Software Technology	83
4.1 Mechatronic Systems in the Car	83
4.2 Automotive Electronics	86
4.2.1 Body Electronics	89
4.2.2 Chassis Electronics	92
4.2.3 Comfort Electronics	94
4.2.4 Driver Assistance Electronics	94
4.2.5 Electronic Control Units	98
4.2.6 Entertainment/Infotainment Electronics	100
4.2.7 Sensor Technology	102
4.3 E/E Architectures and Topologies	109
4.3.1 Objectives	110
4.3.2 Architectures and Topologies	111
4.3.3 Bus Systems and ISO Standards	114
4.4 Functional Safety	121
4.5 Automotive Software Engineering	126
4.5.1 Increasing Software Content and Product Complexity	127
4.5.2 Model-Based Development	130
4.5.3 Hardware-in-the-Loop Tests	133
4.6 AUTOSAR	142
4.7 AUTOSAR Adaptive Platform	147
4.8 GENIVI	147
4.9 Example: Advanced Driver Assistance System	149
4.9.1 ADAS Functionalities	151
4.9.2 ADAS Sensor Types	155
4.9.3 Pros and Cons of the ADAS Sensor Suite	162
4.10 Trends	163
4.11 Exercises	164
References and Further Reading	167
5 The Connected Car	171
5.1 Cyber-Physical Systems	171
5.1.1 Introduction to Cyber-Physical Systems	172
5.1.2 Cyber-Physical Systems Design Recommendations	180
5.1.3 Cyber-Physical Systems Requirements	184
5.1.4 Cyber-Physical Control Systems	189
5.1.5 Cyber-Physical Vehicle Tracking	200
5.2 Internet of Things	206
5.2.1 Internet of Things Enabling Technologies	208
5.2.2 RFID and WSN Technology	210

5.3	Telematics, Infotainment, and the Evolution of the Connected Car	214
5.3.1	Telematics	215
5.3.2	Infotainment	222
5.3.3	Evolution of the Connected Car	224
5.4	Platforms and Architectures	233
5.4.1	Connected Car Architecture and Challenges	234
5.4.2	Connected Car Reference Platform	237
5.4.3	Connected Car in the Cloud	238
5.5	Autonomous Vehicles	241
5.6	GENIVI Alliance	247
5.7	Case Studies	249
5.7.1	BMW ConnectedDrive Store	249
5.7.2	Mercedes COMAND Online	252
5.7.3	HERE: Digital Maps for Fully Autonomous Driving	254
5.8	Exercises	257
	References and Further Reading	260
6	Automotive Cybersecurity	265
6.1	Introduction to Cybersecurity	266
6.1.1	Cybersecurity and Vulnerability	272
6.1.2	Artificial Intelligence	272
6.1.3	Control Theory	282
6.1.4	Epidemic Theory	284
6.1.5	Game Theory	287
6.1.6	Graph Theory	291
6.1.7	Importance of Cybersecurity	294
6.1.8	Automotive IT and Cybersecurity	302
6.1.9	Attack Value Chain	307
6.1.10	Holistic Cybersecurity Solutions	309
6.2	IT Security in Automotive Cyber-Physical Systems	316
6.2.1	Vehicle Network Technologies and Cybersecurity	322
6.2.2	Cyberattack Taxonomy	326
6.3	Hacking and Automotive Attack Surfaces and Vulnerabilities	329
6.3.1	Hacking	329
6.3.2	Automotive Attack Surfaces and Vulnerabilities	330
6.4	Intrusion Detection and Prevention	340
6.4.1	Intrusion Detection	340
6.4.2	Intrusion Prevention	343
6.5	Functional Safety and Security	350
6.5.1	Security for Wireless Mobile Networks	350
6.5.2	Security for Sensor Networks	354
6.5.3	Platform Security	356

6.5.4	Cloud Computing and Data Security	357
6.5.5	Functional Safety	360
6.6	Car Hacking Examples	362
6.6.1	2010: Vehicles Disabled Remotely via Web Application	363
6.6.2	2010 and 2011 CAESS Experimental Analysis	364
6.6.3	2013 Miller and Valasek Physical Hack	365
6.6.4	2015 Miller and Valasek Remote Hack	367
6.7	Exercises	368
	References and Further Reading	371
7	Mobile Apps for the Connected Car	379
7.1	Automotive IT	380
7.1.1	IT Management and Systems in the Automotive Industry	382
7.2	Agile Software Development	384
7.2.1	Challenges and Two-Speed IT	387
7.3	The Smartphone and App Market	388
7.4	iOS	389
7.4.1	The History of iOS	389
7.4.2	The iOS Platform	390
7.4.3	The iOS Architecture	390
7.5	Xcode	393
7.6	Android	395
7.7	iOS and Android in the Car	397
7.8	Objective-C, Swift, and Java App Development	398
7.8.1	Objective-C	398
7.8.2	Swift	403
7.8.3	Java	404
7.9	A Ride-Sharing Example	404
7.9.1	Core Use Cases	405
7.9.2	OOA	407
7.9.3	Design	412
7.9.4	The Ridematching Algorithm	413
7.9.5	Using Google Maps	415
7.9.6	A Code Walk Through	417
7.10	Summary and Recommended Readings	431
7.11	Exercises	433
	References and Further Readings	435
8	Carsharing	439
8.1	The Carsharing Concept	439
8.2	Example car2go	441
8.3	Use Cases and Requirement Analysis for Carsharing	442
8.4	Hardware/Software Modifications for Carsharing	446
8.5	Electric Vehicles and Carsharing	447

8.6	Carsharing Activities by Other OEMs	452
8.7	Cyber Attack Surfaces and Mitigation of Cyber Attacks	453
8.8	Conclusion	454
8.9	Exercises	455
	References and Further Reading	457
9	Car Hailing and Ridesharing	461
9.1	Introduction	461
9.2	Ride-Hailing Companies and Taxi Aggregators	463
9.3	Example Bangalore	468
9.3.1	Cab Types and Prices	468
9.3.2	Services	470
9.4	Surge Prices	472
9.5	Safety in Ridesharing	472
9.5.1	Problem Background	473
9.5.2	Initiatives to Increase Safety	474
9.5.3	Reported Crime Incidents in Ridesharing	476
9.5.4	Government Policies for Ridesharing Companies	477
9.5.5	Legal Cases and Accusations	478
9.6	Cyberattacks and Cybersecurity in Ridesharing	478
9.7	Conclusion	479
9.8	Exercises	479
	References and Further Reading	480
10	Connected Parking and Automated Valet Parking	485
10.1	Parking	486
10.2	Connected Parking	487
10.3	Parking Assistance	492
10.4	Automated Valet Parking	493
10.5	Cyber Threats	496
10.6	Intrusion Detection and Prevention	497
10.6.1	Types of Intrusion Detection Systems	497
10.6.2	Attacks Against Connected Cars	498
10.6.3	Artificial Neural Network-Based IDS Implementation	500
10.7	Conclusion and Recommended Readings	503
10.7.1	Cyber Threats and Cybersecurity	503
10.7.2	Recommended Readings	504
10.8	Exercises	504
	References and Further Reading	507
11	Advanced Driver Assistance Systems and Autonomous Driving	513
11.1	Advanced Driver Assistance Systems	514
11.2	Lane Departure Warning, Lane Keep Assistance, Obstacle Detection, and Crossing Assistance	518
11.2.1	Lane Keeping and Lane Change Assistance	518
11.2.2	Crossing Assistance	523

11.3	Image Processing and Image Analysis	525
11.3.1	Computer Vision and Machine Vision	525
11.3.2	Basic Principles of Image Processing	526
11.3.3	Detection of Moving Objects	533
11.3.4	Optical Flow Algorithm	538
11.3.5	Implementation Using MATLAB	542
11.4	Autonomous Driving	549
11.5	Regulations, Public Acceptance, and Liability Issues	558
11.5.1	Regulations and On-Road Approval	558
11.5.2	Toward a Statutory Framework for Autonomous Driving	558
11.5.3	Acceptance of Autonomous Driving and Ethical Difficulties	559
11.5.4	Test on the Autobahn	560
11.6	E/E Architectures and Middleware for Autonomous Driving	561
11.7	Cybersecurity and Functional Safety	566
11.8	Summary, Conclusion, and Recommended Readings	569
11.8.1	Recommended Reading	570
11.9	Exercises	571
	References and Further Readings	572
12	Summary, Final Remarks, Outlook, and Further Reading	581
12.1	Summary	581
12.2	Final Remarks: Wind of Change	583
12.2.1	Frugal Engineering	583
12.2.2	Rise of Asian Markets	584
12.2.3	E-Mobility	585
12.2.4	Fuel Cells	585
12.2.5	Connected Cars	585
12.2.6	Shared Mobility	586
12.2.7	Autonomous Driving	586
12.2.8	Automotive Cybersecurity	587
12.3	Outlook and Further Reading	588
12.3.1	Outlook	588
12.3.2	Further Reading	590
	References and Further Readings	591
Glossary	595
Index	623

About the Authors

Dietmar P. F. Möller is a professor in the Institute of Applied Stochastics and Operations Research at Clausthal University of Technology (TUC), Germany; a member of the Simulation Science Center (SWZ) Clausthal-Göttingen, Germany; an adjunct professor in the Department of Electrical and Computer Engineering at the University of Nebraska-Lincoln (UNL), USA; and an adjunct professor in the Department of Electrical and Computer Engineering at the University of Alabama in Huntsville (UAH), USA. He is also a member of the Board of the AMSC (Alabama Modeling and Simulation Council), USA. His other publications include the Springer titles *Introduction to Transportation Analysis, Modeling and Simulation* (2014) and *Guide to Computing Fundamentals in Cyber-Physical Systems* (2016).

Roland E. Haas is the founder and CEO of QSO Technologies in Bangalore, India. He has more than 20 years of professional experience in senior techno-managerial, business innovation, and business development assignments in Germany, the USA, India, and Japan with broad experience in automotive R&D, aerospace R&D, engineering and IT services, as well as consulting and strategy. As an entrepreneur, he shares his knowledge as a mentor for startups. He is a book author and a honorary professor at the International Institute of Information Technology (IIIT-B) and an adjunct faculty member of the Indian Institute of Science (IISc). His teachings are in mechatronics, automotive electronics, Car IT, automotive software technologies, information management, and virtual product creation.



Introduction

1

This chapter provides a brief overview of the main topics of the book. Technology is arguably the most important driving force in today's world. Recent progress in the digitalization of everyday objects is removing constraints and enabling new possibilities that affect humans' lives, enterprises, businesses, mobility, and much more. The technological progress has always had a big impact but has accelerated in recent years. The past decade has witnessed remarkable advances in digital technologies that have far surpassed the decade of personal computers through cutting-edge innovations, such as the Internet of Things (IoT) and Open Artificial Intelligence Technologies (OAIT) like Machine Learning (ML) and Deep Learning (DL), as well as Big Data Analytics (BDA), Cloud Computing (CC), and others. These technology advances are fast and breathtaking with regard to the ways they are affecting and changing humans' lives and work as well as companies' business models. The companies that use digital technologies achieve significantly higher levels of profit, productivity, and performance through smarter decision making, elimination of inefficiencies, and a better understanding of their customers (Westerman et al. 2014).

The automotive industry, which encompasses a wide range of companies and organizations, is one of the most important worldwide industries today as it becomes more aware and responsive to its surroundings. Automakers are responsible for the design, development, manufacturing, marketing, and selling of automobiles and trucks, also called motor vehicles or, in short, vehicles. These vehicles provide promising intelligent functionality and get always smarter which can be seen at the Consumer Electronics Show (CES) in Las Vegas or the International Motor Show (IAA) in Frankfurt, the world's leading trade show of the automotive industry sector. The fundamental driving forces for this development are:

- *Digitization*: Process of converting information into a digital format. In this format, information is organized into discrete units of data that can be separately addressed. Hence, digitization is the strongest and most comprehensive driver of

automotive cutting-edge innovation like connected and self-driving commercial vehicles which goes far beyond the driver assistance systems we have to date.

- *Electro Mobility:* Branch of industry that focuses on mobility needs under sustainability aspects by developing and manufacturing vehicles that carry energy storages and electric drives that can vary in degree of electrification. Today, most automakers have vehicle models with hybrid and pure electric drive in their portfolios and on the roads. In the short and midterm, this will enable a more zero emission mobility, which will bring a new quality of life to urban spaces by applying efficient strategies to decarbonize the transport sector. For the near future, this also necessitates digitally networked roadside units (RSU) which are computing devices located on the roadside providing connectivity support to passing vehicles.
- *Smart Transportation:* Digitization enables a quantum leap forward in the direction of smart cities with regard to facilitating more safety, greater efficiency, and a better quality of life due to a smarter form of mobility. There are already more mobility options available to users than ever before. These options include traditional modes of public transportation like rail, bus, paratransit, ferry, and others, as well as private and non-profit oriented mobility services. Thus, the transportation sector is exploring partnerships among the different types of providers (Dinning and Weissenberger 2017). Also, transportation safety is a critical societal issue and has become a worldwide top priority (Mendez et al. 2017). Fortunately, safety is rapidly increasing, for example, the smart vehicle contains surround-view cameras and sensors as well as other innovations so that the blind spot and associated dangers will become a topic of the past. Freight will be delivered on demand, individually, and on time. For smart cities, this kind of smart transportation is still a vision, but it is already conceivable for destinations in rural areas where the vehicle is supported by delivery drones that swarm out and fly to the final destination of delivery. Another important issue in smart transportation is the emergence and evolution of shared mobility services which is changing the field of mobility in transportation.

Technological advances in sensor and navigation technologies, the networked living space through the Internet of Things, and the advances of services in the form of an Internet of Data and Services (IoDaS) will spur the visionary and affordable mobility of the future, the so-called smart mobility.

Compared with this somewhat more futuristic vision, automakers have already incorporated different intelligent assistance and management systems in today's vehicles, one of which is a smart motor management to make the vehicle fuel efficient and environment-friendly while ensuring comfortable driving characteristics. Other systems protect the drivers and passengers by the means of innovative and intelligent active safety measures, entertain the passengers or offer access to different kinds of information sources and services in and outside of the vehicle.

1.1 The Automotive Industry

The automotive industry is one of the world's most important economic sectors by revenue (see Chap. 2). Global sales of passenger vehicles were forecast to hit >80 million vehicles in 2015. Along with China, the USA is counted among the largest automobile markets worldwide, both in terms of production and sales. Approximately 8 million passenger vehicles were sold to US customers in 2014, and around 4.25 million passenger vehicles were produced in the same year in the USA. In terms of revenue, Toyota, Volkswagen (VW), and General Motors (GM) are ranked in the top list of major automakers, while the automotive supplier industry is dominated by Bosch, Continental, Denso, and Magna (URL1 2017).

The big German automakers have been the driving force behind the German economy in the past 10 years. BMW, Daimler, and VW alone represented a considerable share of global sales in the passenger vehicle market at around 20%. Within the German Stock Exchange Index (DAX), the three corporate giants are listed in the top five.

According to the study by Roland Berger and Lazard (URL2 2017), global vehicle production was expected to grow only moderately at around 2% in 2016 and beyond. The cooperation between automakers and automotive suppliers allows the automotive industry to introduce innovative changes in technologies and new mobility concepts for vehicle usage certainly within the next 10 years. On the powertrain side, for example, the development of e-mobility is the main driving force. Technological hurdles may prevail, and a convincing business case for the end user may not be accomplished yet, but tightened emission regulations will likely have a catalytic effect over the coming years. To stay successful in this volatile and rapidly changing environment, automakers and automotive suppliers will have to increase their agility, flexibility, and speed up innovation cycles in developing and running their business. Due to the high demand for ever new innovations in mechanics, electronics, and information technology, automakers and automotive suppliers have developed an excellent knowledge base with respect to development, production, and process integration, which is also being deployed and monetized in other branches of industry.

The term innovation can be defined from a general perspective as follows: Innovation is the process translating an idea or invention into a product or service that creates value or for which customers will pay. Innovation can be divided into two categories:

- *Evolutionary innovations* (continuous or dynamic): Based on many incremental advances in technology or processes
- *Revolutionary innovations* (discontinuous innovations): Often disruptive and new, such as disruptive mobility determining efficient strategies to decarbonize the transport sector

Therefore, innovation is synonymous with risk-taking, and organizations that create highly innovative products or technologies at the frontiers of knowledge take the greatest risk because they create new markets or services.

Technological innovations at the frontiers of knowledge are often considered to be cutting-edge innovations. Cutting-edge technology refers to current and fully developed technology features, unlike bleeding-edge technology, which is so new that it poses unreliability risks to users. In this sense connectivity and connected vehicles can be regarded as cutting-edge innovations of the automotive industry. In contrast, self-driving vehicles represent a bleeding-edge innovative technology because it may pose unprecedented risks with regard to the required digitized and intelligent infrastructure and the interaction of human and self-driving vehicle. Furthermore, the technology puts pressure on governments to make regulatory changes permitting on-road testing of autonomous vehicles.

As a term, cutting-edge technology is somewhat ambiguous and often used in the context of marketing. In connection with the automotive industry, the following cutting-edge technologies are recognized as important:

- *Artificial Intelligence*: Mimics cognitive functions that typically would be associated with human intelligence such as learning and problem solving. Traditionally, AI includes disciplines like reasoning, knowledge representation, planning, learning, natural language processing, and perception. Machine Learning algorithms attempt to model high level abstraction in data and allow to increase the knowledge base from data by identifying underlying structures. AI already shows outstanding results in pattern recognition problems, such as recognizing objects in images, speech recognition, and robotics. Self-driving vehicles rely on AI for sensor fusion, perception, behavior and navigation. Deep Learning is important for intrusion detection and defense discovering intricate structures in large data sets by using backpropagation algorithm to indicate how systems should change their internal parameters that are used to compute the data representation in each layer from the representation in the previous layer.
- *Big Data Analytics*: Big data are data sets that cannot be held and evaluated in conventional databases due to their huge amount of sets (volume), their diversity in structure (variety), and their volatility and availability (velocity), the three V's. Big Data Analytics describes concepts, methods and technologies to handle, structure and visualize large amounts of data, both structured as well as unstructured. In this sense, big data represents a "data tsunami" of an exponentially growing amount of different kinds of information which is threatening to overwhelm vehicle drivers and passengers alike.
- *Internet of (Smart) Things*: The Internet of Everything (IoE) is rapidly emerging which can connect everything with anything from everywhere to anywhere at any place and any time. An autonomous driving vehicle contains a huge number of sensors and actors that need to talk to each other, to central data controllers, and to all other vehicles around it on the road as well as road side units (RSUs). Digitized road infrastructure information such as traffic signs, traffic lights, roadworks, and other components will become IoE enabled and will provide vital information to autonomous cars. Already, these systems have practical applications, for example, a parking area e-plate recognition system can connect back to any driving licensing authority if the driver fails to pay the parking fees.

Traditionally, automakers have distinguished themselves by engine performance, the powertrain, and the vehicle design itself as most important features, and customers have always carefully and critically evaluated these characteristics before making a decision which brand they want to buy. But today the automotive industry is also developing and embedding cutting-edge technologies in their vehicles such as modern information and communication technologies (ICT) which address on the one hand tomorrow's mobility needs and on the other hand today's demands of the younger generation, the so-called digital natives, to be online all the time and to access and control everything with their smartphones.

Car IT considers all of the information flowing into a vehicle and out of a vehicle or within the vehicle itself. Thus, Car IT is the key enabler for accessing innovative information technology (IT) within today's vehicles- from integrating Google or Facebook, services help finding where a car is parked, to remote functions, like closing the sunroof from afar when it rains. Car IT helps automakers to shape and adapt their vehicles to technology trends and market requirements. As a result, Car IT is a dynamically developing subject area for which there is currently no general definition available.

In order to rate the opportunities and risks of Car IT for automotive original equipment manufacturers (OEMs), their suppliers, as well as for vehicle users, Johanning and Mildner (2015) have developed a strengths-weaknesses-opportunities-threats (SWOT) analysis which has been adapted for this book, as shown in Tables 1.1 and 1.2.

Furthermore, the goal of Car IT research and development (R&D) is the definition of connected cars (see Chap. 5) and self-driving vehicles (see Sect. 5.5), including current developments in automotive electric and electronic (E/E) devices and automotive software technology (see Chap. 4), implementation variants, safety, and cybersecurity (see Chap. 6), as well as legal challenges to automotive connectivity. Thus, autonomous vehicles will become more aware, dexterous, and sensitive to

Table 1.1 SWOT analysis for automakers and Tier 1 suppliers (Johanning and Mildner 2015)

Strengths	Weaknesses
Collecting data on the behavior of the vehicle user	Missing or hard to recruit development staff
Creating an attractive user experience	Lack of experience with new IT technologies in the vehicle
Direct contact with the vehicle user	
Increasing comfort and operation	
Relieving the driver through intelligent driver information and assistance systems	
Opportunities	Threats
Developing new mobility and add-on business models	Automotive cybersecurity (hacker attack on the car)
Establishing direct and long-term customer relationships	Dealing with the information overload (big data phenomena)

Table 1.2 SWOT analysis for vehicle users (Johanning and Mildner 2015)

Strengths	Weaknesses
Access to the vehicle and its control from anywhere via app or web portal	Inadequate experience with new IT technologies in the car (specific difficulties: handling and communication)
Higher level of comfort and easy operation of the vehicle	
Higher level of safety while driving	
Opportunities	Threats
Avoiding congestion and detouring around accident situations, and thus arriving faster at the final destination	Can unauthorized persons access the vehicle?
Completing work from the vehicle (car office)	Anonymity still guaranteed?
Increasing traffic safety	Security of personal data guaranteed?

their surroundings based on the data they will generate. For example, the data generated by drones, which started flying in remote access areas at first and then moved on to more populated areas, will be combined with the streams from countless sensors instrumented in just about everything and everywhere.

The term “connected car” refers to the next generation of car technologies making use of the Internet, enabling the passengers of the vehicle to take advantage of numerous new services and features (see Chap. 5). Based on these embedded, advanced information and communication technologies, connected cars promise to provide customers with more effective and safer transportation, with less harm to the environment and increased in-vehicle comfort and safety. Thus, over the next decade, Internet-connected vehicle technologies and autonomous vehicles are set to stir up yet another era of cutting-edge innovation in the automotive sector.

The idea of fully autonomous vehicles seems to be too futuristic for many drivers right now. But for automakers, the path from current models to driverless vehicles is going to be an exciting period of transformation. For passengers, self-driving vehicles offer a comfort advantage, since the driver would be freed from any kind of driving activities. Furthermore, for the group of people who have been partially or completely excluded from the participation in public life, due to their mobility restrictions, self-driving vehicles offer new opportunities for their mobility (Friedrich 2015). All this can be accomplished through innovative developments that represent enormous opportunities, although, for the automotive industry a perilous, unsteady phase is being predicted. Thus, the original equipment manufacturers (OEMs) must navigate the challenges of designing, manufacturing, and upgrading, for example, traditional powertrain models, while staking a claim in emerging technologies and improved customer experiences (URL3 2017). In the future, data generated by these connected car technologies is not only circulating within the vehicle but also, to a large extent, outside of the vehicle making use of

new cloud services offered by the automakers and their suppliers. Therefore, security of data becomes a key issue for the industry and ultimately for vehicle users.

Furthermore, connected car services at the cutting edge of innovation will require cost-intensive special equipment. To keep the costs of these services low, some automakers are offering their customers monthly holdback payment purchasing using mostly cloud-based, connected car services. Thus, some automakers offer selected services directly to end users to reinforce the attraction of their brand while not competing with their brand dealership. This requires a deep transformation in the business model from the traditional business-to-business (B2B) model to a business-to-business-to-customer (B2B2C) model. To evolve this B2B2C model, the respective leading automakers are relying heavily on digital technology, such as mobility, social media, analytics, and smart embedded devices. But the technology necessary to manufacture connected, intelligent, and autonomous vehicles is not within the traditional scope of automakers. This, of course, is an invitation to high-tech companies, such as Apple, Google, and others to develop their own technologies and communications systems for critical components of the networked and autonomous vehicle ecosystem, as reported by PricewaterhouseCoopers (PwC) in ([URL3 2017](#)). These companies will likely prove to have a major influence on the automotive sector in the coming years, mainly because their skills and the industry's needs align perfectly. They are adept at seamlessly and efficiently connecting components to create networks highly valued by consumers for the information, entertainment, and experiences they deliver ([URL3 2017](#)).

In addition, connectivity can also increase road safety and, hence, improve the transit experience. But the more vehicles become connected, the more they are vulnerable to cyber attacks. Being no longer a topic of science fiction, recent events have shown that cyber threats and cybercrime can affect all passenger cars and commercial vehicles equipped with embedded telematics or connectivity solutions from the aftermarket. Thus, automotive cybersecurity is quickly becoming an important factor when purchasing a modern vehicle, due to the increasing proportion of software, digital components, and systems onboard connected to surrounding digital infrastructure. Consequently, this book discusses the situation in which the automobile industry finds itself and addresses the opportunities, challenges, and threats of the digital transformation and the connected vehicle ecosystem.

1.2 Scope of This Book

The automotive industry will be facing numerous sweeping and interlinked changes in the next several decades. Unlike most other industries, the automotive industry, while incorporating modern Internet network-enabled technology, has been forced to completely and fundamentally reinvent itself ([URL4 2017](#)). Compared to other industries, the automotive industry has taken advantage of many efficiency improvements driven by Internet-based technology but has also remained in the

same structure, as opposed to reorganizing its whole ecosystem. There could be a reconceptualization of how the core activity is organized, coordinated, and executed. A number of factors could push the automotive industry into new alliances and organization structures, perhaps ultimately towards futuristic concepts such as smart mobility. Smart mobility characterizes the visionary mobility of the future, available for everyone regardless of location and region, regardless of periods of use and duration, as well as regardless of individual ability and budget (Flügge 2016).

The many new features of the networked vehicle will begin with Google's and Facebook's involvement and extend to services that help users find where they parked their cars, control functions via app remotely, like closing the sunroof when it rains, and ultimately lead to completely new services based on the car data being generated, as described in (Johanning and Mildner 2015). Thus, the automotive industry will be facing a situation of profound change and opportunity in the coming decades due to disruptive innovations (Meyer and Shaheen 2017), which not only substitute existing solutions but will also create new markets and change society with regard to smart mobility, for example:

- Introduction of self-driving vehicles
- Energy- and emission-efficient innovations
- New models of smart transportation and service delivery
- Sharing economy and multimodal mobility
- 3D printing of automotive spare parts

In this context, smart mobility can become a motivator for own projects within the framework of a holistic mobility management. It is an offer that is primarily intended to enable energy-efficient, comfortable, and cost-effective mobility. It also is a paradigm shift to a more flexible and multimodal transport system for hassle-free usage of multiple modes of shared and public transport as key for inner city areas, an example being a proximity-based service that shows information if and when passengers really need it, whereby an integrated mobility platform as information broker allows seamless travel across transport modes. Smart mobility will see the emergence of new business models, for example, Mobility-as-a-Service (MaaS).

The term “connected cars” means that vehicles are now more becoming part of the connected world, continuously Internet connected, generating and transmitting data, which on the one hand enables applications, such as the broadcast of real-time traffic alert to smart watches, but which also raises security and privacy concerns. The decisive feature of a connected car is the ability to do network, both internally as well as externally, with smart devices, other cars, the Internet and applications and platforms on the cloud. With the mandatory introduction of the automatic emergency call system e-call, in the EU, from March 2018, virtually every newly built vehicle will be a connected car.

In the context of the above topics, this book gives a detailed overview of automotive connectivity and the associated cybersecurity issues.

1.3 Overview of Topics

The automotive industry is facing profound changes and opportunities; automakers are dealing with new technologies and vehicle concepts that have the potential to transform the vehicle itself. What is already emerging is the beginning of the connected vehicle, for example, a fully digitalized vehicle with wireless fidelity (Wi-Fi), a wireless networking technology that allows computers and other devices to communicate over the air (OTA). Wi-Fi is based on one of the 802.11 standards developed by the Institute of Electrical and Electronic Engineers (IEEE) and adopted by the Wi-Fi Alliance® for advanced infotainment systems and apps. Furthermore, these vehicles use vehicle-to-vehicle (V2V) communication technology to talk to each other, exchanging essential safety data, such as speed and position, real-time location services and routing based on traffic conditions as well as networked web links, facilitating vehicle diagnostics, maintenance, intervals, and repairs (URL3 2017).

This digital transformation requires a thorough theoretical background on the respective methods and technologies like automotive connectivity, Car IT, autonomous, self-driving vehicles, and automotive cybersecurity. This book also provides a framework within which the reader can integrate the associated essential knowledge from:

- Automotive research and development
- Automotive mechatronics
- Automotive electric and electronic (E/E) systems
- Automotive software technology
- Automotive cyber-physical systems
- Advanced driver assistance systems (ADAS)
- Automotive cybersecurity

Without such a reference, the practitioner is left to ponder the plethora of terms, standards, and practices that have been developed independently and which often lack cohesion, particularly in nomenclature and emphasis. Hence, the intention of this book is to give a comprehensive overview of automotive connectivity and to provide a framework for discussing the many challenges and issues associated with automotive connectivity, both from a technical as well as a business oriented perspective. The chapters are entitled:

1. Introduction
2. The Automotive Industry
3. Automotive Research and Development
4. Automotive E/E and Automotive Software Technology
5. The Connected Car
6. Automotive Cybersecurity
7. Mobile Apps for the Connected Car

8. Carsharing
9. Car Hailing and Ridesharing
10. Connected Parking and Automated Valet Parking
11. Advanced Driver Assistance Systems and Autonomous Driving

The final chapter is Chap. 12.

12. Summary, Outlook, and Final Remarks

Against this background, the book covers, in contrast to other books which focus more on automotive E/E and software technology (Reif 2014, Borgeest 2013, Schäuffele and Zurawka 2013), the essential methodological and theoretical basics from mechatronics, computer networks, distributed systems, software engineering, systems engineering and IT security and elaborates the necessary technological adaptations in future vehicles (Siebenpfeiffer 2014, Swan 2015).

Cyber-physical systems (Möller 2016) in this regard are the backbone of these technologies. These are engineered systems which have significant couplings between cyber (processing, communication, and network) and physical (sensing, actuation, and infrastructure) elements. The couplings result in the dynamic coevolution of cyber and physical properties. In the context of connected cars and self-driving vehicles, the physical domain is defined by the dynamics of vehicle motion together with the dynamics of radio wave propagation. The cyber domain is defined by the data processing in the intra- and inter-vehicle networks and the vehicle-to-infrastructure (V2I) data exchange. This enhanced complexity has also had a huge impact on the vehicle design process, its modularization with the associated platforms, virtual product creation, and the life cycle management for connected cars and autonomous driving vehicles. Based on that background, the required needs in automotive E/E and automotive software technology, as well as the evolution of the connected car, will be derivable. The evolving strong connectivity of future vehicles necessitates a thorough analysis of the vulnerability of connected cars and measures to prevent cyber attacks on vehicles by making use of cybersecurity methods (Graham et al. 2010).

The integrated use cases from different sectors of the automotive domain give a practical perspective and a detailed insight into mobility applications, which are of interest to vehicle users and illustrate new business models for automakers and their suppliers. Chapters 10 and 11 discuss different advanced driver assistance systems (ADAS) and the underlying technologies which support the driver by increasing vehicle safety and are one of the fastest growing segments in automotive electronics. Industry-wide quality standards in vehicular safety systems, such as ADAS, are based on ISO 26262, Road Vehicles—Functional Safety, the international standard for functional safety of electrical and/or electronic systems in automobiles.

Therefore, current scenarios appear as the result of, and conditions for, the development of urban dynamics, considered from the perspective of functions, relations, and actors involved. Functional dynamics relate to patterns of generation and the demand for energy, information, and transportation of goods and people and

more. Some of these patterns affect the spaces of social life by occupying them or conditioning their perception. In contrast, relational dynamics refers to the required quality of social life at any given time (Garcia-Verdugo 2017).

References and Further Reading

- (Borgeest 2013) Borgeest, K.: Electronics in Vehicle Technology–Hardware, Software, Systems, and Project Management (in German). Springer Publ., 2013
- (Dinning and Weissenberger 2017) Dinning, M., Weissenberger, T.: Multimodal Transportation Payments Convergence – Key of Mobility, pp.121–133, In: Disrupting Mobility – Impacts of Sharing Economy and Innovative Transportation on Cities, Springer Publ., 2017
- (Flügge 2016) Flügge B. (Ed.): Smart Mobility–Trends, Concepts, Best Practices for Intelligent Mobility (in German). Springer Publ., 2016
- (Friedrich 2015) Friedrich, B.: Traffic Impact of Autonomous Vehicles (in German), pp. 331–350, In: Maurer, M., Gerdes, J. C., Lenz, B., Winner, H., (Eds.) Autonomous Driving – Technical Aspects and Societal Aspects, Springer Publ., 2015
- (Garcia-Verdugo 2017) Garcia-Verdugo, L. V.: Mobilescapes: A New Frontier for Urban, Vehicle and Media Design, pp. 335–349, In: Disrupting Mobility – Impacts of Sharing Economy and Innovative Transportation on Cities, Springer Publ., 2017
- (Graham et al. 2010) Graham, J., Olson, R., Howard, R.: Cyber Security Essentials. CRC Press, 2010
- (Johanning and Mildner 2015) Car IT Compact – The Car of the Future - Driving Connected and Autonomously (in German). Springer Publ., 2015
- (Maurer et al. 2015) Maurer, M., Gerdes, J. C. Lenz, B., Winner, H. (Eds.): Autonomous Driving – Technical, Legal and Social Aspects (in German). Springer Publ., 2015
- (Mendez et al. 2017) Mendez, V. M., Monje, C. A., White, V.: Beyond Traffic: Trends and Choices 2045 – A National Dialouge About Future Transportation Opportunities and Challenges, pp.3–20, In: Disrupting Mobility – Impacts of Sharing Economy and Innovative Transportation on Cities, Springer Publ., 2017
- (Meyer and Shaheen 2017) Meyer, G., Shaheen, S. (Eds.): Disrupting Mobility – Impacts id Sharing Economy and Innovative Transportation on Cities. Springer Publ., 2017
- (Möller 2016) Möller, D.P.F.: Guide to Computing Fundamentals in Cyber-Physical Systems – Concepts, Design Methods, and Applications. Springer Publ., 2016
- (Reif 2014) Reif, K.: Automotive Electronics (in German). Springer Publ., 2014
- (Schäuffele and Zurawka 2013) Schäuffele, J., Zurawka, T.: Automotive Software Engineering – Efficient use of Basics, Processes, Methods and Tools (in German). Springer Publ., 2013
- (Siebenpfeiffer 2014) Siebenpfeiffer, W. (Ed.): Networked Automobile – Safety, Car IT, Concepts (in German). Springer Publ., 2014
- (Swan 2015) M. Swan.: Connected Car: Quantified Self becomes Quantified Car.
- (Westerman et al. 2014) Westerman, G., Bommet, D., McAfee, A.: Leading Digital. Harvard Business Review Press, 2014

Links

- (URL1 2017) <https://www.statista.com/topics/1487/automotive-industry/>
- (URL2 2017) <https://www.rolandberger.com/en/press/Automotive-Suppliers-2016.html>
- (URL3 2017) <http://www.strategyand.pwc.com/media/file/2016-Auto-Trends.pdf>
- (URL4 2017) <http://www.mdpi.com/2224-2708/4/1/2/>



The Automotive Industry

2

This chapter provides an overview of the global production and sales of the automotive industry. Thus, Sect. 2.1 reports on the current global automotive market. The focus of Sect. 2.2 is on the megatrends in the automotive industry, such as tighter emission controls and the rise of electric vehicles (Sect. 2.2.1), car ownership versus mobility Sect. 2.2.2, and Chaps. 5 and 8), connectivity (Sect. 2.2.3), advanced driving assistance systems (ADAS) (see Chap. 11) and autonomous driving (Sect. 2.2.4 and Chap. 6), and digitalization (Sect. 2.2.5). Section 2.3 introduces the supply chain between original equipment manufacturers (OEMs) and suppliers. Section 2.4 describes new players and challenges. Finally, Sect. 2.5 introduces the background of the digital transformation in the automotive industry. Section 2.6 contains a comprehensive set of questions on the challenges, while the last section includes references and suggestions for further reading.

2.1 The Automotive Market

The automotive industry is one of the most important industries in the world generating a total revenue of more than 3 trillion € in 2015 (URL1 2017) producing nearly 95 million units (passenger cars, light commercial vehicles, minibuses, trucks, buses, and coaches) in 2016 (URL2 2017). There are more than 1 billion (bn) cars in use worldwide. Traditionally, the product spectrum is divided into passenger cars and commercial vehicles. The term passenger car does not only include the classic sedan and station wagon type of vehicles but also encompasses sport utility vehicles (SUVs) and multipurpose vehicles (MPVs). The segment of light commercial vehicles includes pickup trucks, which are particularly popular in the USA, and is defined by a weight class of <3.5 t trucks (medium >3.5 t and heavy), buses, and coaches which form the classic commercial vehicle segment.

In Germany, the automotive industry and its vast supply chain account for 20% of its overall industry production with a turnover of more than 400 billion €

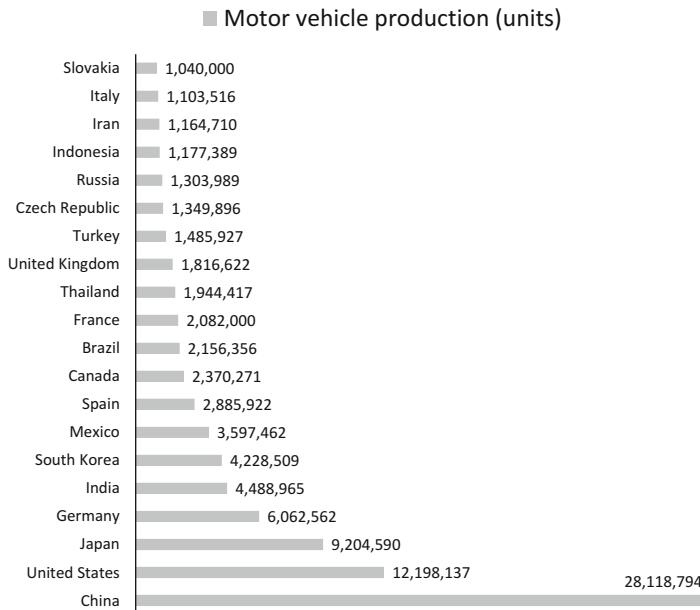


Fig. 2.1 Global vehicle production by country (see URL11 2017)

(URL3 2017). Other countries with large automotive industries are France, Spain, Italy, Great Britain, Japan, the USA, Mexico, South Korea, and China, as can be seen in Fig. 2.1.

One of the most noticeable trends is the shift toward Asia where China is increasing its lead as the most important automotive market, both in terms of production and sales (URL1 2016). China alone is responsible for more than 30% of all vehicles produced and sold globally (URL18 2017), as shown in Fig. 2.2.

The global automotive market has always undergone cycles. The last years after the financial crisis in 2007–2008 have seen an incredible growth driven by low fuel prices and low interest rates. The US market has recovered in an amazing way and this happened after a near bankruptcy of the leading automotive manufacturers in Detroit (Dietz et al. 2016). Figure 2.3 gives a perspective of the market trends, comparing the production numbers of the so-called Triade (NAFTA, Europe, Japan, Taiwan, Hong Kong, South Korea, Singapore) with BRIC (Brazil, Russia, India, and China) and the Rest of World (RoW) for the years 2000 and 2014 (Dietz et al. 2016).

Registration and production numbers in a particular country differ (Dietz et al. 2016). Examples are Germany and the USA. Germany is the number one exporter in terms of the size of its own market and produces more than 6 million cars, while the USA is the number one importer. The USA produced >12 million cars in 2016, while the total market size of new cars sold was >16 million. The latest registration statistics for the first quarter of 2017 are shown in Fig. 2.4.

By far, the biggest market is China (Dietz et al. 2016). The growth has been unprecedented if one takes into account that the Chinese market was only one-third

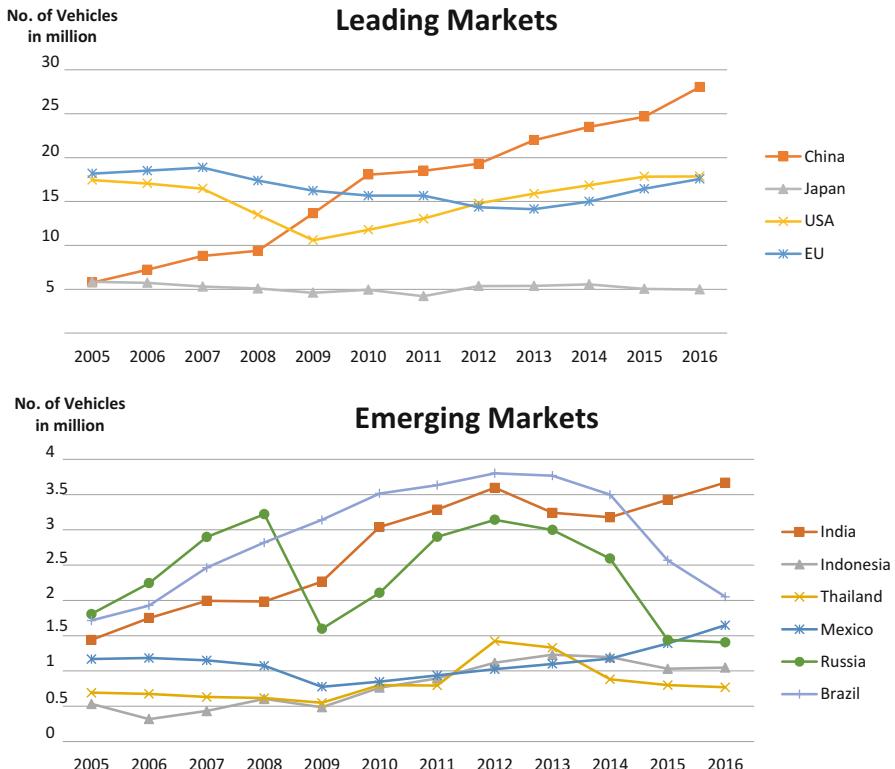


Fig. 2.2 Global vehicle sales by region (see URL1 2016)

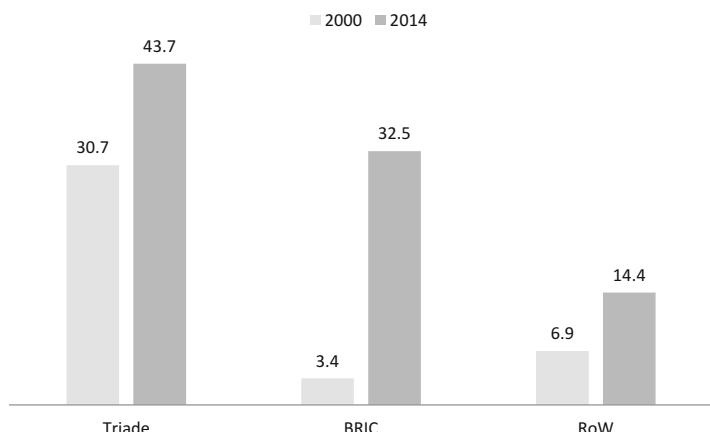


Fig. 2.3 Worldwide production of cars (sources: International Organization of Motor Vehicle Manufacturers OICA (URL2 2014), (Dietz et al. 2016))

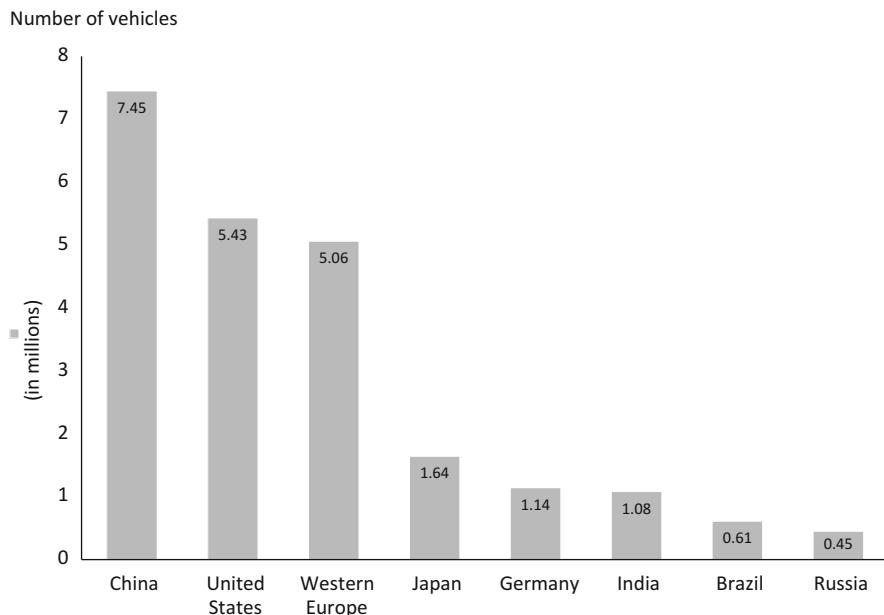


Fig. 2.4 Registration numbers in the first quarter of 2017 ([URL11 2017](#))

of this size in 2008 (Dietz et al. 2016). Europe accounts for a little more than 15 million units, as it can be seen in Fig. 2.2. This is roughly the same number as the number of cars sold in the U.S. ([URL1 2016](#); [URL18 2017](#)).

The passenger car market in India has been sluggish from 2008 to 2013 but showed promising signs of healthy growth during the last few years ([URL15 2017](#)). It has surpassed 3.5 million passenger cars annually with a growth of more than 10% ([URL14 2017](#)). The leading passenger car manufacturers in India, by unit sales volume, are Maruti Suzuki, Tata Motors, Mahindra, and Hyundai ([URL14 2017](#)).

Worldwide, 2.9 million trucks were sold in 2016. One out of every three of these was sold in China, i.e., in 2016 that amounted to nearly 1 million units ([URL4 2017](#)). In India, the market reached nearly 300,000 units in 2016, an increase of 7% compared to 2015 ([URL4 2017](#)).

The worldwide number of buses sold is around 500,000; 170,000 of those in China ([URL16 2017](#)). India is already the second largest market for buses, as measured by absolute numbers, and the fastest growing. New entries, such as Daimler's Bharat-Benz, have created competition and eat up market shares from established players, such as Ashok-Leyland and Tata Motors.

The turnover by revenue of the largest manufacturers of commercial vehicles worldwide is shown in Fig. 2.5. If one looks at the number of units sold, the ranking is different; then, the top producers come from China, with Dongfeng at the top ([URL19 2017](#)).

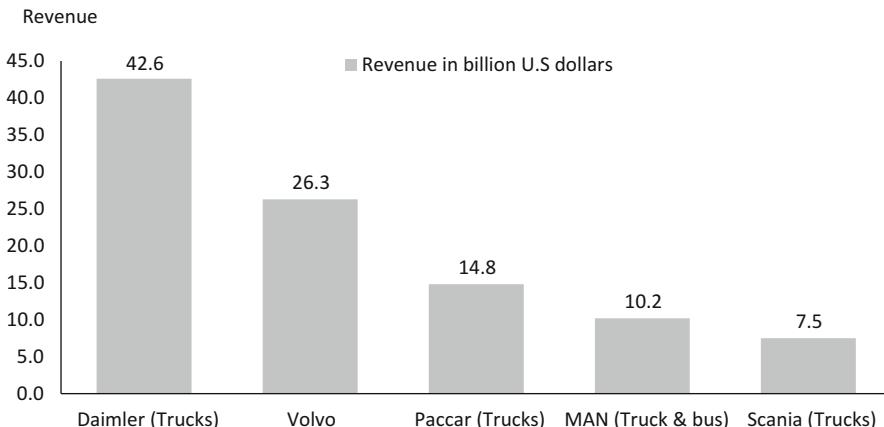


Fig. 2.5 Largest commercial vehicle manufacturers by revenue in FY 2015, in million USD (URL1 2017)

An important figure is the number of cars per person, which differs widely. Saturated markets, such as the U.S., have more than one car per every three citizens, as shown in Figs. 2.6 and 2.7 (Dietz et al. 2016). In 2005, the European market had, on average, 448 cars per 1000 inhabitants, China had 11 cars per 1000 inhabitants, and India had only 6 cars per 1000 inhabitants.

The graph in Fig. 2.7 shows the situation in 2012. While the vehicle density numbers for the U.S., Europe, and Japan have not changed that much, the number of cars per 1000 inhabitants has literally exploded in China. Also, India has seen a near doubling of the numbers for 2005. This clearly shows the potential of the Chinese and the Indian markets in the years to come.

The automotive aftermarket revenues in Germany from 2007 until 2015 are shown in Fig. 2.8. It reached a turnover of nearly 42 bn € in 2015, while the European aftermarket reached a total revenue of more than 180 bn €.

Currently, there are 45 million cars in Germany; 25% of these are older than 8 years. Every year the ownership of over 6 million used cars changes. The aftermarket is very important as it contributes to the bottom line of retailers and garages in a major way (Reindl et al. 2016).

One typically differentiates between:

- Accident repair
- Wear and tear repair
- and Maintenance

As the quality of cars has increased significantly with less wear and tear repairs today, there are fewer than 0.8 repair jobs per vehicle, per year (Reindl et al. 2016).

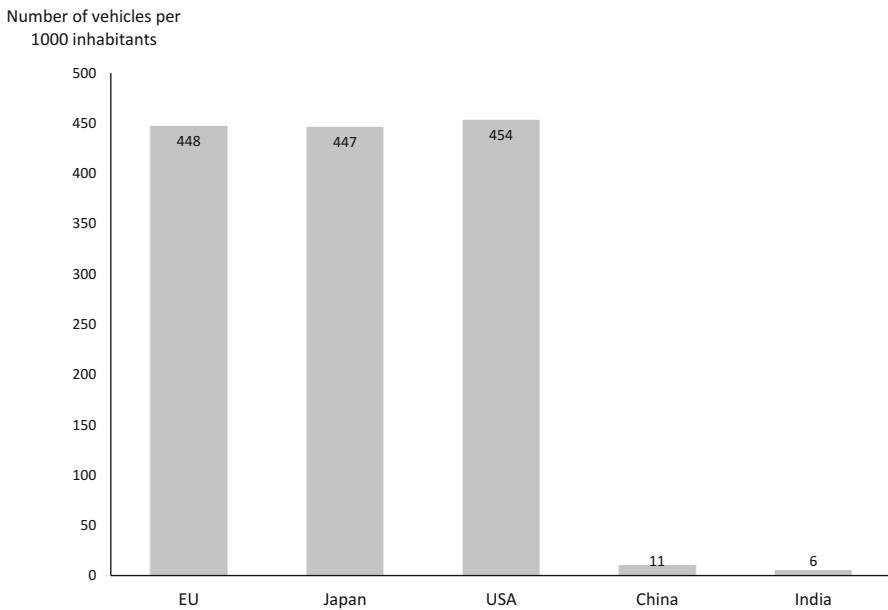


Fig. 2.6 Vehicle density (number of cars per thousand inhabitants) in 2005 (sources: European Automobile Manufacturers Association ACEA ([URL31 2017](#)), International Organization of Motor Vehicle Manufacturers OICA ([URL16 2015](#)), Dietz et al. [2016](#))

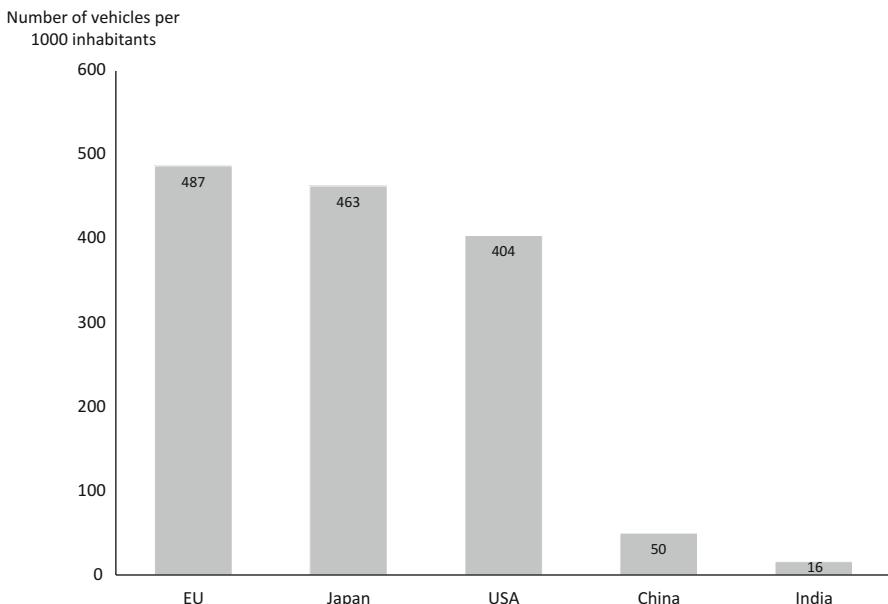


Fig. 2.7 Vehicle density (number of cars per thousand inhabitants) in 2012 (sources: European Automobile Manufacturers Association ACEA ([URL31 2017](#)), Dietz et al. [2016](#))

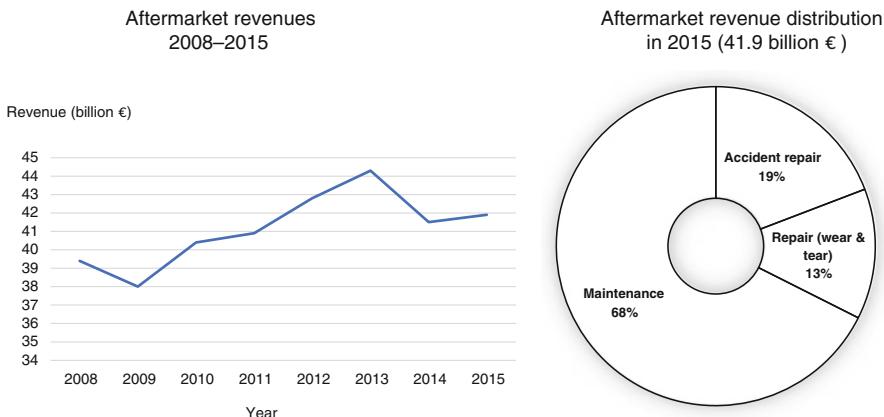


Fig. 2.8 The automotive aftermarket in Germany (source: DAT report 1995–2015, see also Reindl et al. (2016))

There is fierce competition going on between the 38,000 branded and independent garages in Germany. While new cars (Segment I, < 4 years) are predominantly serviced in OEM-branded workshops, older cars (Segment III, > 8 years, and Segment IV, >10 years) are often taken to independent repair shops because of their lower costs.

Another important player in the automotive market are the car insurance companies. They face huge cost pressures; but as car insurance policies are a means to connect with customers, they are an essential part of the service offerings.

Usage-based insurance (UBI), digital retail, and connected aftermarket services are upcoming trends that are based on connectivity and the digital transformation of value chains.

2.2 The Automotive Megatrends

In addition to traditional combustion engine vehicles with their carbon emissions, the number of electric vehicles is on the rise. Therefore, the automotive industry is facing a challenge in vehicle powertrain technology. Also, car ownership has become less important; and mobility on demand focuses more on the flexibility to choose between different modes of transportation. Another trend is vehicle connection with the Internet, which is becoming an important criterion for vehicles because of their increasing connectivity to other systems. However, with connectivity comes the threat of cybercriminal attacks with different kinds of risks; and the automakers are faced with the need for intrusion detection and defense against malware. Finally, embedding of digital technologies will change automotive industry business models and will provide new revenue and value-producing opportunities.

2.2.1 Tighter Emission Controls and the Rise of Electric Vehicles

There is a rising concern about health problems in Europe, the USA, and emerging countries due to carbon emissions from a rapidly growing fleet of cars. The problem can clearly be seen if one considers the enormous growth in car ownership and car density, as shown in Figs. 2.6 and 2.7.

Moreover, if one looks at the concentration of cars in Asia, for example, where the largest concentrations of cars are mostly in large cities with growth rates of more than 15%, the seriousness of the situation becomes clear. The metropolitan areas in crowded Asian countries are suffering from an increasing pollution load of small particles. The smog in China's capital, Beijing, has become infamous with a rise in particulates, especially during the winter season, from November to April, when heating is being used. Indian cities, such as Delhi, Mumbai, and Bangalore, also suffer from severe traffic congestion and smog, as shown in Fig. 2.9.

The danger to the health of citizens is now well documented and can no longer be ignored. China already has enforced electrical propulsion for two wheelers in cities (Hinderer et al. 2016). Other countries will follow. Also, many second and third tier

Fig. 2.9 Traffic jam in Bangalore, India



cities in India are suffering from high pollution due to vehicle emissions. Figure 2.9 shows the congested traffic situation during rush hour in Bangalore.

Vehicle emissions account for the majority of the particle emissions in southern Indian cities. The Volkswagen (VW) “dieselgate scandal” (Gates et al. 2015; URL12 2016) has accelerated the shift towards electric cars and boosted electrical drivetrain technologies (Hinderer et al. 2016). The Volkswagen group has committed itself to offering a full range of electric cars and is focusing on the electric drive as the dominant powertrain technology (URL2 2016).

Recently, Germany’s state government discussed legislation that bans combustion engines from 2030 onward (Schmitt 2016a). This is an ambitious goal, which certainly may be relaxed and weakened (Schmitt 2016b). However, it shows a clear trend and a general social acceptance of electric cars (URL8 2016; Kampker et al. 2013; Hinderer et al. 2016).

The BMW i8 hybrid, shown in Fig. 2.10, is an example of the shift from the classical internal combustion engine powertrain (ICE) to hybrid and full electrical powertrain technologies.

Nevertheless, the number of electric vehicles sold in Europe so far is still small. Some countries, such as Norway, have taken the lead; but in Germany, for example, actual sales still lag behind the original government plans and projections (Hinderer et al. 2016). This is due to several reasons, most importantly:

Fig. 2.10 BMW invests heavily in e-mobility, e.g., the hybrid i8 seen here at the 2016 Paris Motor Show



- Cost of the vehicle, with the battery as the primary cost driver
- Inadequate charging infrastructure
- Limited range
- Time needed for charging

Without an adequate charging infrastructure, sales of electric vehicles will remain slow; and without enough vehicles on the road, there is no incentive for investors to provide an adequate charging infrastructure. The same holds true for battery prices. They remain high when only a few electric cars are being sold; and the high battery prices in turn affect the attractiveness of electric cars. Another factor which slows down the market penetration of electric cars is the competing charging standards. In Europe, there are at least three infrastructure standards for fast charging (Kampker et al. 2013; Hinderer et al. 2016):

- *Charge De Move (CHAdeMO™)*: Trade name of a cross-brand electrical interface of a battery management system for electric vehicles, developed in Japan. With this DC-based interface, the accumulator of an electric vehicle or plug-in hybrid vehicle can be charged directly with high-voltage electrical power up to 43 kWh.
- *Combined Charging System (CCS)*: A quick charging method for electric vehicle batteries, delivering high-voltage DC via a special electrical connector with high charging power up to 50 kWh.
- *Type 2/Mode 3*: Load clutch connector for charging electric vehicles.

Another option was recently introduced by Bosch, the so called charging app (URL1 2018), an approach described in the following six steps:

- Step 1: Register and download app for Android or iOS for free and register once - without contract and basic charges.
- Step 2: Searching with map / filter to have the nearest charging station automatically displayed and refines the search via address input or filter.
- Step 3: Plan route and app will navigate you to the nearest available charging point.
- Step 4: Control charging meaning watch the entire charging process through the app and start or stop at any time.
- Step 5: Paying by easy payment via Paypal, credit or debit card.
- Step 6: View history by keeping an eye on all downloads and costs in your logbook.

Fortunately, things are changing; and favorable governmental policies, social trends, and upfront investments in fast-charging infrastructure are turning the market step by step. This is also beginning to affect battery costs; and over the last year, one could see significant drops in battery prices (Hinderer et al. 2016).

Ambitious projects, such as Tesla's Gigafactory, a joint venture with Panasonic, are expected to accelerate the trend of declining costs (Kampker et al. 2013).



Fig. 2.11 Daimler started the new EQ brand for the company's e-mobility activities

A price range of 200 \$US per kWh is seen as a game changer, where the cost of electric vehicles will actually fall below the costs of internal combustion engine (ICE) cars (Hinderer et al. 2016).

There are already several electric vehicle models available in the European market, e.g., BMW's i3 and the hybrid i8, shown in Fig. 2.10, Nissan's Leaf, and Tesla's Model S and the future Model X (Braun 2016). Renault already has quite a bit of experience and has experimented with new designs. Recently, they announced the new Renault Zoe with a range of up to 400 km.

Mercedes has launched a new brand called EQ for its e-mobility activities, as shown in Fig. 2.11 (URL9 2016; URL10 2016; URL13 2016). So far, the model lines with pure electric drive comprise the B class and the Smart-E-ForTwo. Also, a new smart Smart ForFour was recently introduced as an e-drive version. It is interesting to note that electric vehicles were quite common in the early days of the automotive industry (Kampker et al. 2013), so in a way, the industry has come full circle. Figure 2.12 shows a picture of such a vehicle, which was presented at the eCarTec in Munich in October 2016 (URL11 2016).

2.2.2 Car Ownership Versus Mobility

Over the last few years, a clear trend has emerged in Europe, especially among the younger population (Knieps 2016). Car ownership has become less important, and many younger people don't even have a driver's license anymore (Haas 2015). The main focus

Fig. 2.12 Electric cars are not new; a vintage e-car in Munich



is on mobility and the flexibility to choose between different means of transportation, such as train, bus, taxi, aircraft, shared car, etc. A car is regarded as a costly asset, which refers not only to the purchase price but to many other factors too, such as:

- Depreciation costs, which are typically very high in the first two years
- Fuel costs
- Insurance
- Maintenance and repair costs
- Parking space, which is a particular problem in metropolitan areas (Rees 2016)
- Taxes

A similar trend can be seen in other economic sectors, too (URL2 2015). Airbnb, for example, has threatened the classical hotel business as practically everyone can rent out spare rooms to visitors using the Airbnb platform (URL23 2017). As the billing is done exclusively through the Internet platform, there is no problem with no-shows and late cancellations.

For many younger people, car ownership has lost its attractiveness as there are many alternative means of transport, such as car sharing, car rental, ridehailing, public transport, etc., with no fixed costs, pay-per-use business models, and a high degree of flexibility. However, this flexibility also has a flip side that is discussed extensively in (Freitag 2016; Meyer and Shaheen 2017; Schultz 2016).

2.2.3 Connectivity

Connectivity refers to the connection between cars and other systems, such as Car2Car (C2C) or Vehicle-2-Vehicle (V2V), Vehicle-2-Infrastructure (V2I), and Vehicle-2-Backend (V2B), and often involves a connection to the Internet (see Chap. 5, and Siebenpfeiffer 2014). This concept and the related business models have the potential to disrupt the automotive industry, as illustrated in Fig. 2.13. Driven by the rapid adoption of the smart phone, car owners have become demanding. Forecasts predict that nearly every car sold in 2025 will be connected penetration of connected cars in developed countries by 2025 (URL1 2013). Connectivity is typically based on a global system for communication (GSM), a connection which provides access to the Internet and backend systems (Spehr 2016; Johanning and Mildner 2015). Navigation benefits largely from connectivity as traffic information can be shared in real time. An important topic will be C2I communication as this provides the basis for advanced driver assistance systems (ADAS) (see Chap. 11) and higher levels of automation. Connected cars provide a platform for many new services and stronger customer interaction between OEMs and customers (see Chap. 5 and Viereckl et al. 2016). However, with connectivity comes the threat of cyberattacks (see Chaps. 5 and 6, and Greenberg 2013; Lobe 2016; Grünweg 2016a; Gerhager 2016).

Multiple cyberattacks were reported over the last few years, and automotive OEMs now take this threat very seriously (Gerhager 2016); even Google's autonomous car and mobility division is concerned (URL29 2017). Many have started to include intrusion detection and prevention systems that constantly monitor the data which is being exchanged between the outside world and the car's internal electrical/

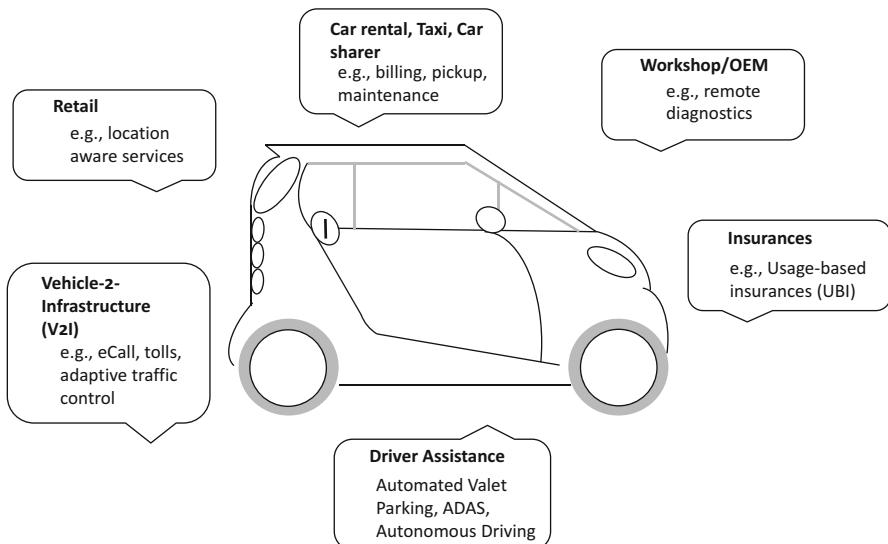


Fig. 2.13 Application areas of the connected car (see Doll and Fuest 2015)



Fig. 2.14 Car Hacking Village at the 2016 DEF CON conference

electronic (E/E) systems (see Chaps. 4 and 6 and Haas et al. 2017). This topic is also a current theme at conferences on cybersecurity, as shown in Fig. 2.14. The 2016 DEF CON® Hacking Conference (URL24 2017; URL25 2017) organized a special session, called Car Hacking Village, that dealt with the topic of cybersecurity in cars and invited interested parties, such as students, professionals, and automakers, to discuss and learn about car hacking, automotive cybersecurity, and protection mechanisms (Haas et al. 2017; Möller et al. 2017; URL25 2017).

2.2.4 Safety and Advanced Driver Assistance Systems

The effect of regulatory measures and the introduction of safety systems can be seen very clearly when looking at the number of fatal traffic accidents in any given year. Figure 2.15 shows these numbers with regard to the German Automotive Trust (DAT) report and (Dietz et al. 2016) for Germany from the early 1950s up to now (URL30 2017).

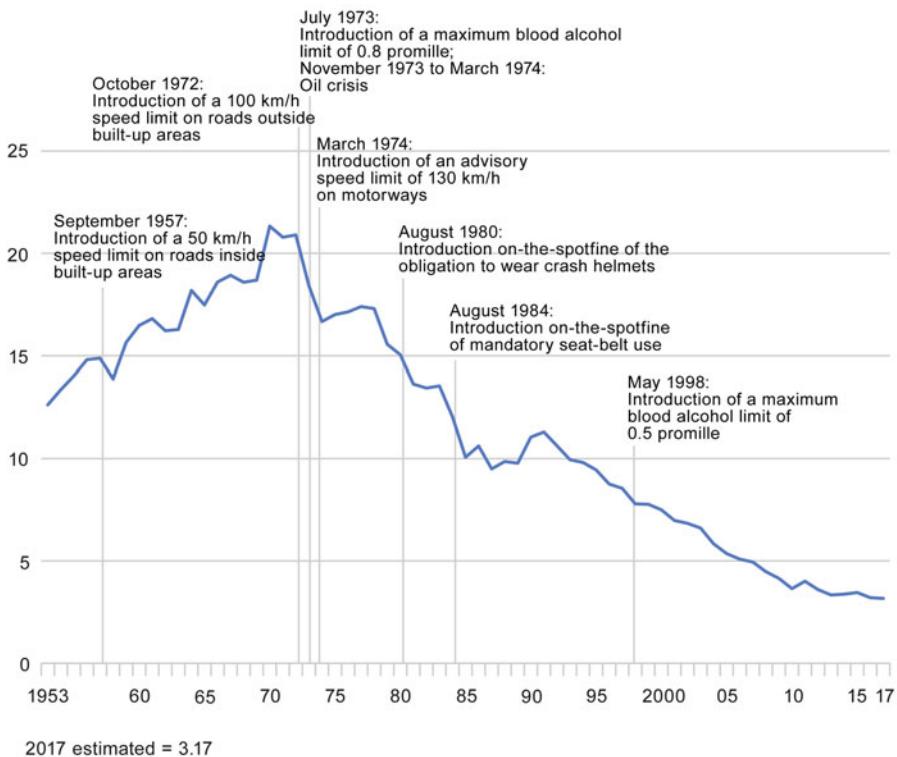
Major deflection points are due to the following:

- Introduction of speed limits (50 km/h) within cities in 1957.
- Introduction of speed limits (50 km/h) on roads (except motorways) outside of cities in 1972.
- Introduction of a general limit of alcohol blood level permitted for driving in 1973.
- Introduction of a 0.5 per 1000 limit for blood alcohol.
- Safety belts became mandatory.

It is important to note that the years 1950–1970 saw a tremendous increase in road traffic. Without regulatory measures and their proper enforcement the number of

Trend in the number of persons killed in road traffic accidents

Thousand



© Statistisches Bundesamt (Destatis), 2018

Fig. 2.15 Impact of regulations and safety measures on traffic casualties ([URL30 2017](#))

fatal accidents would have exploded (Dietz et al. 2016). Today, the number of fatal road accidents in Germany has come down to around 3300 per year. This, of course, should still be decreased substantially but to set this number in perspective, it is interesting to look at China and India, where fatal road accidents with respect to the total population size are many times higher.

Automotive manufacturers continue to work on increasing safety in various ways:

- Advanced driving assistance systems (ADAS)
- Passive safety measures (crashworthiness improvements to the car body)
- Protection for pedestrians (e.g., soft bumpers)

The development of sensor technology and signal processing algorithms laid the foundation for the rapid development of the ADAS market. With the increased safety standards and consumer demand for safety performance, the ADAS market has become one of the fastest-growing segments in automotive E/E systems (see Chaps. 4 and 11). Although the technological barrier for unmanned driving, also called autonomous driving, is relatively high, this area is seen as an attractive opportunity for high-tech companies (such as Google) to enter the automotive industry. The development of unmanned vehicles may drive efficient automotive sharing and improve vehicle utilization resulting in a significant reduction in traffic accidents, which will have a disruptive impact on OEMs, parts manufacturers, and car financing and insurance companies (Grünweg 2016b; Beck 2016; URL5 2016; Freitag 2016).

2.2.5 Autonomous Driving

Autonomous driving is one of the most important cutting edge technological innovations in the automotive industry today (Maurer et al. 2015). However, it is by no means a new topic. Research in this field dates back decades, for example, the Prometheus project (URL26 2017). Daimler, for example, was active in a research project in the 1990s to explore the possibility of a self-driving car; and other OEMs had similar initiatives (Oagana 2016). Daimler also did a lot of research in service robotics. In the 1990s, however, the computing platforms were not that powerful; and building a self-driving vehicle on an affordable budget was out of scope. Today, this has changed; and the cheap access to computing power in the range of gigaflops or even teraflops (Tanenbaum and Austin 2013) has sparked new interest in self-driving vehicles. The embedded computing power of even a smartphone is large enough for complex image processing and analysis. With an ever-increasing demand for low-cost mobility in passenger and freight transportation, autonomous vehicles are now at the center of many OEM and Tier 1 suppliers' research and development (R&D) initiatives.

Different steps toward full autonomous driving are required according to the European and US definitions (URL4 2015; URL27 2017). The first step defines classical driving without any interference from driver assistance systems; the next steps include assistance functions of various degrees of sophistication and complexity. In highly automated driving, the onboard computers can do most of the driving, which can be compared with the autopilot systems of an aircraft; however, the driver can still interfere. Finally, fully automated driving does not need any interference at all. It is clear, that although advanced driver assistance system already take over a lot of responsibilities, full autonomy is still some time away (URL5 2017).

The reasons for this are manifold:

- Cybersecurity issues: an autonomous car that is hacked could turn into a potential weapon.
- Ethical issues: if an accident is unavoidable, would one hurt a child or an elderly person?

- Handover from full autonomy back to driver interaction.
- Heterogeneous or mixed mode traffic with fully autonomous, semiautonomous, and classic human-driven cars.
- Integration of high-definition (HD) maps, onboard driving assistance (lane keeping), and infrastructure information, such as traffic signals, traffic lights, and others.
- General functional safety of autonomous driving.

Some OEMs have made bold announcements (Doll 2015; Lambert 2017; URL5 2017), while others have taken a more cautious stance (Beck 2016). The timeline for autonomous driving is the focus of a lively debate both in public as well as in scientific and industrial task forces. The impact will not only be technical but also social and ethical as the transport industry offers many jobs that are being challenged. Thus, the topic of autonomous driving is addressed in more detail in the upcoming chapters of this book, examining it from different perspectives, especially from a general cyber physical system point of view (Möller 2016) and from connectivity and cybersecurity perspectives.

2.2.6 Digitalization

Digitalization and the digital transformation of value chains has become a central topic for the economy (URL5 2017), and the automotive industry takes this very seriously (Gnirke 2016; URL1 2015). It is interesting to note that the automotive industry is no stranger to information technology (IT)-based innovations and the digital transformation of processes. The product development process, for example, is highly sophisticated and digital in nature. The computer has become an indispensable tool for design and analysis (Gusig and Kruse 2010; Sinha and Haas 2006; Grieb 2010), so much so that one refers to the activities of automotive engineering as virtual product creation (see Chap. 3). All relevant data today is digital, and computational models dominate the product development process. Automotive manufacturing is already very advanced, and most manufacturing processes are analyzed and optimized on the computer. No factory is built until everything, from manufacturing processes, tooling, logistics, and even ergonomics, has been simulated thoroughly (Grieb 2010; Bracht et al. 2011). For an in-depth coverage of the digital factory see Bracht et al. (2011) and also refer to Chap. 3.

The direct interaction with the customer, however, apart from using social media channels, has been less digital (Eckl-Dorna 2016a). Also, the aftersales market still has a huge potential for digital transformation. Maintenance costs can be estimated, the workflow for a repair determined, and spare parts can be ordered in real time—these are just a few examples of what is possible (URL1 2014; URL15 2016).

2.3 Automotive OEMs and Suppliers

The automotive industry is dominated by several large companies, also called original equipment manufacturers (OEMs), as shown in Figs. 2.16 and 2.17:

- General Motors (GM)
- Toyota
- Volkswagen (VW)
- Renault-Nissan
- Hyundai

The biggest automakers, VW and Toyota, are multibrand groups that have a global footprint and sell around 10 million units a year (URL11 2017). The premium passenger car sector is dominated by German companies, such as Daimler, BMW, Porsche, and Audi.

Commercial vehicles form another important segment of the global automotive market. Again, big international groups, such as Daimler, with its brands Actros, Freightliner, Fuso, Bharat-Benz, etc., and Volkswagen, with its brands, VW, Scania, and MAN, dominate the market. In terms of unit sales, however, the Chinese

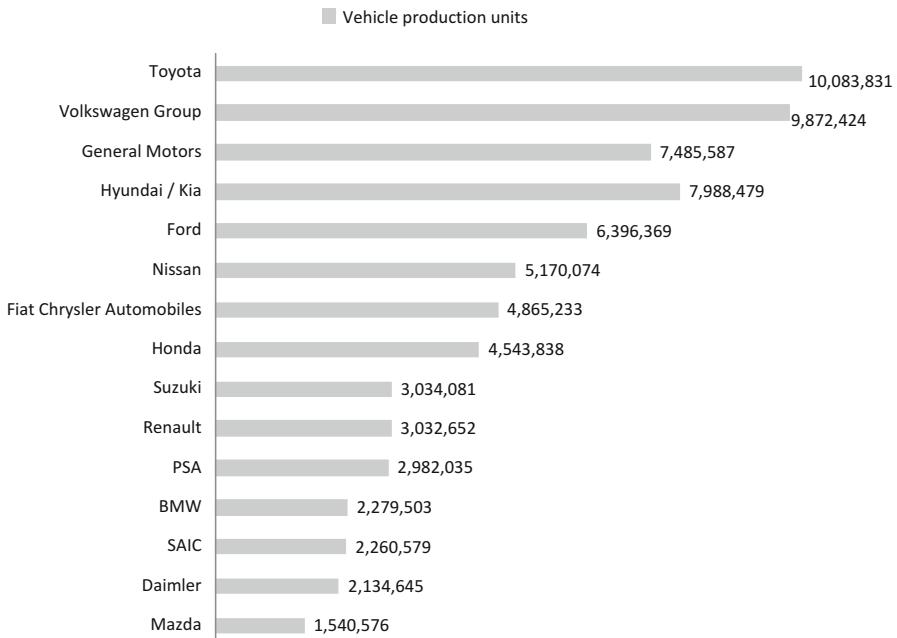


Fig. 2.16 The largest automotive OEMs (URL11 2017)

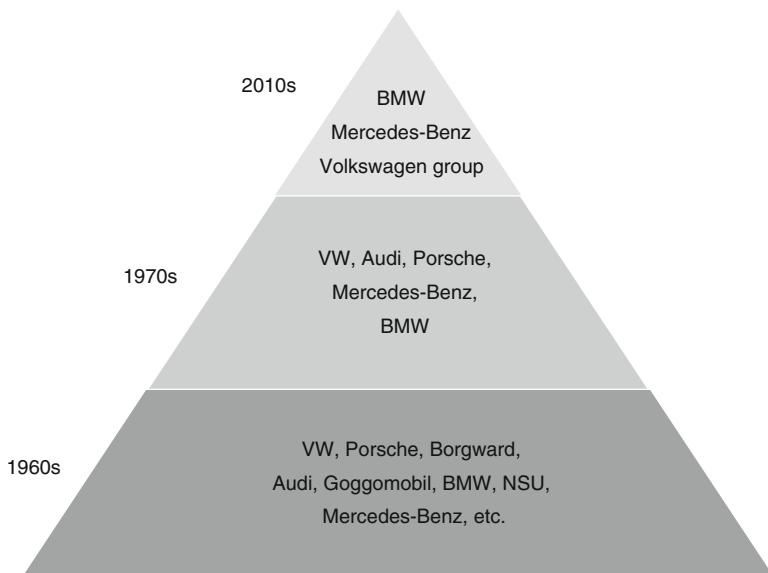


Fig. 2.17 Mergers and acquisitions in the German automotive industry from the 1960s until now (modified after (Dietz et al. 2016))

manufacturer, Dongfeng, is now the largest commercial vehicle manufacturer in the world (URL19 2017).

The supply chain is also dominated by big players, such as Bosch, Conti, Denso, ZF/TRW, Aptiv/Delphi, etc., also called first-tier suppliers. The biggest, Bosch and Continental, account for roughly 20% each of the total revenues in automotive E/E (URL20 2017); and mergers and acquisitions are still going on, as the recent merger between the German auto supplier, ZF Friedrichshafen AG, and the US TRW Automotive Holdings Corporation clearly shows.

This even had ripple effects on the semiconductor market, largely driven by market opportunities. Qualcomm, for example, planned to take over NXP which had already bought FreeScale in 2015 (URL3 2015). Figure 2.18 gives an overview of the largest players in the automotive E/E market (URL20 2017; Borgeest 2013) and shows their shares of the total market. Leading suppliers such as Bosch and Conti together account for 40% of the global market (URL20 2017).

As shown in Figs. 2.19 and 2.20, the concentration wave in the OEMs peaked around 1910, while the number of suppliers was highest in the middle 1970s (Dietz et al. 2016; Bopp 2016b).

Figure 2.17 illustrates the mergers and acquisition activities in the German automotive industry. In the 1960s, there were nearly 50 different manufacturers

Fig. 2.18 Automotive E/E suppliers by market share
(URL20 2017)

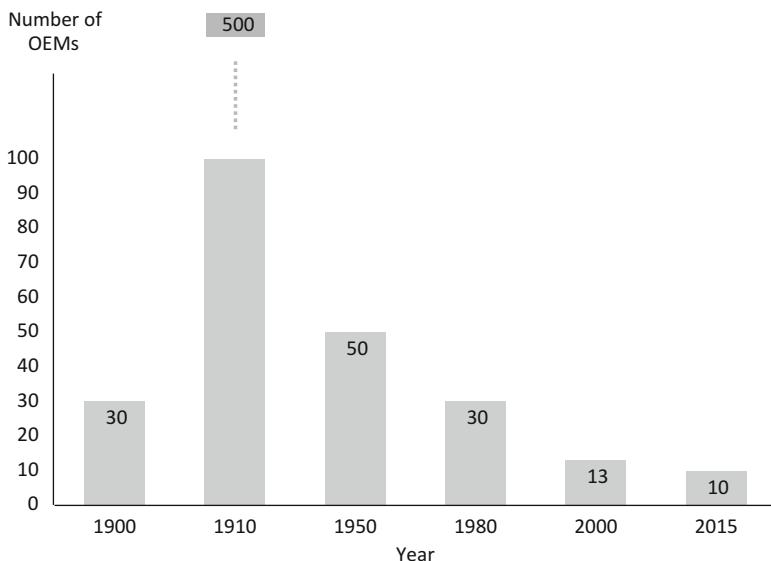
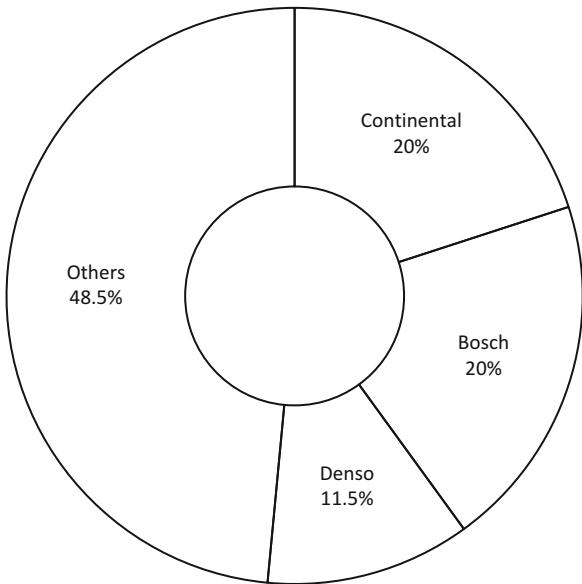


Fig. 2.19 Number of automotive OEMs from 1900s to today (source: Kalmbach 2004, modified after Bopp 2016b)

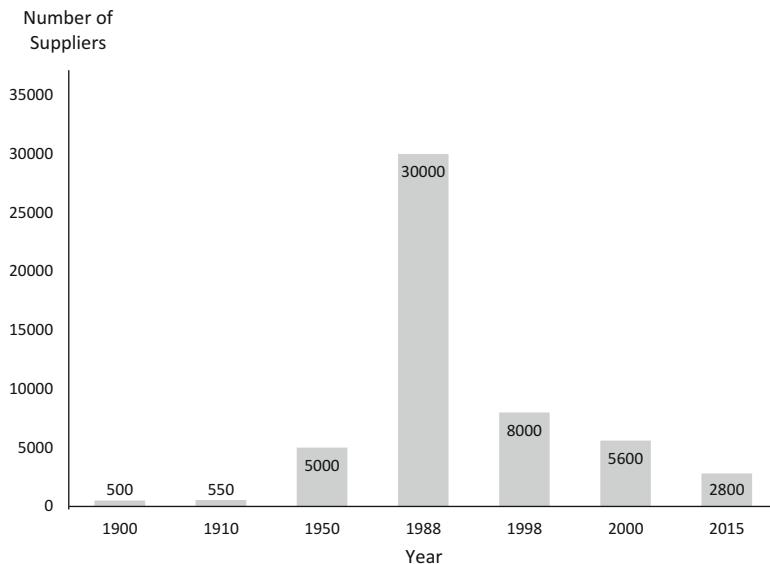


Fig. 2.20 Number of suppliers from the 1900s to today (source: Kalmbach 2004, modified after (Bopp 2016b))

active in Germany. This number was reduced to 4 in the 1990s and eventually came down to 3 today:

- Volkswagen group (with a total number of 12 brands, including Audi and Porsche)
- BMW
- Mercedes-Benz

2.4 New Players and Challenges

A 3-trillion Euro market, like the global automotive industry, attracts new players; the electric vehicle market is especially dynamic. Each year, new start-ups, many from China or funded by Chinese investors, are coming up, e.g., Byton, Faraday Future, Karma, BYD, and others (Sorge 2016). Other new players come from totally different industries (Kahnert 2016) and are attracted by the many hours drivers and passengers spend in a car every day. This is the perfect time to provide content, entertainment, and information, especially if driver assistance functions and autonomous driving will free up the driver in the future. Google, for example, has been experimenting with autonomous driving for years (URL22 2017; Burkert 2015).

Originally launched as a “Google X Moon Shot” project, the company’s autonomous driving activities are steered by the subsidiary, Waymo (URL21 2017; URL22 2017). Every day, Google cars generate thousands of test kilometers in California

and elsewhere. The goal is not to build Google car factories but to use the company's vast digital infrastructure, such as search capabilities, maps, the Android mobile operating system, speech-based assistance functions, and so forth (Doll 2015; Hecking 2016). Google has recently partnered with Fiat Chrysler Automobiles (FCA) to develop an autonomous minivan (Eckl-Dorna 2016b). They have agreed to work together to build a fleet of 100 self-driving minivans, marking the first time that a Silicon Valley firm has teamed up with a traditional automaker to develop an autonomous vehicle.

Also, the biggest IT and consumer electronics company, Apple, is looking at the car market with great interest (Eckl-Dorna 2016d). Apple launched a highly secretive project code named Project Titan headed up by Vice President Steve Zadesky, a veteran of Ford who helped to build the first iPod, to explore the feasibility of a highly autonomous electric Apple car fully integrated into the Apple ecosystem. Apple's Project Titan research facility was set up last year, away from Apple's main campus in Cupertino, California, for which Doug Betts, an automotive executive from FCA, was hired. Betts spent nearly 30 years in the industry and is an expert in manufacturing. Assuming Apple ultimately decides to bring the electric car to market, a commercial rollout will still be several years away. Currently, Apple is running a fleet of vehicles equipped with camera rigs. Following some speculation that this could be testing of a self-driving vehicle technology, the company revealed that the vehicles were collecting data to improve its Maps products. As such, the information captured will probably manifest as an Apple equivalent of Google's Street View product—but it's a sign the company is continuing to invest in technologies relevant to the automotive industry. Although, the recent news about Apple restructuring the project and laying off automotive experts (URL6 2016; Gurman and Webb 2016) has caused some amusement within the traditional automotive industry, the interest of the IT giants and their resilience should not be underestimated (Freitag and Rest 2016). It might be difficult to build a vehicle from scratch; but the suppliers are already responsible for a large part of the technology, and even factories can be rented/leased/subcontracted (Dietz et al. 2016). It is also important to note that the cost and part structure of an electric car is very different from one with an internal combustion engine (ICE) where the engine needs a lot of special manufacturing knowledge and accounts for a major chunk of the total cost and internal value added in the traditional automotive industry. In an electric vehicle, the battery is a major cost factor, while the cost of power electronics and software increases significantly (Kampker et al. 2013; Hinderer et al. 2016; Steinacker 2016). This shift in cost structure, value added, and supply chain for electrical components is an encouragement for new players as many of the established automotive companies lack experience in these domains.

2.5 The Digital Transformation of the Automotive Industry

The smart watch, shown in Fig. 2.21, is an example of a new human-machine interface (HMI) to the vehicle. It can be used as a key to open the vehicle door, to display status information about key physical parameters, or even as a remote control to switch on lights and the A/C, or to trigger an acoustic alarm (Eckl-Dorna 2016c).

Fig. 2.21 Smartwatch and smartphone get connected to the car



In a few years from now, nearly all vehicles will be connected and offering full access to the Internet (URL1 2014; URL6 2017; Viereckl et al. 2016). There is an evolving value chain around this connected vehicle ecosystem (Werle 2015). Although, it is not yet clear which players will benefit in the end and how many of the new business models will be adopted, the opportunities for more than a billion vehicles globally—most of them connected—inspires the pundits (URL8 2016; Viereckl et al. 2016). The possibilities seem endless. Refined diagnostic instruments and radio-telemetrics will enable mechanics to diagnose and partly solve problems, without the need to bring the vehicle to the garage. Integrated information systems, computerized motor management, and opto-electronic displays will enhance safety, performance, and comfort.

Toyota recently introduced a small robot that can be bought as a companion (Mullen 2016). The SoftBank-funded Pepper robot (URL7 2017) was one of the stars at the 2017 CeBIT fair and is shown in Fig. 2.22.

One way automotive OEMs reacted to this challenge of digitalization and widespread impact of IT on every aspect of our lives is their close cooperation with innovative start-ups. Daimler's initiative, called STARTUP AUTOBAHN, is shown in Fig. 2.23 (URL8 2017). STARTUP AUTOBAHN is a program jointly run with Plug and Play and modeled as an accelerator and hub for new companies. Plug and Play is a well-known accelerator/incubator company from Silicon Valley that hosts start-ups in an early phase (URL9 2017). The service includes office space, mentoring, consulting, and introductions to potential customers and venture capital firms.

Daimler collaborates with Plug and Play to work with start-ups on various innovation projects. The range is quite wide from production optimization to new ideas in HMI, cybersecurity, and big data. Earlier in 2017, the program was extended to include other OEMs such as Porsche, first-tier suppliers such as ZF, and IT companies such as Hewlett Packard Enterprise (HPE) under the umbrella of Arena 2036 (URL1 2017).

Fig. 2.22 SoftBank's Pepper robot



Fig. 2.23 STARTUP AUTOBAHN is Daimler's start-up accelerator/hub program



Fig. 2.24 A starship delivery robot in Tallinn

Another indicator of how much the megatrends are changing the automotive business is the general increase in R&D budgets. Daimler, for the first time in decades, for example, has increased its spending on R&D for the running year by a whopping 18% (URL10 2016). Recently, Daimler's van division invested in the robotics company Starship Technologies in Tallinn, which develops autonomous delivery robots, as shown in Fig. 2.24.

2.6 Exercises

What is meant by the term *global automotive market*?

Describe some key aspects of the global automotive market.

What was the *worldwide production of cars in 2012*?

Describe the production numbers w.r.t. to regions.

What shifts have occurred in the last decade regarding the different markets?

Describe the shifts w.r.t. to American, Asian, and European regions.

What is meant by *penetration of cars/the ratio of cars to inhabitants*?

Describe the differences between different countries.

What is meant by the term *aftermarket*?

Describe the aftermarket constraints.

What is meant by the term *Triade market*?

Give an example of the Triade market.

What is meant by the term *OEM*?

Give an example of some automotive OEMs.

What is meant by *concentration trends among OEMs over the last decades*?

Give an example of the concentration trend.

What is meant by the term *x-tier supplier*?

Give an example of some x-tier suppliers.

What is meant by the term *automotive megatrends*?

Give an example of some automotive megatrends.

What is meant by the term *electromobility*?

Give an example of the drivers of electromobility.

What is meant by the term *electric cars*?

Give an example of hurdles to more electric cars on the roads.

What *role does price of the battery play in electric cars*?

Give an example of battery prices w.r.t kWh in electric cars.

What *price level in batteries (in GW/h) is supposed to be a game changer*?

Give an example.

What is the *projected output of Tesla's Gigafactory concept*?

Describe Tesla's Gigafactory concept.

Who are the *corporate partners for the Gigafactory*?

Describe the partners involved.

What *charging concepts do you know*?

Describe the concepts in detail.

What are the *benefits of electric cars*?

Describe the benefits in detail.

What are the *disadvantages compared to combustion engines*?

Describe the disadvantages in detail.

What is meant by the term *autonomous driving*?

Describe the concept of autonomous driving.

What *mobility trends are changing the automotive industry*?

Describe the trends in detail.

What does the term *carsharing mean*?

Give an example.

What is meant by the term *digital transformation*?

Give an example.

What are the *effects of digital transformation on the automotive industry*?

Give an example.

What is meant by the term *connected car*?

Give an example.

What services does a connected car offer?

Give an example.

What are the *projections for connected cars in the future*?

Give an example.

What are the *benefits to automotive OEMs and suppliers from technology start-ups*?

Give an example.

What is a *start-up accelerator*?

Give an example.

What is meant by the term *Daimler's STARTUP AUTOBAHN*?

Describe the term in detail.

References and Further Reading

- (Beck 2016) Beck, T.: Do we need autonomous driving? (in German). ATZ elektronik, 1/2016
- (Bernhart 2016) Bernhart, W.: Israel – Under-estimated high-tech location (in German). ATZ elektronik, 4/2016
- (Bopp 2016a) Bopp, R.: Management of Automotive Value Creation. In: Dietz, W., Reindl, S., Bracht, H., editors. Basic Principles of the Automotive Business (in German), 6th edition (in German), Springer Automotive Media, 2016
- (Bopp 2016b) Bopp, R.: Manufacturer-supplier relationship: structures and perspectives. In: Dietz, W., Reindl, S., Bracht, H., editors. Basic Principles of the Automotive Business, 6th edition (in German), Springer Automotive Media, 2016
- (Borgeest 2013) Borgeest, K.: Electronics in vehicle technology – hardware, software, systems and project management (in German). Vieweg and Teubner Publ., 2013
- (Braun 2016) Braun, J.: The recipe for success at Tesla is called leadership (in German). ATZ elektronik, 1/2016
- (Bracht et al. 2011) Bracht, U., Geckler, D., Wenzel, S.: Digital Factory – Methods and Practical Examples (in German). Springer Publ., 2011
- (Burkert 2015) Burkert, A.: The big deal with the data (in German). ATZ elektronik, 4/2015
- (Burt 2016) Burt, M.: Volkswagen unveils Moia, its new mobility services brand. Autocar online. December 5th 2016. Available from: <https://www.autocar.co.uk/car-news/industry/volkswagen-unveils-moia-its-new-mobility-services-brand>
- (DAT 1995-2015) DAT report, 1995-2015: Ostfildern
- (Dietz et al. 2016) Dietz, W., Reindl, S.: Structure and importance of the automotive market in Germany (in German). In: Dietz, W., Reindl, S., Bracht, H., editors. Basic Principles of the Automotive Business (in German), 6th Edition. Springer Automotive Media, 2016
- (Doll 2015) Doll, N.: In five years from now the driverless car will already be there (in German), Welt online. March 2nd 2015. Available from: <http://www.welt.de/wirtschaft/article137958214/Schon-in-fuenf-Jahren-gibt-es-das-fahrerlose-Auto.html>
- (Doll and Fuest 2015) Doll, N., Fuest, B.: Why, in the future, my car can sneak on me. Welt online. February 11th 2015. Available from: <https://www.welt.de/wirtschaft/article137341236/Warum-mich-mein-Auto-kuenftig-verpetzen-kann.html>
- (Eckl-Dorna 2016a) Eckl-Dorna, W.: Pariser Autosalon 2016 - Where car manufacturers still have problems with the digitization (in German). Manager Magazin online. September 28th 2016. Available from: <https://www.manager-magazin.de/unternehmen/autoindustrie/pariser-autosalon-2016-digitalisierungautobranche-wo-es-hakt-a-1114318.html>
- (Eckl-Dorna 2016b) Eckl-Dorna, W.: Savior instead of aggressor: Fiat Chrysler courts Google (in German). April 29th 2016. Available from: <http://www.managermagazin.de/unternehmen/autoindustrie/roboterauto-allianz-warum-fiat-chrysler-mitgoogle-kooperieren-will-a-1090052.html>
- (Eckl-Dorna 2016c) Eckl-Dorna, W.: VW cooperates with LG – it's not the first networking deal (in German). July 6th 2016. Available from: <http://www.manager-magazin.de/magazin/artikel/vernetzte-autos-vw-kooperiert-mit-lg-und-nicht-mitapple-a-1101629.html>
- (Eckl-Dorna 2016d) Eckl-Dorna, W.: Apple speaks out, what all know, finally, Apple comments on self-driving cars (in German). December 5th, 2016. Manager Magazin online. Available from: <http://www.manager-magazin.de/unternehmen/auto-industrie/icar-apple-deutet-in-brief-arbeit-an-selbstfahrenden-autos-an-a-1124531.html>
- (Freitag 2016) Freitag, M.: Robotic Cars - German Manufacturers in Pole Position (in German). July 26th 2016. Available from: <https://www.manager-magazin.de/unternehmen/autoindustrie/roboterautos-deutsche-autobauer-fuehrena-1104783.html>
- (Freitag and Rest 2016) Freitag, M., Rest, J.: Alex Hitzinger moves to Silicon Valley - Why Porsche loses a top developer to Apple (in German). December 16th 2016. Available from: <http://www.manager-magazin.de/unternehmen/autoindustrie/porsche-top-entwickler-axel-hitzinger-wechselt-zu-apple-a-1126243.html>

- (Freitag 2017) Freitag, M.: Disruption takes time (in German). Manager Magazin online. April 24th 2017. Available from: <http://www.manager-magazin.de/unternehmen/autoindustrie/disruptiv-on-roland-berger-autoindustrie-im-umbruch-a-1144503.html>
- (Gates et al. 2015) Gates, G., Ewing, J., Russell, K., Watkins, D.: How Volkswagen's 'Defeat Device' worked. NY Times online. 2015, updated on March 16th 2017. Available from: <https://www.nytimes.com/interactive/2015/business/international/vw-diesel-emissions-scandal-explained.html>
- (Gerhager 2016) Gerhager, S.: Why auto makers might soon get into the focus of blackmailers (in German). Focus online. October 17th 2016. Available from: http://www.focus.de/auto/experten/autoindustrie-warum-autohersteller-fokus-von-erpressern-geraten-koennte_id_6081085.html
- (Gebhardt 2016) Gebhardt, M.: This is how we park tomorrow (in German). Zeit online. May 10th 2016. Available from: <https://www.zeit.de/mobilitaet/2016-04/autonomes-fahren-parken-bosch>
- (Gnirke 2016) Gnirke, K.: VW and Toyota against Google and Apple - reluctantly (in German). <http://www.spiegel.de/wirtschaft/unternehmen/volkswagen-und-co-zoegernd-in-den-kampf-mit-google-und-apple-a-1094147.html>
- (Greenberg 2013) Greenberg, A.: Hackers reveal nasty new car attacks-with me behind the wheel. Forbes online. July 24th 2013. Available from: <https://www.forbes.com/sites/andygreenberg/2013/07/24/hackers-reveal-nasty-new-car-attacks-with-me-behind-the-wheel-video/#5a9acdfe228c>
- (Grieb 2010) Grieb, P.: Digital Prototyping – Virtual product development in mechanical engineering (in German). Carl Hanser Publ., 2010
- (Grünweg 2016a) Grünweg, A.: Cyberattacks on cars - The enemy drives along with you (in German). <http://www.spiegel.de/auto/aktuell/cyberattacken-auf-autos-der-feind-faehrt-mit-a-1084059.html>
- (Grünweg 2016b) Grünweg, T.: Ford strategy – Autonomy for All (in German). Spiegel online. October 11th 2016. Available from: <http://www.spiegel.de/auto/aktuell/ford-plant-roboter-taxiflotte-wie-uber-a-1114025.html>
- (Gurman and Webb 2016) Gurman, M., Webb, A.: Bloomberg online. October 17th 2016. Available from: <https://www.bloomberg.com/news/articles/2016-10-17/how-apple-scaled-back-its-titanic-plan-to-take-on-detroit>
- (Gusig and Kruse 2010) Gusig, L. O., Kruse, A.: Vehicle development in the automotive industry – Current tools for practical use (in German). Carl Hanser Publ., 2010
- (Haas et al. 2017) Haas, R., Möller, D., Bansal, P., Ghosh, R., Bhat, S.: Intrusion Detection in Connected Cars. In: Proceed. IEEE/EIT 2017 Conference, pp. 516–519. Ed.: Izadian, A., Catalog No. CFP17EIT-USB 978-1-5090-4766-6, 2017
- (Hecker et al. 2012) Hecker, F., Hurth, J., Seeba, H.-G. (Eds): Aftersales in the automotive industry – concepts for their success (in German). Springer Automotive Media, 2012
- (Hecking 2016) Hecking, M.: Hardware battle between Google, Apple and Amazon -fight for the heads (in German). Manager Magazin online. October 5th 2016. Available from: <http://www.manager-magazin.de/unternehmen/it/hardwareschlacht-zwischen-google-apple-und-amazon-a-1115268.html>
- (Hinderer et al. 2016) Hinderer, H., Pflugfelder, T., Kehler, F. (Eds): Electromobility – Opportunities for suppliers and manufacturers (in German). Springer Automotive Media, 2016
- (Hirn 2016) Hirn, W.: Car Rental Services in China – Didi vs. Uber – The billion Dollar battles of the Chinese cousins (in German). Manager Magazin online. July 27th 2016. Available from: <http://www.manager-magazin.de/finanzen/artikel/a-1105011.html>
- (Johanning and Mildner 2015) Johanning, V., Mildner, R.: Car IT compact – The car of the Future – Driving Connected and Autonomously (in German). Springer Publ., 2015
- (Kahnert 2016) Kahnert, S.: Market is changing: Daimler, BMW and Audi are suddenly getting new adversaries (in German). Focus online. October 21st 2016. available online. https://www.focus.de/finanzen/news/oberklasse-autobauer-noch-unter-sich-markt-aendert-sich-daimler-bmw-und-audi-bekommen-ploetzlich-neue-gegner_id_6098424.html
- (Kalmbach 2004) Kalmbach, R.: FAST 2012: An industry is changing – Facts, Figures and Trends (in German). In Automobile Production, 04/2004

- (Kamper et al. 2013) Kamper, A., Vallee, D., Schnettler, S.: Electromobility – the basis of a future technology (in German). Springer and Vieweg Publ, 2013
- (Knieps 2016) Knieps, S.: Humans will always want to drive themselves - Daimler Board member Entemann about the future of the community car and how the autonomous car will change the business model of Car-2-go (in German). Bilanz Magazin. July 7th 2016
- (Lambert 2017) Lambert, F.: Elon Musk clarifies Tesla's plan for level 5 fully autonomous driving: 2 years away from sleeping in the car. elektrek. April 29th 2017. Available online. <https://electrek.co/2017/04/29/elon-musk-tesla-plan-level-5-full-autonomous-driving/>
- (Lobe 2016) Lobe, A.: Hacker Alert – In a modern car today are computers and info systems that are easy to manipulate. How do the manufacturers deal with the security gap? (in German). Zeit online. August 25th 2016. Available from: <http://www.zeit.de/2016/34/elektroautos-steuerung-hacker-gefahr-sicherheit-hersteller>
- (Markoff 2016) John Markoff, Artificial Intelligence Swarms Silicon Valley on Wings and Wheels, The New York Times online. July 17th 2016. Available from: <http://nyti.ms/2a0Awys>
- (Maurer et al. 2015) Maurer, M., Gerdes, J. C. Lenz, B., Winner, H. (Eds.) : Autonomous Driving – Technical, Legal and Social Aspects (in German). Springer Publ. 2015
- (Meyer and Shaheen 2017) Meyer, G., Shaheen, S. (Eds.): Disrupting Mobility – Impacts of Sharing Economy and Innovative Transportation on Cities. Springer Publ. 2017
- (Möller 2016) Möller, D. P. F.: Guide to Computing Fundamentals of Cyber-Physical Systems – Concepts, Design Methods, and Applications. Springer Publ. 2016
- (Möller et al. 2017) Möller, D. P. F., Haas, R., Akhilesh, K.B.: Automotive Electronics, IT, and Cybersecurity. In: Proceed. IEEE/EIT 2017 Conference, pp. 575-580. Ed.: Izadian, A., Catalog No. CFP17EIT-US. 978-1-5090-4766-6, IEEE, 2017
- (Mullen 2016) Mullen, J.: Toyota wants this baby robot to be your friend. CNN tech. October 4th 2016. Available from: <http://money.cnn.com/2016/10/03/technology/toyota-robot-kirobo-mini/index.html>
- (Oagana 2016) Oagana, A.: A short history of Mercedes-Benz Autonomous Driving Technology. autoevolution.com. January 25th 2016. Available from: <https://www.autoevolution.com/news/a-short-history-of-mercedes-benz-autonomous-driving-technology-68148.html>
- (Poulsen 2010) Poulsen, K.: Hacker disables more than 100 cars remotely. Wired online. March 17th 2010. Available from: www.wired.com/threatlevel/2010/03/hacker-bricks-cars
- (Reindl and Maier 2016) Reindl, S. Maier, B.: The automobile as the dominant means of transport (in German). In Dietz, W., Reindl, S., Bracht, H., Basic Principles of the Automotive Business (in German), 6th Edition, Springer Automotive Media, 2016
- (Rees 2016) Rees, J.: Mobility – Never have to park yourself (in German). Wiwo online. May 6th 2016. Available from: <https://www.wiwo.de/technologie/mobilitaet/mobilitaet-nie-mehr-selber-einparken-muessen/13529696.html>
- (Rungg 2016) Rungg, A.: Alphabet - Why Google has really renamed itself (in German). Manager Magazin online. May 9th 2016. Available from: <http://www.manager-magazin.de/magazin/artikel/alphabet-warum-google-sich-wirklich-umbenannt-hat-a-1088043.html>
- (Randak 2016) Randak, S.: BMW, Daimler and VW cornered by Apple? Tesla? The danger for German automakers is lurking somewhere else (in German). December 2016. <http://www.manager-magazin.de/unternehmen/artikel/autobauer-in-gefahr-zulieferer-haben-bessere-entwicklungskompetenz-a-1124068.html>. Accessed: May 20th 2018
- (Reindl et al. 2016) Reindl, S., Kluemper, M., Maier, B.: Mobility services in the automotive industry (in German). In: Dietz, W., Reindl, S., Bracht, H.: Basic Principles of the Automotive Business (in German), 6th Edition, Springer Automotive Media, 2016
- (Schmitt 2016a) Schmitt, B.: Germany's Bundesrat Resolves End Of Internal Combustion Engine. Forbes online. October 8th 2016. Available from: <https://www.forbes.com/sites/bertelschmitt/2016/10/08/germanys-bundesrat-resolves-end-of-internal-combustion-engine/>
- (Schmitt 2016b) Schmitt, B.: German Transport Minister: ICE Ban By 2030 “Utter Nonsense”. Forbes Online. October 11th 2016. Available from: <https://www.forbes.com/sites/bertelschmitt/2016/10/11/german-transport-minister-ice-ban-by-2030-utter-nonsense/#320d92c79668>

- (Schultz 2016) Schultz, M.: Billion loss at Uber – The evil of button-press capitalism (in German). Spiegel online. August 26th 2016. Available from: <http://www.spiegel.de/forum/wirtschaft/milliardenverlust-bei-uber-das-uebel-des-knopfdruck-kapitalismus-thread-505569-1.html>
- (Siebenpfeiffer 2014) Siebenpfeiffer, W. (Ed.): Networked Automobile – Safety, Car IT, Concepts (in German). Springer Publ. 2014
- (Sinha and Haas 2006) Sinha, K., Haas, R.: Architecture for integrated simulation driven design. Industrial Simulation Conference (isc 2006), Palermo, Italy, 2006
- (Sopha 2016) Sopha, W.: Challenge Globalization: the framework for a holistic strategy for automotive manufacturers (in German). In: Dietz, W., Reindl, S., Bracht, H., Basic Principles of the Automotive Business (in German). 6th edition, Springer Automotive Media, 2016
- (Sorge 2016) Sorge, N.-V.: Warren Buffett's Electric Car Chinese – BYD is attacking Daimler with its own factory in Europe (in German). Manager Magazin online. October 13th 2016. Available from: <http://www.manager-magazin.de/unternehmen/autoindustrie/byd-warren-buffetts-elektroauto-beteiligung-greift-an-a-1116320.html>
- (Spehr 2016) Spehr, M.: Internet connection in the Audi A4, Behind the steering wheel, Google shows the world (in German), FAZ online. August 18th 2016. Available from: <http://www.faz.net/aktuell/technik-motor/motor/kommunikationstechnik-des-audi-a4-im-test-14387527/der-audi-a4-kommt-mit-14390627.html>
- (Steinacker 2016) Steinacker, L.: Code capital – The software code becomes a crucial factor (in German). Wiwo online. September 11th 2016. Available from: <http://www.wiwo.de/my/technologie/digitale-welt/code-kapital-der-software-code-wird-zur-entscheidenden-groesse/14483036.html?ticket=ST-1890525-cMZzrlfHxLQkzDBkIbVd-ap3>
- (Tanenbaum and Austin 2013) Tanenbaum, A. S., Austin, T.: Structured Computer Organization. Pearson Education, 6th edition, 2013
- (Viereckl et al. 2016) Viereckl, R., Ahlemann, D., Koster, A., Hirsh, E., Kuhnert, F., Mohs, J., Fischer, M., Gerling, W., Gnanasekaran, K., Kusber, J., Stephan, J., Crusius, D., Kerstan, H., Warnke, T., Schulte, M., Seyfferth, J., Baker, E. H.: Connected car report 2016: Opportunities, risk, and turmoil on the road to autonomous vehicles. September 28th 2016. Available from: <https://www.strategyand.pwc.com/reports/connected-car-2016-study>
- (Werle 2015) Werle, K.: World in digital change – the game changer – BMW smartphone on wheels (in German). Manager Magazin. November 23rd 2015. Available from: <http://www.manager-magazin.de/unternehmen/artikel/game-changer-bmw-sieger-in-wettbewerb-von-bain-und-mm-a-1063812.html>

Links:

2013

- (URL1 2013) https://www.gsma.com/iot/wp-content/uploads/2013/06/cl_ma_forecast_06_13.pdf

2014

- (URL1 2014) <https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/whats-driving-the-connected-car>
- (URL2 2014) <http://www.oica.net/category/production-statistics/2014-statistics/>

2015

- (URL1 2015) <http://www.wiwo.de/unternehmen/auto/emobility/digitalisierung-der-autoindustrie-kuenftig-braucht-man-das-lenkrad-nicht-mehr/11602152.html>
- (URL2 2015) <http://www.pwc.com/us/en/technology/publications/assets/pwc-consumer-intelligence-series-the-sharing-economy.pdf>
- (URL3 2015) <https://www.cnbc.com/2015/12/07/nxp-closes-deal-to-buy-freescale-and-create-top-auto-chipmaker.html>
- (URL4 2015) <http://www.mckinsey.com/industries/automotive-and-assembly/our-insights/ten-ways-autonomous-driving-could-redefine-the-automotive-world>
- (URL5 2015) https://www.accenture.com/_acnmedia/Accenture/Conversion-Assets/DotCom/Documents/Global/PDF/Industries_18/Accenture-Automobilwoche-Supplement-2015-English.pdf
- (URL6 2015) <http://www.oica.net/category/vehicles-in-use/>

2016

- (URL1 2016) <http://www.businessinsider.com/2016-was-a-record-breaking-year-for-global-car-sales-and-it-was-almost-entirely-driven-by-china-2017-1?IR=T>
- (URL2 2016) VW shifts focus to electric cars with US expansion plan, <https://www.theguardian.com/environment/2016/nov/22/vw-shifts-focus-to-electric-cars-with-us-expansion-plan>
- (URL3 2016) Tesla enables autonomous driving in all cars (in German) <http://www.wiwo.de/unternehmen/industrie/elektroautopionier-tesla-ermoeglicht-autonomes-fahren-in-allen-autos/14713474.html>
- (URL4 2016) <https://www.tesla.com/blog/all-tesla-cars-being-produced-now-have-full-self-driving-hardware>
- (URL5 2016) Self-driving cars, Uber wants to chauffeur by autopilot (in German) <http://www.welt.de/wirtschaft/article157748150/Uber-will-Fahrgaeste-per-Autopilot-chauffieren.html>
- (URL6 2016) Is Apple not building his own car? (in German) <http://www.faz.net/aktuell/wirtschaft/netzwirtschaft/apple-steve-jobs/project-titan-baut-apple-doch-kein-apple-car-14485096.html>
- (URL7 2016) A great alliance for networked cars is emerging (in German). <http://www.faz.net/aktuell/wirtschaft/unternehmen/audi-bmw-co-eine-grosse-allianz-fuer-vernetzte-autos-entsteht-14455528.html>
- (URL8 2016) <https://www.morganstanley.com/ideas/car-of-future-is-autonomous-electric-shared-mobility>
- (URL9 2016) <https://www.daimler.com/innovation/specials/elektromobilitaet/case.html>
- (URL10 2016) <https://www.daimler.com/documents/investors/reports/annual-report/daimler/daimler-ir-annualreport-2016.pdf>
- (URL11 2016) <http://www.e-car-tech.de/?Lang=en>
- (URL12 2016) https://en.wikipedia.org/wiki/Volkswagen_emissions_scandal
- (URL13 2016) Daimler is transforming the Group for digitization (in German). <http://www.faz.net/aktuell/wirtschaft/f-a-z-exklusiv-daimler-baut-konzern-fuer-die-digitalisierung-um-14424858>
- (URL14 2016) Volkswagen - New mobility services have their own brand (in German). http://www.wiwo.de/unternehmen/auto/volkswagen-neue-mobilitaetsdienste-erhalten-eigene-marke/v_detail_tab_print/14616692.html
- (URL15 2016) <http://www.acea.be/press-releases/article/automotive-industry-joins-forces-on-access-to-vehicle-data>

2017

- (URL1 2017) <https://www.mckinsey.com/de/news/presse/autoindustrie-langfristiger-wachstumskurs-ist-intakt-aber-erlosquellen-verandern-sich> (in German)
- (URL2 2017) https://en.wikipedia.org/wiki/List_of_countries_by_motor_vehicle_production
- (URL3 2017) <http://www.gtai.de/GTAI/Navigation/DE/Invest/Industries/Mobility/automotive.html>
- (URL4 2017) <https://www.vda.de/en/press/press-releases/20170201-Global-commercial-vehicle-market-expands-in-2016.html>
- (URL5 2017) <https://www.tesla.com/autopilot>
- (URL6 2017) <http://www.huawei.com/en/news/2017/3/5GAA-EATA-sign-a-partnership-MoU>
- (URL7 2017) <http://www.cebit.de/speaker/pepper-the-robot/1322>
- (URL8 2017) <http://www.startup-autobahn.com/en/>
- (URL9 2017) <https://plugandplaytechcenter.com>
- (URL10 2017) <https://www.arena2036.de/>
- (URL11 2017) https://en.wikipedia.org/wiki/Automotive_industry
- (URL12 2017) <https://de.statista.com/statistik/daten/studie/164769/umfrage/groesste-automaerkte-weltweit- weltweit-nach-pkw-neuzulassungen/>
- (URL13 2017) https://en.wikipedia.org/wiki/List_of_countries_by_motor_vehicle_production
- (URL14 2017) <http://www.siamindia.com/statistics.aspx?mpgid=8&pgidtrail=13>
- (URL15 2017) <http://www.makeinindia.com/sector/automobiles>
- (URL16 2017) https://www.sci.de/fileadmin/user_upload/presse/pdf_downloads/151013_Weltweiter_Busmarkt.pdf
- (URL17 2017) <https://www.statista.com/statistics/270293/worldwide-leading-truck-manufacturers-based-on-production-figures/>
- (URL18 2017) <http://focus2move.com/world-car-market/>
- (URL19 2017) https://en.wikipedia.org/wiki/Dongfeng_Motor
- (URL20 2017) <http://www.eenewseurope.com/news/bosch-conti-dominate-automotive-electronics-market>
- (URL21 2017) <https://en.wikipedia.org/wiki/Waymo>
- (URL22 2017) <https://waymo.com>
- (URL23 2017) www.airbnb.com
- (URL24 2017) www.blackhat.com
- (URL25 2017) www.defcon.com
- (URL26 2017) <http://www.eurekanetwork.org/project/id/45>
- (URL27 2017) https://en.wikipedia.org/wiki/Autonomous_car
- (URL28 2017) https://en.wikipedia.org/wiki/Starship_Technologies
- (URL29 2017) Waymo boss reveals/Google cars avoid the internet for fear of hackers (in German).
<http://www.manager-magazin.de/unternehmen/it/google-roboterwagen-fahren-aus-angst-vor-hackern-ohne-internet-a-1129346-druck.html>
- (URL30 2017) https://www.destatis.de/EN/FactsFigures/EconomicSectors/TransportTraffic/Traffic_Accidents/TrafficAccidents.html
- (URL31 2017) <http://www.acea.be/statistics/tag/category/vehicles-in-use>

2018

- (URL1 2018) <https://www.bosch-si.com/de/mobilitaet/elektromobilitaet/charging-apps/charging-apps.html>



This chapter gives an overview of the research and development approach in the automotive industry. Therefore, Sect. 3.1 focuses on the automotive development process, specifically the complexity involved in developing a new vehicle model, an elaborate process involving thousands of engineering staff employed by the automaker and its Tier 1 suppliers. This section describes proven processes and new technologies, such as the Stage-Gate®controlled development process, the digital mock-up process, requirements engineering with regard to automotive electrical/electronic (E/E) systems (see Chap. 4), and the diverse disciplines that enable new product creation processes which lead to constantly shrinking development times, better maturity, and overall product quality. The focus in Sect. 3.2 is on modularization and platforms used in the automotive industry which will allow to cope with an ever-increasing multibrand vehicle model line. In Sect. 3.3, virtual product creation is introduced which integrates the product data management concept to fully achieve the required computer-aided design (CAD)-based development pipeline as an integrated CAD format. Section 3.4 introduces the idea of product life cycle management, an approach that facilitates collaborative work processes for the various phases of the product or system life cycle represented by a number of phases and activities spread out across the automakers organization and its suppliers, each of which builds on the results of the preceding phase or activity. The sum of all these activities is called the product or system life cycle, which can be described using a model that contains the conceptualization phase, the utilization phase, the evolution phase, and the ultimate disposal phase. Section 3.5 contains a comprehensive set of questions on automotive research and development, while the final section includes references and suggestions for further reading.

3.1 The Automotive Development Process

The automotive development process ranges from the first vehicle idea to the final automotive prototype with the following main objectives:

- Develop the vehicle to match or exceed customer requirements
- Develop the vehicle as soon as possible
- Utilize the invested resources as efficiently as possible

These objectives are related to the value creation of the automotive development process as they are linked to timing and size of the resulting invested cash flows. The objectives are neither independent nor necessarily fixed. Rather, they are influenced by the development process employed because the automotive development process is mostly portrayed as linear due to the intrinsic sequential structure of decision-making. Let us assume the decision(s) available at time t influences the set of decisions taken at time $t + 1$.

With regard to the specific course of actions included in the development process, various authors list somewhat different generic product development processes but all share the same basic structure. Differences can be found in the level of detail (LOD) covered (Sørensen 2006). The development of a new vehicle model is an elaborate process that can include thousands of employees at the automaker and its Tier 1 suppliers. Over the years, a trend has emerged of original equipment manufacturers (OEMs) doing less work internally, while Tier 1 suppliers and their supply chains provide more. This can be extended to the model that suppliers deliver complete systems as done, for example, for the Smart vehicle production in Hambach, France, established in 1994 as a joint venture of Daimler-Benz and the Swiss watch manufacturer Swatch to produce the Smart car. Today, it is fully owned by Daimler AG. Smart has the lowest in-house production depth in the automotive sector. It accounts for only 10%, while most other automakers often still have 20–40%. This means that 90% of the production steps to complete the vehicle are performed by component suppliers.

When a new vehicle is being developed, automakers typically work on an extensive benchmark with competitors, spending a significant amount of time on market research and customer analysis (Gusig and Kruse 2010) including the scope of a correct systematic approach of value engineering and target costing in cost management. Value engineering and target costing are complementary processes. One allows the identification of where in the development cost reduction could be achieved; the other shows the target to be achieved to guarantee the long-term profitability of a company. Based on value engineering methodology, work plans can be developed, taking into account the three subsequent stages:

- *Conceptualization Phase:* Occurs in the initial design activity when the scope of the project is drafted and a list of the desired design features and requirements is created.
- *Project Phase:* Determine the work to be performed, the budget, time schedule, what resources are needed, and assignment of responsibilities. As work

progresses, the status of the project is compared to the actual plan and schedule. During this phase, schedules may need to be adjusted to keep the project on track.

- *Validation Phase:* After completion, the project validation is done, highlighting project success and the lessons learned from the project implementation.

Therefore, the value engineering methodology focuses on product cost, functionality, and quality in accordance with customer needs and the company's product strategy. Hence, key drivers for the product strategy in the automotive domain will be:

- Advanced driver assistance system (ADAS) features (see Chap. 11)
- Interior design
- Passenger capacity
- Powertrain technology
- Production plant
- Research and development budget
- Safety features
- Special equipment and bundling strategy
- Target price level
- Telematics/Connectivity
- Type of vehicle model

Thus, the complexity of a vehicle's research and development (R&D) process is tremendous. One way to control quality and maturity of the developed components is to introduce a Stage-Gate-controlled process (Gusig and Kruse 2010; Sendler and Wawer 2011; Eigner and Stelzer 2013), as shown in Fig. 3.4.

The Stage-Gate process consists of the following stages:

- *Stage 1—Scoping:* A quick and inexpensive assessment of the technical merits of the project and its market prospects.
- *Stage 2—Building a Business Case:* This is the critical homework stage, the one that makes or breaks the project. Technical research, marketing, and feasibility are assessed resulting in a business case which has three main components:
 - Product and project definition
 - Project justification
 - Project plan
- *Stage 3—Development:* Here, plans are translated into concrete deliverables. The actual design and development of the new product takes place, the manufacturing or operations plan is mapped out, the marketing launch and operating plans are developed, and the test plans for the next stage are defined.
- *Stage 4—Testing and Validation:* Thus the purpose of this stage is to provide validation of the entire project: the product itself, the production/manufacturing process, customer acceptance, and the economics of the project.

After Stage 4 has been successfully completed, full production and commercial market launch of the product begins. The structure of each stage is similar and can be expressed as follows:

- *Activities*: Work the project leader and the team must undertake, based upon the project plan.
- *Integrated Analysis*: Project leader and team's integrated analysis of the results of all of the functional activities, derived through cross-functional interaction.
- *Deliverables*: Presentation of the results of the integrated analysis which must be completed by the team for submission to the gate.

Preceding each stage is a decision point, or gate, which serves as a go/kill and prioritization decision point. Gates are where mediocre projects are culled out and resources are allocated to the best projects. Gates deal with three quality issues: quality of execution, business rationale, and the quality of the action plan. The structure of each gate is similar:

- *Deliverables*: Inputs into the gate review: what the project leader and team will deliver to the meeting. Deliverables are defined in advance and are the result of actions from the preceding stage. A standard menu of deliverables is specified for each gate.
- *Criteria*: What the project is judged against in order to make the go/kill and prioritization decisions. These criteria are usually organized into a scorecard and include both financial and qualitative criteria.
- *Outputs*: Results of the gate review. Gates must have clearly articulated outputs including a decision (go/kill/hold/recycle) and a path forward (approved project plan, dates, and deliverables for the next gate agreed upon).

Thus the Stage-Gate-controlled development process results in a more effective, efficient, faster process that improves product innovation results (Cooper 2017).

Many companies have introduced product innovation processes; however, they are still struggling to achieve the financial results they expect. Thus focusing on innovation productivity, the question to be answered is, "How can the company identify waste, streamline the product development process, remove bureaucracy, and improve profits?" The answer is a lean, rapid, and profitable new product development process (Cooper and Edgett 2005). The need for a lean, rapid, and profitable new product development process has never been greater than today because product life cycles have become shorter, competition is more intense, and customers are more ambitious. Therefore, leading automakers have overhauled their product innovation processes, incorporating critical success factors discovered through best practices research, which is possible with the help of the Stage-Gate new product development process. The Stage-Gate process has become an industry standard for managing new product innovations. It integrates numerous performance driving practices into an easy-to-understand, successful approach. Its robust design engages users at all decision levels and functions, enabling quality execution, timely

go/kill decisions, alignment, and speed. This results in superior products reaching the market faster and generating more profit (URL1 2017).

The Stage-Gate-controlled process was introduced in the 1990s as a straightforward basic concept. The development project is broken down into well-defined phases and the maturity of the product is being checked at each stage. At the end of a particular phase, the results are checked against the specifications and pass if all quality criteria are satisfied. Since delays are problematic, a development project is typically managed very tightly. Tough competition, enormous cost pressure, and stringent time-to-market deadlines have a great impact on today's engineering processes (Gusig and Kruse 2010; Seiffert and Rainer 2008; Haas and Sinha 2004). A huge advantage in efficiency is realized through the deployment of specific tools such as:

- **CAD Systems for Geometric Design:** Computer-based design tasks commence with the use of CAD systems to generate detailed geometric models. A central task in dealing with geometric constraints for CAD is the generation of an optimal decomposition plan that not only supports efficient solutions but also captures design intent and assists with conceptual designs. With the assimilation of CAD systems and analysis tools into major industrial processes, an integrated approach is worthwhile. However, the need for a systematic way of considering the relationship between geometry and functional aspects of the geometric model becomes a preferential treatment, as shown for the automotive CAD examples in Figs. 3.1 and 3.2.

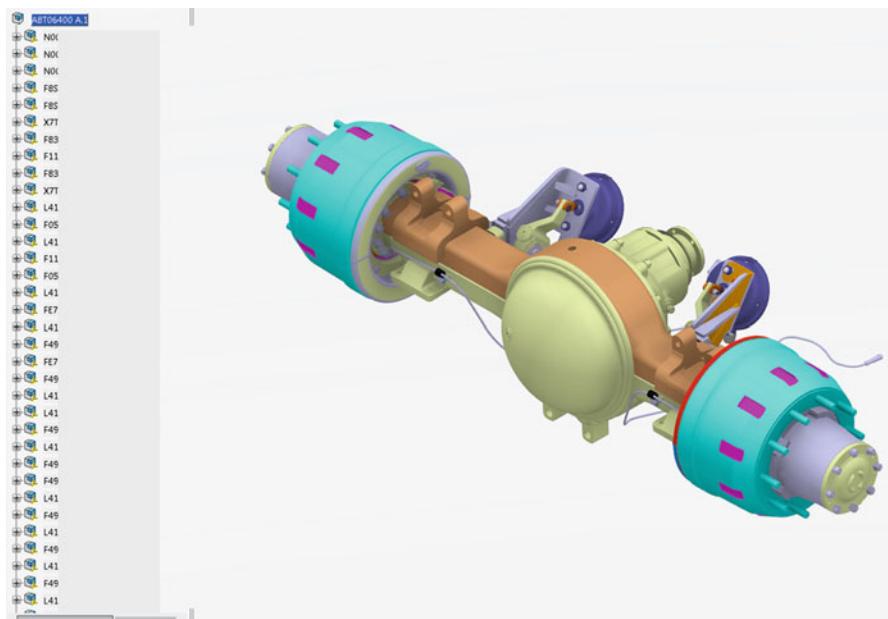


Fig. 3.1 Geometry design with the CAD system CATIA. Example axle (With kind approval/courtesy from Ashok Leyland ([URL9 2017](#)))

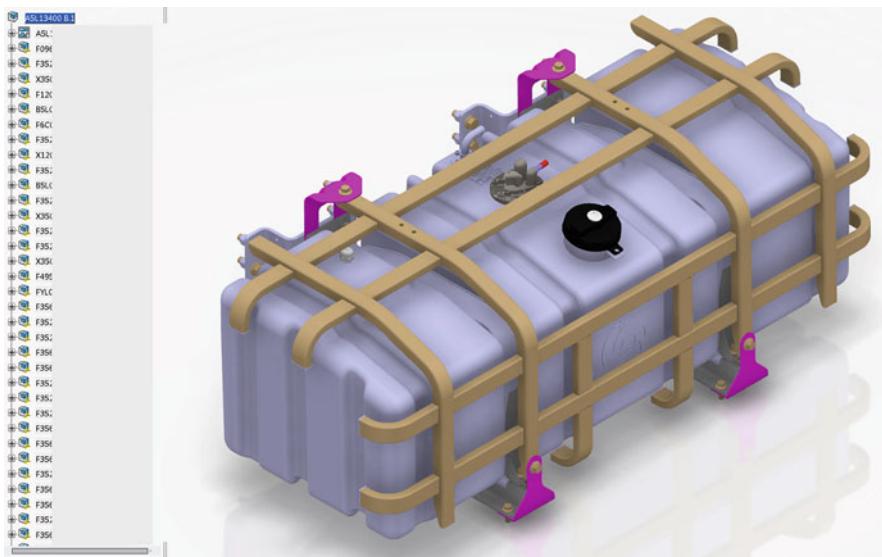


Fig. 3.2 Geometry design with the CAD system CATIA. Example fuel tank (With kind approval/courtesy from Ashok Leyland ([URL9 2017](#)))

In geometric design, the geometry of complex shapes is represented in terms of polynomial functions. The nature of the surface obtained using such polynomial-based methods usually depends on the type of polynomial chosen. Examples of such surfaces are:

- **Bézier Surfaces:** Many algorithms for curves in Bezier representation can be understood and derived using polynomials. A Bézier surface is defined by a set of control points. Similar to interpolation, a key difference is that the surface does not pass through the central control points; rather, it is stretched toward them as though each were an attractive force. They are visually intuitive and are used for many applications including CAD, computer graphics, and finite element modeling (FEM) (Bézier 1986). A dimension count shows that the $n + 1$ linearly independent Bernstein polynomial B_i^n is the basis for all polynomials of degree $\leq n$. Therefore, every polynomial curve $b(u)$ of degree $\leq n$ has a unique n th degree Bézier representation

$$b(u) = \sum_{i=0}^n c_i B_i^n(u)$$

Since the Bernstein polynomial represents a basis, every polynomial surface $b(x)$ has a unique Bézier representation

$$b(x) = \sum_i b_i B_i^n(u)$$

with respect to the reference simplex A . The coefficient b_i is called the Bézier point of b . They are the vertices of the Bézier net of $b(x)$ over the simplex A (Prautzsch et al. 2002).

- *B-spline Surface:* With regard to the Bézier representation of polynomial curves, it is desirable to write a spline $s(u)$ as an affine combination of some control points c_i as follows:

$$s(u) = \sum c_i N_i^n(u)$$

where the $N_i^n(u)$ are basic spline functions with minimal support and certain continuity properties. Schoenberg introduced the name B-splines for those functions (Schoenberg 1967).

The B-spline can be defined by the recursion formula

$$N_i^0(u) = \begin{cases} 1 & \text{if } u_i \in [a_i, a_{i+1}] \\ 0 & \text{otherwise} \end{cases}$$

and

$$N_i^n(u) = a_i^{n-1} N_i^{n-1}(u) + (1 - a_{i+1}^{n-1}) N_{i+1}^{n-1}(u),$$

where

$$a_i^{n-1} \frac{(u - a_i)}{(a_{i+n} - a_i)}$$

is the local parameter with respect to the support of N_1^{n-1} (Prautzsch et al. 2002).

B-spline surface permits the use of more control points in the characteristic polyhedron while retaining low-order basis functions. B-spline basis functions are nonzero only over a given finite interval and enable the effect of a control point on the surface shape to be localized. Another advantage of the B-spline formulation is its ability to preserve arbitrarily high degrees of continuity over the complex surface patch. These characteristics make the B-spline surfaces popular for use in an interactive modeling environment (Woodward 1987).

- *Rational B-splines:* Standard for surface modeling in CAD and computer graphics. Any typical surface forms, such as flat planes and quadratic surfaces, e.g., cylinders, spheres, ellipsoids of revolution, as well as more complex fully sculptured surfaces, are easily and accurately represented by rational B-spline surfaces. As with rational curves, rational forms of Bézier surfaces are possible. A Cartesian product rational B-spline surface in four-dimensional homogeneous coordinate space is as follows:

$$Q(u, w) = \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} B_{i,j}^h N_{i,k}(u) M_{j,l}(w)$$

where the $N_{i,j}^h$ s are the four-dimensional homogeneous polygonal control vertices, and $N_{i,k}(u)$ and $M_{j,l}(w)$ are the nonrational B-spline basis functions. An algorithm for a simple rational B-spline surface is given in Rogers (2001) as follows:

```

Specify number of control vertices in the u, w directions
Specify order in each of the u, w directions
Specify number of isoparametric lines in each of the u,
    w directions
Specify (or acquire) the control net
    and store in an array
Calculate (or acquire) the knot vector in the u direction
    and store in an array
Calculate (or acquire) the knot vector in the w direction
    and store in an array
For each parameter value, u
    Calculate the basis functions,  $N_{i,k}(u)$ 
        and store in an array
For each parameter value, w
    Calculate the basis functions,  $M_{j,l}(w)$ 
        and store in an array
    Calculate the SUM function.
        For each control vertex in the u direction
            For each control vertex in the w direction
                Calculate the surface point,  $Q(u, w)$ 
                    and store in an array
        end loop
    end loop
end loop
end loop

```

A pseudocode implementation of the algorithm is given in (Rogers 2001).

- *Nonuniform Rational B-splines*: Nonuniform rational B-splines are commonly referred to as NURBS. They have become the de facto industry standard for the representation, design, and data exchange of geometric information processed by computers. NURBS provide a unified mathematical basis for representing analytic shapes, such as conic sections and quadratic surfaces, as well as free-form entities, such as car bodies and ship hulls (Körber and Möller 2003). NURBS are

generalizations of nonrational B-splines and rational and nonrational Bézier curves and surfaces (Piegl and Tiller 1997).

A rational B-spline curve is the projection of a nonrational polynomial B-spline curve defined in four-dimensional homogeneous coordinate space back into the three-dimensional physical space which results in

$$P(t) = \sum_{i=1}^{n+1} B_i^h N_{i,k}(t)$$

where the B_i^h are the four-dimensional homogeneous control polygon vertices for the nonrational four-dimensional B-spline curve, and $N_{i,k}(t)$ is the nonrational B-spline basis function.

Vector-valued polynomials with convenient properties are valuable for complex modeling and simulation purposes. They are mainly utilized for industrial developments such as automotive systems, avionic systems, and others. A NURBS surface $S(u,v)$ can be defined as follows:

$$S(u, v) = \frac{\sum_{i=0}^m \sum_{j=0}^n P_{i,j} w_{i,j} N_{i,p}(u) N_{j,q}(v)}{\sum_{i=0}^m \sum_{j=0}^n w_{i,j} N_{i,p}(u) N_{j,q}(v)}$$

$$0 \leq u, v \leq 1$$

with the control point matrix

$$P_{i,j} = \begin{pmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,n} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{m,1} & p_{m,2} & \cdots & p_{m,n} \end{pmatrix}$$

the basic functions

$$N_{i,0}(u) = \begin{cases} 1 & \text{if } u_i \leq u \leq u_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u)$$

and the knot vector $U = \{u_0, \dots, u_m\}$, $u_i \leq u_{i+1}$.

The surface $S(u,v)$ has $(m + 1) \times (n + 1)$ control points $P_{i,j}$ and weights $w_{i,j}$. Assuming the degrees of basis functions along the u and v axes to be $p-1$ and $q-1$, respectively, the number of knots is $(m + p + 1) \times (n + q + 1)$. The nondecreasing knot sequence is $t_0 \leq t_1 \leq \dots \leq t_{m+p}$ along the u direction and $s_0 \leq s_1 \leq \dots \leq s_{n+q}$ along the v direction with the parameter domain in the range: $t_{p-1} \leq u \leq t_{m+1}$ and $s_{q-1} \leq v \leq s_{n+1}$. If the knots have multiplicity p and q in the u and v directions, respectively, the surface computation will interpolate the four corners of the boundary control points.

In (Piegl and Tiller 1997), an algorithm for relatively fast computation of a vertex on a NURBS surface is discussed, exploiting redundancies and the property that most basis functions, the $N_{i,p}(u)$ and $N_{j,q}(v)$, evaluate to zero for given u and v .

The key to rendering complex objects lies in computing and rendering only those parts of the object which are visible to the viewer. This, in particular, is challenging in the case of complex surface rendering because it's impossible to perform any kind of space partitioning and utilize one of the traditional occlusion culling methods. The reason for this property is that for a given point of view, the whole surface could possibly be overlooked. Moreover, omitting only those parts of the object which don't lie in the viewing area of the viewer doesn't suffice. The remaining triangle count could be far too high.

The key to surface rendering lies in the level of detail (LOD), i.e., rendering those parts of the surface which are far away from the viewer or which are rather smooth with less detail, i.e., less triangles and those parts which are close to the viewer or which are rough with more detail, which works for large objects only as small objects are always perceived in full detail (Piegl and Tiller 1997).

- *Digital Mock-Up (DMU)*: With the increasing complexity of technical systems, the design-related analyses have been successful based on the concept of DMUs. A DMU allows the aggregation of 3D CAD models of individual components into a 3D model of the product for its entire life cycle. This aggregation takes place according to the product or product structure and the position and orientation of each component within the product space enriched by all activities that contribute to describing the product. Product design engineers, manufacturing engineers, and support engineers work together to create and manage the DMU. The product structure is derived from the parts list or from the product data management (PDM). On the basis of the DMU, various analyses, such as collision of components, mountability, disassembly, etc., can be carried out. The integration of further aspects in addition to the shape, such as the movement behavior, the strength, or the thermal behavior, leads to virtual prototyping. This is a computer-based representation of the complete product and important knowledge of the future of the supported product to replace any physical prototype with virtual ones using 3D computer graphic techniques. It is also frequently referred to as digital prototyping (DP) or virtual prototyping (VP). These two specific definitions refer to the production of a physical prototype, but they are part of the

DMU concept. DMU allows engineers to design and configure complex products and validate their designs without ever needing to build a physical model. Various types of analyses can be carried out to demonstrate functional capability. Thus, the term VP has also been used for this purpose: partial models which describe individual aspects and are created and analyzed by the product under development. This saves time and money because the construction and testing of real prototypes can largely be eliminated. A pioneer in using DMUs is the aviation industry. In the early 1990s, Boeing was able to advance the complete development of its 777 airplane in this way and significantly reduce its development time. Some other typical DMU applications are (Gausemeier and Plass 2013):

- **Building Space and Assembly Analyses:** Building space analyses pursue the goal of optimally utilizing the available space. With the aid of the digital mock-up, it is possible to examine whether a container for the cooling liquid can be installed in the engine compartment. Furthermore, animations can be used to create very intuitive training documents for assembly and disassembly.
 - **Collision Investigations:** These are conducted in computer-assisted analysis of collisions, especially in complex kinematic systems with volume-bearing objects. Investigations of such analyses are contact (contact), component overlap (clash), and clearance violations (clearance). A DMU of a gearbox in Dassault’s CATIA environment which can be used for collision analysis of the moving parts is shown in Fig. 3.3.

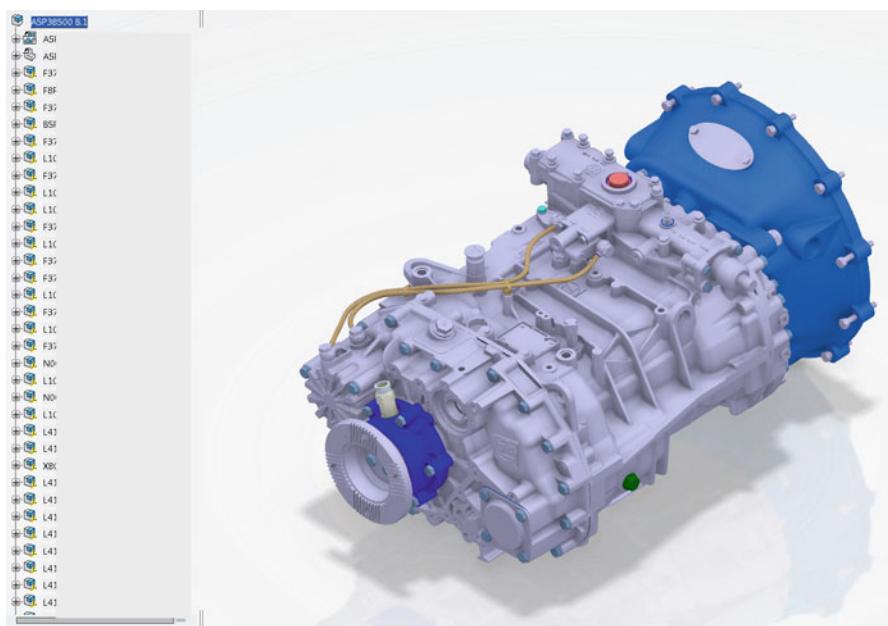


Fig. 3.3 Digital mock-up of a gearbox in CATIA (With kind approval from Ashok Leyland (URL 2017))

- *Creation of Sales Documents:* Today's efforts are driven by marketing and sales to produce realistic product presentations even though the product is not yet available. The basis for this is 3D CAD models. Their properties are assigned to properties such as color, texture, reflection, etc. Ray-tracing software puts the later product into the right light and delivers realistic-looking product images. In addition, the static 3D CAD models are animated in order to demonstrate their characteristic features in a realistic product animation.
- *Computer-Aided Engineering (CAE):* This term refers to the analysis of an engineering artefact. Computer-aided engineering includes finite element analysis (FEA), computational fluid dynamics (CFD), durability, optimization, and process analysis for manufacturing. A lot of these tasks are highly automated and deploy a wide variety of knowledge-based engineering methodologies, e.g. it is possible to automatically generate a FEA mesh from the CAD model.

These technologies are enabling new product creation processes, which lead to constantly shrinking development times, better product maturity, and overall product quality. In this context we have:

- *Concurrent Engineering (CE):* Concurrent engineering is a method of designing and developing products in which design teams attempt to integrate the different stages of design and development to run them simultaneously rather than consecutively. Concurrent engineering also refers to simultaneous or parallel engineering. In considering tooling, assembly, and routing during product design, concurrent engineering results in decreasing product development time and time to market, leading to improved productivity and reduced costs. It relies on the introduction of new process models, with appropriate team and organizational structure, as well as the deployment of new information technologies to manage the overall complexity of information and knowledge flows.
- *Simultaneous Engineering (SE):* Simultaneous engineering is a method for shortening product development by means of parallel initiation of the necessary development work. Thus, simultaneous engineering refers to the joint parallel development of a product and the necessary production plant. The overlapping of these two steps shortens the entire time until the market launch of the new product idea. The better information available to all parties involved in simultaneous engineering enables an accelerated and at the same time production-appropriate product development. Simultaneous engineering is often referred to as concurrent engineering. This includes the targeted and interdisciplinary collaborative and parallel work of the parties involved. OEM and Tier 1 suppliers can also be involved within the simultaneous engineering process in order to speed up and optimize the overall development. In this case data security and/or cybersecurity are of paramount importance.

These methods describe the product creation process which is highly parallel in nature and spans the complete product life cycle, from concept and feasibility studies, through manufacturing and market introduction, to disposal and recycling. Manufacturing and product face lifts, as well as recycling aspects, have to be taken into account at a very

early stage. The goal significantly minimizes the number of design changes based on a thorough understanding of the overall product creation process, stringent process plans with suitable control mechanisms, and quality gates, as shown in Fig. 3.4.

A flood of information needs to be managed through concept ideas, geometries, design documents, guidelines, cost information, project management data, 3D CAD modeling, DMU, CAE tools, and new visualization techniques, such as:

- **Virtual Reality (VR):** Like most technologies, VR did not suddenly appear. The emergence of VR was closely related to the maturity of other technologies, such as real-time computer graphics to visualize highly realistic 3D models. The required VR technology embraces all of the hardware utilized by the user to support a VR task. Typically this includes a head-mounted display (HMD); head-coupled display (HCD); 3D interactive devices, such as 3D mice and data gloves; headphones; and 3D trackers. 3D trackers are typically used to monitor the positions and orientation of objects in 3D space. Typically, a stationary transmitter radiates electromagnetic signals that are intercepted by a mobile detector attached to the user's head. When these signals are received by the detector, they are decoded to reveal the relative position and orientation between the transmitter and receiver. These signals are then passed onto the run-time system for transmission to the 3D graphic environment. The same principle is used for tracking the 3D mouse, whose signals are used to control an icon in the user's field of view. An HMD isolates the user from the real world and substitutes binocular views on the virtual environment (VE). The data glove can monitor the status of the user's fingers. This is achieved through the use of fiber optics attached to the back of the glove fingers. When the user's fingers are flexed, the optical characteristics of the fiber optics alter, this can be measured and scaled into an output signal.

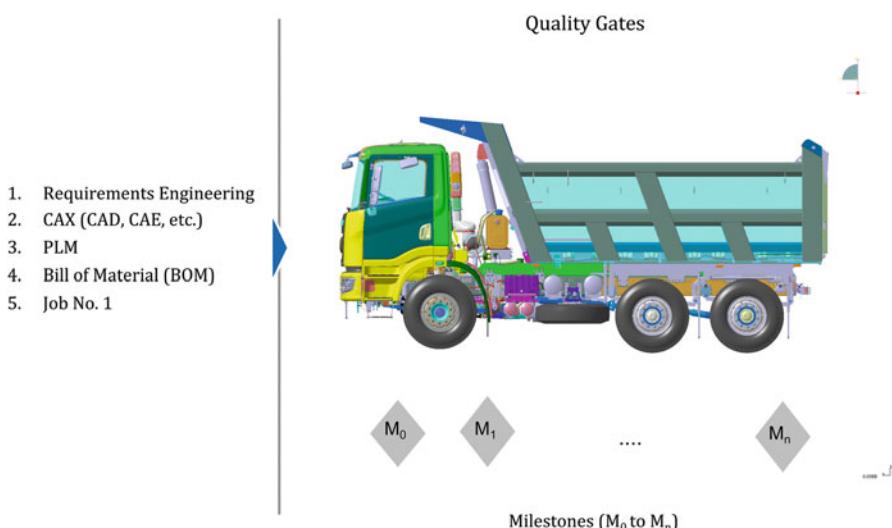


Fig. 3.4 A vehicle development system defining a set of milestones (M_0 – M_n) from concept to Job No. 1. For example, Truck Development System at Ashok Leyland (Truck model with kind approval from Ashok Leyland) and Bharat-Benz

VR systems were not developed to meet a specific need; they were developed because they were possible. Hence, through the integration of real-time computing, VR enables the user to get deeper insight by moving around in the virtual space of the respective application domain. By embedding temporal concepts, VR can be used as the basis for simulation, analysis, and prognosis of complex processes. Furthermore, underlying databases offer the ability to efficiently store and retrieve huge amounts of data for the modeling of real-world process domains. The data itself can be visualized in rendered 3D models (Möller 2000, 2016).

- *Augmented Reality (AR)*: Computer-assisted perception is the integration of digital information with the user's environment in real time. Unlike VR, which creates a totally artificial environment, AR uses the existing environment and overlays new information on top of it which expands the real environment around the virtual aspect. Augmented reality is routinely used to produce high-quality designs which have the required mechanical, thermal, and acoustic properties. It is this high degree of digitization that has popularized the term virtual product (Haas and Sinha 2004; Grieb 2010; Eigner and Stelzer 2013). The advantages of a fully digital model are compelling, such as the reuse of common parts, elimination of tedious prototyping, faster development cycles, fewer design flaws, and others. By using AR, physical mock-ups and artificially generated information can be mixed as desired. Thus, maintenance information, e.g., about the path of certain cables in the trunk of a vehicle or flow lines in a study of cabin interiors, can be combined. In AR and VR, the position of the head and the viewing direction can be recorded through sensors. Through this, the spatial orientation of the VR-scene is determined.

The need for new tools and methods in engineering is driven by the fast pace of change in this field:

- *Development work* is being outsourced more and more to suppliers offering engineering services. As a consequence, complex supply chains have to be managed. Both the OEM and the suppliers have to agree on rules and standards, share the same IT infrastructure, exchange complex results, and integrate them. If the supplier does not have the same IT infrastructure, a solution needs to be found for exchanging product data and information in an efficient manner.
- *The development process* itself is becoming increasingly distributed. Today, design teams across the world work on different parts of the structure and systems of a vehicle. They need tools that support efficient exchange of product data, collaborations, and the exchange of thoughts and ideas as well (Haas and Sinha 2004).
- *Pressure to shorten development cycles* increases the need for efficient knowledge sharing. Internet and web technologies play a fundamental role in sharing engineering knowledge across teams, business units, and companies (Haas 2000; Haas and Sinha 2004).

3.1.1 Requirements Engineering

Requirements engineering (RE) can be described as a process by which engineers identify a problem's context, locate the customers' needs and requirements within that context, and deliver specifications that meet those needs (Verner et al. 2005). In this regard, the requirements engineering process requires a considerable reduction in the complexity of the real-world problem's context. This reduction may incorporate a subset within which the customer's organization operates, resulting in a set of requirements defined by business needs and requirements (BNR) and stakeholder needs and requirements (SNR). These can then be converted into a system requirement specification (SyRS), as introduced by (Faulconbridge and Ryan 2014). As can be deduced from the literature, most requirements engineering methodologies have been created by engineers who like:

- Diagnosticity
- Precision
- Rigor

As described in Dorfmann and Thayer (1990), requirements engineering is the discipline concerned with analyzing and documenting requirements. In the publication of Kotonya and Sommerville (1998), requirements engineering is defined as systematic process of:

- Analyzing
- Documenting
- Eliciting
- Managing
- Understanding

The ISO/IEC 29148 (ISO/EIT 2011) standard defines the requirements of an engineered system and the life cycle process (see Sect. 3.4) as a process by which the acquirer and the suppliers of a system:

- Articulate
- Discover
- Document
- Review
- Understand

Hence, requirements engineering is a core process in product development, defining, documenting, and maintaining requirements, which document the physical and functional needs that a particular design of a product or a process must be able to perform. The fields concerned with requirements engineering are systems and software engineering. The term came into general use in the 1990s with the publication of an IEEE Computer Society tutorial by Thayer and Dorfmann (1997)

and the establishment of a conference series on requirements engineering. Alan M. Davis maintains an extensive bibliography of requirements engineering (Davis 2011).

The activities involved in requirements engineering vary widely, depending on the type of system being developed and the specific style guide practices of the organization involved. These may include (URL3 2017):

1. *Requirements Elicitation*: Practice of collecting requirements of a system from users, customers, and other stakeholders; sometimes also called requirements gathering.
2. *Requirements Identification*: Verify that acceptance criteria are established in the software requirements for each of the identified requirements. Such criteria should be used for verification and validation (V&V) planning and performance as defined in each related life cycle phase.
3. *Requirements Analysis and Negotiation*: Tasks determining the needs or conditions to meet for a new product with regard to possibly conflicting requirements of the various stakeholders.
4. *Requirements Specification*: Documenting the requirements in a requirements document.
5. *System Modeling*: Developing models of the new system, often using a notation such as the [Unified Modeling Language](#) (UML); UML profiles are intensively used in modeling domain-specific distributed applications.
6. *Requirements Validation*: Checking that the documented requirements and models are consistent and meet stakeholder needs.
7. *Requirements Management*: Managing changes to the requirements as the system is developed and put into use.

These requirements are presented as chronological stages; in practice, there is considerable interleaving of these activities.

3.1.2 Design as a Multiparameter Optimization Problem

In today's highly competitive business environment, large-scale engineering (LSE) organizations are constantly searching for ways to differentiate their products from their competitors. Typically, this is achieved by increasing the level of innovation (see Chap. 1), embedding additional valuable features and functionality in their products, and reducing the development cycle time. This requires a well-integrated product development process that ensures intrinsic innovation and optimal designs that reduce the number of prototype iterations. This is accomplished by making use of an optimal design mechanism which is predictive in a single-dimensional setting with regard to the underlying probabilistic and mathematical statistical methods. Let's assume the shortest return on investment (ROI) time with regard to a design is one that has to compare different scenarios based on what will be able to decide as best fit. In the case of two designs where two engineering groups work independently at different locations, the ROI

can be proven by the means of random numbers from the uniform distribution on [0,1]. With regard to the expected ROI, the calculation can be done based on the two assumed uniform random variables [0,1] and a comparison setting of 1/3. Let v_1 and v_2 denote the valuations of the designs. With the definition of uniform random variables,

$$\Pr[v_i < x] = x$$

implying that

$$\Pr[v_i > x] = 1 - x$$

Since v_1 and v_2 are assumed to be independent results,

$$\Pr[\min(v_1, v_2) > x] = \Pr[v_1 > x \wedge v_2 > x] = (1 - x)^2$$

In the case of an expectation of nonnegative random variables, one gets

$$E[\min(x_1, x_2)] = \int_{\infty}^0 \Pr[\min(v_1, v_2) > x] dx$$

$$\int_0^1 (1 - x)^2 dx = \int_0^1 (1 - 2x - x^2) dx = x - x^2 + \frac{1}{3}x^3 \Big|_0^1 = \frac{1}{3}$$

As a result, the question at this point is whether it is possible to do better and how.

Multidimensional settings are unlikely to permit succinct descriptions of optimal mechanisms. Optimal mechanisms in multidimensional settings are unlikely to have practical implementations. In comparison to this, product data management (PDM) typically encompasses multiple products' technical specifications, engineering models, design drawings, bills of material (BOMs), and related documents. PDM typically controls product-related data and organizes workflow through the entire product life cycle. It provides version control and security to ensure that the information stored in the central repository is accurate and up-to-date, which in turn can reduce data processing and make operations more efficient.

In this regard, PDM is a precursor and major component of product life cycle management (PLM), a strategy for managing and collaborating around product information (see Sect. 3.4). PLM is a holistic, enterprise-wide approach to managing all product data and processes across the entire life cycle, from development and production to sales and maintenance. The potential benefits derived from PLM can be seen in all areas along the value chain, based on the life cycle approach and the holistic view that accompanies it. Using this concept, for example, products can be brought to the market more quickly, the product in the service phase can be supported by better services, and the disposal can be anticipated ex ante in the product concept. In summary, PLM's potential benefits can be divided into the following four areas:

- *Business Performance*: This includes, in particular, the value of data and information as well as their management.
- *Financial Advantages*: Are particularly evident in an increase in profit, which is realized, for example, by an early market launch of the product, and thus higher sales figures, or by lowering costs.
- *Quality Advantages*: Product quality refers not only to the quality of the product but also to the conformity of customer requirements and product performance. This means that PLM cannot only detect production errors at an early stage but also reduce rework and/or customer complaints. It can be used to implement product variants that are offered with a reasonable cost-benefit ratio and are valued by the customer.
- *Time Advantages*: An efficient design leads to a reduction in project processing times, throughput times, problem-solving times, and the time to market.

PLM can be applied to several areas of a company, but the penetration, breadth and sophistication of the solution is domain-specific. PLM was developed mainly for use in product development, and many PLM-based methods and tools focus on this area of the value chain. The PLM function model consists of core data management, product data creation, process management, and process data integration management (Schuh 2012).

While PDM is a standard component of PLM, it is also offered as a dedicated module in many enterprise resource planning (ERP) suites of integrated applications for core business processes using common databases maintained by a database management system. The suite of integrated business activities is used to collect, store, manage, and interpret data from many business activities, including:

- Finance
- Inventory management
- Manufacturing
- Marketing and sales
- Materials management
- Product planning
- Production planning
- Purchase
- Service delivery
- Shipping and payment

This allows the ERP to share data across the various departments of the company, such as accounting, manufacturing, purchasing, sales, and others, that provide the data. Therefore, ERP facilitates information flow between all business functions and manages connections to outside OEM and Tier 1 suppliers.

3.2 Automotive Modularization and Platforms

The use of platforms and the development of product families based on platforms are not new and have been widely used globally in different industrial sectors. The production of product families based on platforms is a strategy that has been successfully applied in the automobile sector to cope with a rapidly increasing product portfolio in the late 1990s up to the first years of the twenty-first century. This expansion of the product portfolio was made possible by using a standard platform for different models in the same segment (Lampón et al. 2015). This has led to greater efficiency in design and development processes, as well as greater economies of scale in production and procurement due to greater standardization (Becker and Zirpoli 2003; Cusumano and Nobeoka 1998; Muffatto 1999; Muffatto and Roveda 1999; Wilhelm 1997).

The greatest changes regarding process flexibility and efficiency took place around the turn of the century, when platforms were reduced and standardized to develop a single common platform for different models within the same segment (Holweg 2008). The main objective of such standardization was to rationalize the number of platforms and to share common components and systems among those models assembled on a single platform (Patchong et al. 2003). This standardization strategy focused on aspects of product development—the simplification of engineering and design processes, reduction of costs and development time, and ability to update products (Muffatto 1999; Suk et al. 2007). It also aimed to take advantage of the economies of scale resulting from a greater number of common units per platform, such as savings on the purchase of components (Korth 2003). From a manufacturing perspective, the platform standardization strategy offered advantages for globalizing production processes because it allowed flexibility among plants, possibility of transferring production from one plant to another (Robertson and Ulrich 1998; Smith and Reinertsen 1998), and cost reduction by using resources on a worldwide scale (Wilhelm 1997).

In recent years, the platform strategy has been reviewed, and new modular platforms have been, and will continue to be, adopted in the sector (Sehgal and Gorai 2012; Global Automotive Modular Platform Sharing Market Analysis 2013–2023). From a technical point of view, such modular platforms are configured differently based on a single scalable design, allowing for changes in structural dimensions. This modularization of the structural element of an automobile, the platform, means that it is possible to not only assemble several models within a single segment (same size), as with the classic standard platforms, but also several models in different segments (different sizes) (Buiga 2012; Lampón et al. 2015). Although the modular platform strategy has not yet been fully implemented, the forecasts of several automobile manufacturers (e.g., Volkswagen, PSA Peugeot-Citroën, Nissan-Renault) mention savings in product development costs and in the procurement of components from the auxiliary industry (Lampón et al. 2015; Lampón and Cabanelas 2014).

The result of the platform standardization strategy is an improved operational flexibility. The modular platform strategy allows plants in different segments to

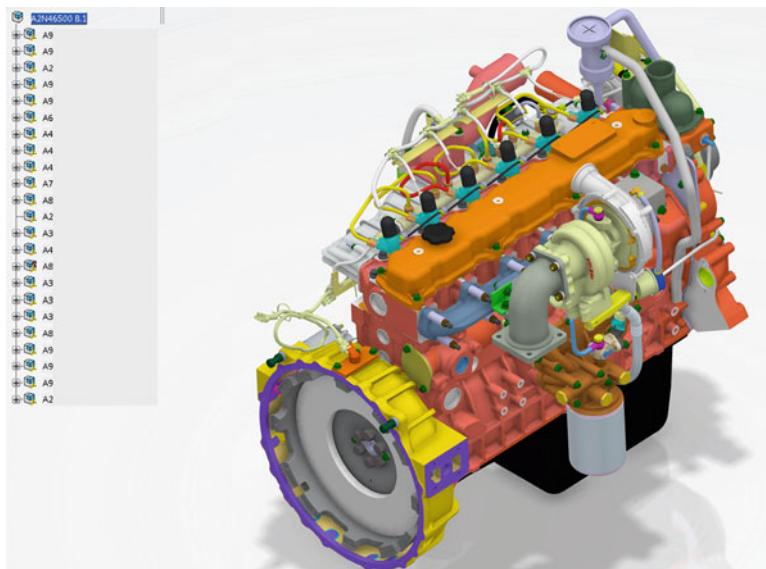


Fig. 3.5 Modular CAD model of an engine in CATIA system environment. It defines the basis for a modularization platform (Courtesy Ashok-Leyland ([URL9 2017](#)))

share the same modular platform, so the production network can include a larger number of plants. Manufacturing mobility, expressed in terms of operational network manufacturing flexibility, is the ability to make optimal use of the total global capacity of the network by shifting production volumes between plants in order to cope with volatility in the international business environment (Lampón et al. 2015).

In Fig. 3.5 an example of a CAD model of an engine in the CATIA system environment is shown.

3.3 Virtual Product Creation

Product data management (PDM) systems help engineers manage data and processes related to the product development life cycle. As sophisticated and automated design tools (e.g., CAD systems) became available, the amount of design data increased dramatically. PDM systems offer the technology to manage such data, and keep track of various product data that already exist in various forms. Hence, one can view PDM systems as having meta-knowledge about the product development life cycle. This meta-knowledge is in the form of knowledge about product structure, processes, and access/change management rules. Basic functions of a PDM system include (Bilgic and Rock 1997; Grieves 2006; Eigner and Stelzer 2013):

- *Change Management:* The ability to define and manage changes to product data over the life cycle. Change management is process oriented, defining the events in the cycle of reviewing and approving changes.
- *Classification:* The ability to classify parts by their structure, function, or processes for manufacturing.
- *Design Release Management:* Process of controlling design data with an electronic vault with check-in/check-out, release level maintenance, access security, and review and approval management. This function encompasses the management of all forms of digital product data: CAD files, geometric models, images, documents, and others.
- *Impact Analysis:* The ability to detect the effects of a design change on the overall product design life cycle.
- *Product Structure Management:* The ability to define, create, maintain, modify, and display multiple versions of the product structure, including design options and activities over the product data life cycle.
- *Systems Management:* Usually perceived as the use of project-oriented scheduling techniques with work breakdown structures but which should be able to manage any facet of systems design (cost, quality, risk, in addition to workflow).

Thus, PDM is going through a natural evolution cycle. It started as a typical engineering data storage and retrieval system; but today it encompasses a number of processes and technologies because PDM systems are designed to manage product data and documents produced by a mixture of CAD, CAM, CAE, material requirements planning (MRP), and other engineering and manufacturing control systems (Sendler and Wawer 2011; Grieb 2010). With regard to the major functionalities of PDM systems listed above, some others are:

- *Configuration Management, Product Structure Management, and Engineering Change Management:* This functionality focuses on creating and managing production configurations and BOMs. Features include storing and managing previous build versions, managing effectivities, supervising several different configurations of the product simultaneously, managing engineering change requests, and providing information about different aspects of the production, based on the design discipline (such as mechanical, electronic, and even financial).
- *Program Management:* PDM-based project management includes real-time resource scheduling and project tracking, progress reporting, status and location of deliverables, and identifying individuals working on various deliverables. This functionality is still evolving and is often supported by third-party applications.
- *Workflow:* This is a key element in both process and configuration management which automatically transfers the results of one set of tasks to the next and can alert the concerned individual (or send a necessary interrupt to the next process), that work is pending or in progress. The work defined in a workflow system can be serial or concurrent; in either case, work can proceed simply once a step is completed, or it can be based on conditional requirements. Repetitive processes

are ideal for workflow operations. If equipped with a graphical interface, the workflow enables users to see where resources and deliverables are lodged within the overall engineering processes.

- *Interfaces and Data Exchange:* Several processes in the product development life cycle are supported by various tools. These tools generate data in the different required formats.
- *Vault:* As the name suggests, a vault stores, organizes, controls, retrieves, and protects product data. Data need not be colocated. The idea is merely to organize wherever there is a need. The key is in the supervision of the data access by various parameters defined by the organization, such as individuals, groups/teams, projects, supervisors, public, etc.
- *Process Management, Process Modeling, and Design Control:* This functionality is for controlling the creation and modification of product configurations, part definitions, data relationships, and other product data throughout the product development life cycle. Product data management records each step in the creation of data, thus providing an audit trail that is crucial for tracking design intent and design changes reversed.

One of the biggest challenges faced by PDM systems is to integrate all these data formats through standards like ISO 10303, Standard for the Exchange of Product Model Data (STEP), a series of standards that are the essential basis for product data technology, and to interface various tools through technologies such as system integration and middleware technologies. With the increasing distribution in various engineering disciplines, approaches to the integration of computer-aided systems without STEP are unthinkable today. Against this background, the introduction of STEP is a major challenge for all technical companies, as well as the further development of the product data model (Anderl and Trippner 2000).

With the accelerated pace of technological change that global manufacturing companies have to face, they must implement manufacturing engineering with proper consideration given to the manufacturing process and resources that are appropriate for the particularities of each manufacturing site. Particularly for effective ERP system implementation under a global manufacturing environment, the PDM integration for product data is one of the important keys to success. For such product data integration, engineering bills of material (EBOMs) need to be transformed to manufacturing bills of material (MBOMs), but the MBOM transformation must be done in such a way as to fit the particularities of each manufacturing site. In this process, a methodology appropriate for integration and transformation is required. Therefore, digital manufacturing can be proposed as the key tool for data integration between PDM and ERP. Digital manufacturing, as a technology possessing the physical and logical computer modeling and simulation technique for actual manufacturing, provides the methodology for transforming EBOM to MBOM that fits the particularities of each manufacturing site, based on the process and resource models which reflect the particularities of each manufacturing site. It also provides the methodology for MBOM verification and process and resource

Big Picture 2016: Global Cooperation Processes based on JT

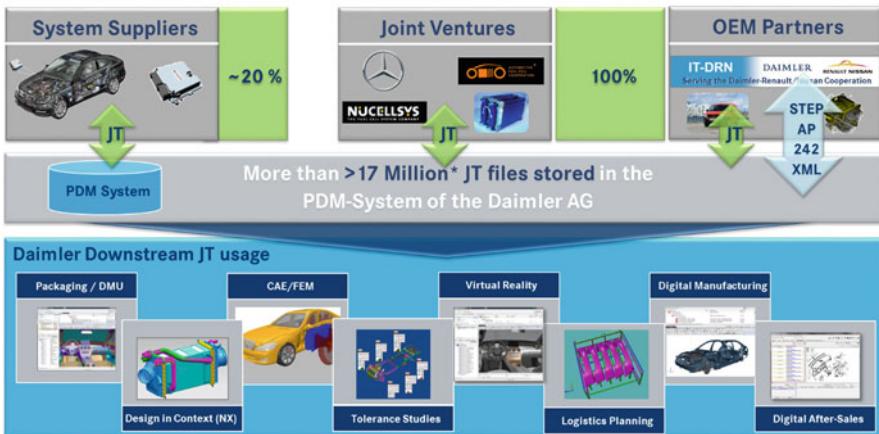


Fig. 3.6 Virtual product creation and global collaboration processes at Mercedes-Benz cars (With kind approval from Daimler)

model integration. Using such a method, the MBOM and the process and resource data, verified and appropriate for each manufacturing site, can be sent to the ERP system (Lee et al. 2011).

The virtual product creation and global collaboration processes based on the Jupiter Tesselation (JT) exchange format shown in Fig. 3.6.

PDM systems are designed to:

- Store, control, and manage documents, such as text, graphics, drawings, etc., and other information about products.
- Enforce and manage the process of creating and using these documents.
- Enable engineering teams to share information on products and process quickly and with consistent accuracy. Permit the integration of techniques such as computer-aided design and manufacturing (CAD/CAM) and FEM into coherent business systems across an entire enterprise.

Integration with PDM technologies is critical as well because the design data is typically managed through these tools. The blending of technologies and methods is an integral part of PLM initiatives—solutions that facilitate collaborative work processes, supplier integration, enterprise application integration, and a host of other approaches addressing the needs of the extended enterprise (Eigner and Stelzer 2013; Seiffert and Rainer 2008). Change management defines a workflow for initiating and approving design changes. The workflow itself is an integral part of the PDM infrastructure. In a typical development project, there will be thousands of change requests every day. With a web conferencing infrastructure in place, change management is supported by sharing important background information ad hoc,

communicating the reasons for a particular change request, and by getting immediate feedback which allows faster decision-making and shorter time to market. The CAD-based PDM is shown in Fig. 3.7.

The team-oriented structure of modern organizations means that many decisions require insight and approval from a lot of different sources. Electronic meeting tools enable dispersed teams to collaborate easily, ultimately delivering faster time to consensus and, hence, a shorter time-to-market (TTM) for new products and services (Cusumano 2008). In today's increasingly competitive world, a shorter TTM delivers one of the highest payoffs. Furthermore, presence-based visual collaboration tools enable workers to immediately locate and communicate with co-workers and partners, regardless of their current location. That way, issues can be resolved in real-time and without the wasted cycles of voice mail and email exchanges. The result is decreased frustration, faster problem resolutions, and increased satisfaction.

- *Productivity/Efficiency:* Videoconferencing and visual collaboration tools are moving away from the scheduled environment of the departmental conference room to the ad hoc, unscheduled work style of the desktop. Conferencing on demand delivers immediate productivity boosts and time savings to all knowledge workers by enabling them to integrate visual communications and desktop-based collaboration tools into their normal workflow process. The result is an immediate impact on the bottom line (Spath and Kern 2003) as employee productivity is a main concern across different industries.
- *Higher Impact and Focus:* Videoconferencing can help an organization inject higher impact into their meetings and conference calls, especially when compared to an audio-only meeting. Higher impact during meetings translates into shorter, more effective meetings with minimal workflow disruption. Studies have shown that videoconferencing meetings tend to be shorter than in-person meetings, leading to less wasted time (Prasad 2003).
- *Competitive Advantage:* Using web and videoconferencing can generate a competitive advantage. For example, a company that recruits by web-based video-conference rather than flying recruiters or candidates around the country can interview more people, from more locations, in less time, and with less cost and disruption to executive schedules, thereby making better hiring decisions. Using advanced collaboration tools enables companies to better support remote workers and build better dispersed teams, thereby giving more employees more choices on where they want to work.
- *Improved Management of Dispersed Teams:* Large research and development organizations have subject matter experts and qualified resources located around the world. Web-based videoconferencing allows companies to more easily deploy and manage those globally dispersed resources by allowing impromptu, face-to-face meetings between managers, subordinates, and remote peers.

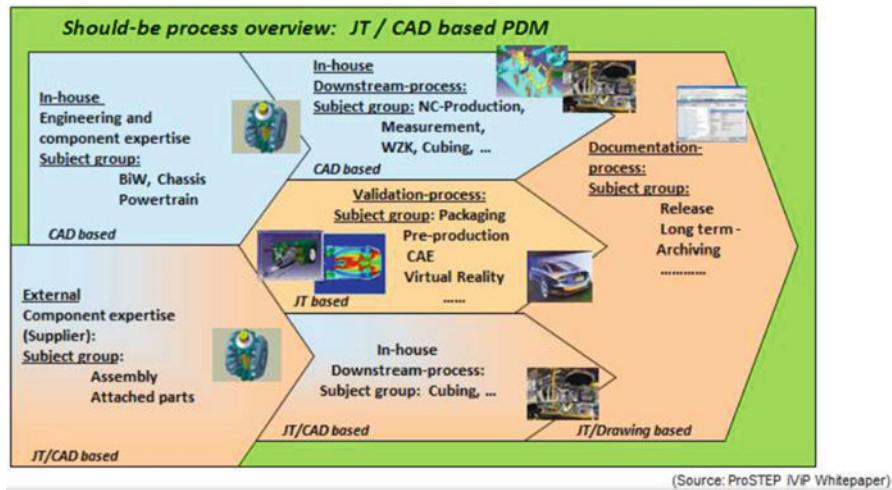


Fig. 3.7 JT/CAD-based PDM (ProSTEP (URL7 2017))

3.4 Product Life Cycle Management

Product life cycle management is an important task in today's automotive systems and software engineering processes (Eigner and Stelzer 2013). It represents the business activity of managing a company's products across their life cycles, from the first conceptual ideas for a product all the way through until it is retired and discarded. At its highest level, product life cycle management is used to increase product revenues, reduce product-related costs, and increase the value of the product portfolio as well as the value of current and future products for both customers and shareholders (Stark 2011, 2016). Thus, product life cycle management can be interpreted as some kind of digital archive, enabling the seamless integration of all information generated during the life cycle of a product.

The term product, as used in this book, refers to a physical, tangible product that can be owned, traded, and distributed to different places at different times without changing its identity. However, a product can also be something very intangible such as a piece of software, an algorithm, etc. Hence, a product with regard to product life cycle management is referred to by different names, such as (Saaksvuori and Immonen 2008):

- Intangible products, nonphysical products that are not services
- Physical, tangible products
- Services

All kinds of products, as mentioned above, must be managed during all phases of the product life cycle. This will ensure that a product works well with regard to the preassigned product specs; but managing a product across its life cycle is not easy to achieve. During the development of a product, it does not physically exist; and once it becomes a physical, tangible product, it will be used by its customers where it is difficult for a company to keep control over it. Moreover, responsibility for a product within a company is often difficult to manage at different phases of the product life cycle. At one time, marketing may be responsible; and at another time, engineering, after sales services, and maintenance may be responsible. Maintaining a common coherent approach among these different organizational responsibilities, which may have different objectives, working methods, and applications, can be difficult and time consuming. Thus, losing control over a product may result in serious consequences for the company (Stark 2011, 2016).

3.4.1 Loss of Control in Life Cycle Management

Loss of control in product life cycle management can happen at different stages of the product life cycle. The earliest possibility of a loss of control may occur in the product development phase, where a delayed product market launch causes the project to exceed the targeted cost and has a huge impact on the product's ROI. Assuming a product has an expected cumulative turnover of \$100 million over a period of 5 years. The monthly loss with regard to the delayed market launch may result in a \$1.6 million loss.

In the case of a company losing life cycle control due to a customer's dissatisfaction with the quality of a product, or worse, damages resulting from the use of the product, may result in damage to the company's image and a loss of customers concerned about product problems. Moreover, this could also include a loss of revenue to companies that launch products much faster into the market, as well as reduced profit due to the cost of recalls and legal liabilities resulting from product use (Stark 2011). The monetary losses cannot be quantified in this case, as they are dependent on the number of items sold, the market penetration, and the list price.

Loss of control of the product life cycle can also occur during product manufacturing. For example, in 2006, computer makers, such as Apple Computer, Dell, Hitachi, Lenovo, and Toshiba, announced the replacement of Sony-made lithium-ion batteries that could overheat in certain circumstances and pose a safety risk. In January 2013, after problems with lithium-ion batteries on JAL and ANA Boeing 787 s, the Federal Aviation Administration (FAA) ordered all 787 s to be grounded. The order was lifted in April 2013 after battery and containment systems had been redesigned (Stark 2011, 2016).

Loss of control of the product life cycle can also occur during product use. For example, in 2016, smartphone manufacturer Samsung had a serious problem with its new Galaxy Note 7 product. It could overheat in certain circumstances and pose a safety risk, which would in turn damage the company's image and result in the loss of customers concerned about product problems. Meanwhile, Samsung developed

an extensive battery check protocol to ensure safety of the battery from component to the complete device.

Loss of control in the product life cycle can also occur when products are sold in big numbers on the market. In October 2003, Nissan Motor Company said it would recall 2.55 million cars at an estimated cost of ¥15–16 billion (\$138–148 million) due to an engine defect. Over a few months in late 2009 and early 2010, Toyota announced recalls of more than 8 million cars due to concerns over accelerator pedals and floor mats. The cost was estimated at \$2 billion. In January 2010, Honda announced the recall of more than 600,000 cars to fix a switch defect that could lead, in some cases, to a fire. In June 2010, GM recalled over a million vehicles due to thermal incidents with heated washer fluid systems. GM listed 84 recalls affecting 30.4 million vehicles (Stark 2011, 2016).

An unprecedented loss of control in the product life cycle was discovered in the so-called diesel gate scandal of the Volkswagen Group—Europe's largest automotive group. Volkswagen publicly acknowledged on September 20, 2015, that it had massively manipulated several of its models of diesel vehicles, with the help of software, in order to comply with the legal emission standards. Several million vehicles are affected worldwide, and they will be recalled to either fix them or buy them back.

3.4.2 Systems Engineering Approach

The systems engineering approach gives insight into how engineering concepts and procedures could be applied to individual and/or special requirements of products. This finally has resulted in a wide range of definitions of systems engineering, each of which is subtly different because it tends to reflect the particular focus of its source. The following are some of the more accepted and authoritative definitions of systems engineering from relevant standards and documents given in Faulconbridge and Ryan (2014):

“Systems engineering is the management function which controls the total system development effort for the purpose of achieving an optimum balance of all system elements. It is a process which transforms an operational need into a description of system parameters and integrates those parameters to optimize the overall system effectiveness.” (DSMC 1990)

“An interdisciplinary approach encompassing the entire technical effort to evolve and verify an integrated and life cycle balanced set of system, people, product, and process solutions, that satisfy customer needs. Systems engineering encompasses: the technical efforts related to the development, manufacturing, verification, deployment, operations, support, disposal of, and user training for, system products and processes; the definition and management of the system configuration; the translation of the system definition into work breakdown structures, and development of information for management decision making.” (EIA/IS-632-1998 1994)

“Systems engineering is the selective application of scientific and engineering efforts to transform an operational need into a description of the system configurations which best satisfies the operational need according to the measures of effectiveness; integrate related technical parameters to ensure compatibility of all physical, functional, and technical program interfaces in a manner which optimizes the total system definition and design;

and integrate the efforts of all engineering disciplines into the total engineering efforts.” (SECMM-95-01 1995)

“Systems engineering is an interdisciplinary, comprehensive approach to solving complex system problems and satisfying stakeholder requirements.” (Lake 1996)

“System engineering is an interdisciplinary approach and means to enable the realization of successful systems. It focusses on defining customer needs and required functionality early in the development cycle, documenting requirements, and then proceeding with design synthesis and system validation while considering the complete problem: operations, cost and schedule, performance, training and support, test, manufacturing, and disposal. Systems engineering considers both the business and the technical needs of all customers with the goal of providing a quality products that meets the user needs.” (Haskins 2006)

It should be noted that the above definitions come principally from earlier standards. Systems engineering standards, such as SITEC 15288, ANSI/EIA-632, and IEEE-STD-1220, do not contain any definition of systems engineering but refer more generically to engineering of systems. The worldwide standardization bodies, International Standards Organization (ISO), also called International Organization for Standardization, and the International Electrotechnical Commission (IEC), specialize in the development of international standards through technical committees established by the respective organization to deal with particular fields of technical activity. The ISO and IEC technical committees collaborate in fields of mutual interest. In this regard, ISO/IEC 15288 is a system engineering standard covering engineering development processes and life cycle phases. The initial planning for the ISO/IEC 15288:2002(E) standard started in 1994 when the need for a common systems engineering process framework was recognized. In 2004, this standard was adopted as IEEE 15288; and it was updated on February 1, 2008. Standard ISO 15288 is managed by ISO/IEC JTC1/SC7, which is the ISO committee responsible for developing ISO standards in the area of software and systems engineering. Standard ISO/IEC 15288 also provides processes that support the definition, control, and improvement of the life cycle processes used to manage a product across its life cycle. Organizations can use these life cycle processes when acquiring and supplying products.

The life cycle phases from ISO/IEC 15288, Systems Engineering-System Life Cycle Processes, vary according to the nature, purpose, use, and prevailing circumstances of the engineered product. Despite an infinite variety of life cycle models, the most important phases are the conceptualization of the business needs for the product, its realization, its evolution, and its ultimate disposal, as shown in Fig. 3.8.

From Fig. 3.8, it can be seen that the engineered product will be phased out after operational use and product support, which are part of the ultimate disposal phase. It also can be seen in Fig. 3.8 that the systems engineered product remains in service in

Conceptualization Phase	Utilization Phase	Evolution Phase	Ultimate Disposal Phase
-------------------------	-------------------	-----------------	-------------------------

Fig. 3.8 Phases of a generic product life cycle model



Fig. 3.9 ISO 15288 life cycle phases

Table 3.1 Systems engineering effort across ISO 15288, life cycle phases

Phase	Conceptualize	Develop	Operational test and evaluation	Transition to operation
% Effort	23	36	27	14
Standard derivation	12	16	13	9

the utilization phase. During utilization, the engineered product can undergo modifications and upgrades to rectify performance shortfalls in order to meet changing operational requirements or external environmental constraints. This enables ongoing support for the product to be maintained or current performance or reliability to be enhanced, which is represented by the evolution phase in Fig. 3.8. At the very beginning of the conceptualization phase, the product to be engineered is defined in terms of technical specifications as determined by business needs and requirements, ensuring that only feasible, cost-effective products are launched in the market.

The phases shown in Fig. 3.8 can be slightly modified to reflect the influence of the ANSI/EIA 632 model, as shown in Fig. 3.9 (Valerdi and Wheaton 2005).

Each stage shown in Fig. 3.9 has a distinct purpose and contribution to life cycle management and represents one of the major life cycle periods associated with a product. The stages also describe the major progress and achievement milestones of the product throughout its life cycle. Understanding when systems engineering is performed relative to the product life cycle defines anchor points for the life cycle model. The typical distribution of systems engineering effort across the life cycle phases across the entire organization studied is shown in Table 3.1. It is important to note the standard deviation for each of the phases (Valerdi and Wheaton 2005).

3.4.3 Product Life Cycle Stages

In contrast to the systems engineering approach to life cycle management discussed in Sect. 3.4.2, the product life cycle approach in this section is based on four stages, each with its own characteristics and activities based on the business trying to manage the life cycle of its own particular products, as shown in Fig. 3.10 (URL4 2017).

From Fig. 3.10, it can be seen that after introduction of the product to the market, sales increases quickly in the life cycle growth phase. The transition from the life cycle growth phase to the life cycle maturity phase is shown in the graph in Fig. 3.10

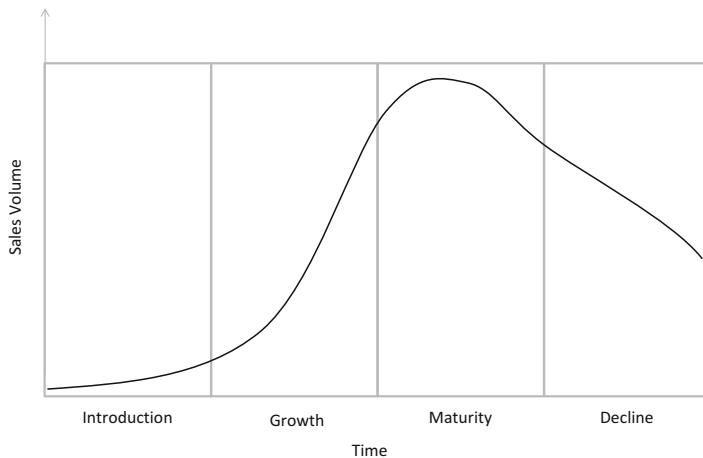


Fig. 3.10 Product life cycle stages

illustrating a recognizable limited value of product sales with a trend to reduced sales numbers, equivalent with a reduced sales volume in the life cycle maturity phase. The decline in sales turnover in the life cycle saturation phase indicates that market volume and market penetration are no longer expandable, which calls for the market launch of a revised product version with additional features or a new product which may result in renewed growth after market launch.

Thus, the product life cycle stages represent the following activities with their corresponding revenues:

- ***Introduction Stage:*** This product life cycle stage may be expensive for a company launching a new product. The size of the market for the new product is small at product launch, which means sales are low; however, sales will increase. On the other hand, the cost of activities, such as R&D, consumer testing, and the marketing needed to launch the product, can be high, especially if the product is targeting a competitive sector.
- ***Growth Stage:*** This product life cycle stage is typically characterized by a strong growth in sales and profits; and because the company can start to benefit from economies of scale in production, the profit margins, as well as the overall amount of profit, will increase. This makes it possible for businesses to invest more money in promotional activity to maximize the potential of this product life cycle stage.
- ***Maturity Stage:*** During this product life cycle stage, the product is established; and the aim for the manufacturer is to maintain the market share it has built up. This is probably the most competitive time for most products, and businesses need to invest wisely in any marketing they undertake. They also need to consider any product modifications or improvements to the production process which might give them a competitive advantage.

- ***Decline Stage:*** During this product life cycle stage, the market for a product will start to shrink, known as the decline stage. This shrinkage could be due to the market becoming saturated or because the consumers are migrating to a different type of product. While this product life cycle stage may be inevitable, it may still be possible for companies to make some profit by migrating to less expensive production methods and cheaper markets.

3.4.4 Software Life Cycle Processes

Standard ISO/IEC 12207:2008 establishes a common framework for software life cycle processes which is based on a well-defined terminology that can be used by the software industry. It contains processes, activities, and tasks that are to be applied during the acquisition of a software product or service and during the supply, development, operation, maintenance, and disposal of software products. Software includes the software portion of firmware ([URL5 2017](#)).

Standard ISO/IEC 12207 was published on August 1, 1995, and was the first international standard to provide a comprehensive set of life cycle processes, activities, and tasks for software that is part of a larger system and for stand-alone software products and services. That international standard was followed in November 2002 by ISO/IEC 15288 which addressed system life cycle processes. The ubiquity of the software meant that the software and its design processes should not be considered separately from those systems but be considered as an integral part of the system and system design processes. The ISO/IEC 12207 amendments in 2002 and 2004 added process purpose and outcomes to the international standard and established a process reference model in accordance with the requirements of ISO/IEC 15504-2 ([URL6 2017](#)).

This international standard, a revision of the amended ISO/IEC 12207, is an initial step in the SC7 harmonization strategy to achieve a fully integrated suite of system and software life cycle processes and guidance for their application.

This revision integrates ISO/IEC 12207:1995 with its two amendments and applies SC7 guidelines for process definition to support consistency and improved usability. Project execution was carefully coordinated with the parallel revision of ISO/IEC 15288:2002 to align structure, terms, and corresponding organizational and project processes ([URL6 2017](#)).

This international standard can be used in one or more of the following modes ([URL6 2017](#)):

- ***By an Organization:*** To establish an environment of desired processes. These processes can be supported by an infrastructure of methods, procedures, techniques, tools, and trained personnel. The organization may then employ this environment to perform and manage its projects and progress systems through their life cycle stages. In this mode, this international standard is used to assess conformance of a declared, established set of life cycle processes to its provisions.

- *By a Project:* To select structure and employ the elements of an established set of life cycle processes to provide products and services. In this mode, this international standard is used in the assessment of conformance of the project to the declared and established environment.
- *By an Acquirer and a Supplier:* To develop an agreement concerning processes and activities. Via the agreement, the processes and activities in this international standard are selected, negotiated, agreed to, and performed. In this mode, this international standard is used for guidance in developing the agreement.
- *By Organizations and Assessors:* To perform assessments that may be used to support organizational process improvement.

This international standard contains requirements in four clauses ([URL6 2017](#)):

- *Clause 6:* Defines the requirements for the system life cycle processes
- *Clause 7:* Defines the requirements for specific software life cycle processes
- *Clauses of Annex A:* Provides requirements for tailoring of this international standard
- *Clauses of Annex B:* Provides a process reference model which may be used for assessment purposes

Five informative annexes support the harmonization strategy initiated by this revision ([URL6 2017](#)):

- *Annex C:* Expands on the history and rationale for the changes and provides high-level traceability among the international standards which were used as the input to this revision
- *Annex D:* Describes the alignment of the processes of ISO/IEC 15288 and ISO/IEC 12207, a key focus of this revision
- *Annex E:* Provides an example of a process view for usability intended to illustrate how a project might assemble processes, activities, and tasks of ISO/IEC 12207 to provide focused attention to the achievement of product characteristics that have been selected as being of special interest
- *Annex F:* Contains some examples of process descriptions that are considered useful to some readers of this international standard
- *Annex G:* Provides support for IEEE users and describes relationships of this international standard to IEEE standards

Readers of this international standard are advised to consult Clause 5 to gain an understanding of the key concepts used.

A future technical report (ISO/IEC TR 24748) will describe the relationships between this international standard and ISO/IEC 15288:2008.

3.5 Exercises

What is meant by the term *automotive development process*?

Describe the characteristics of an automotive development process.

What is meant by the term *conceptualization phase*?

Describe the characteristics of a conceptualization phase.

What is meant by the term *project phase*?

Describe the characteristics of a project phase.

What is meant by the term *validation phase*?

Describe the characteristics of a validation phase.

What is meant by the term *Stage-Gate-controlled development process*?

Describe the characteristics of a Stage-Gate-controlled development process.

What is meant by the term *computer-aided design*?

Describe the characteristics of a computer-aided design.

What is meant by the term *Bézier surface*?

Describe the characteristics of a Bézier surface.

What is meant by the term *rational B-splines*?

Describe the characteristics of a rational B-spline.

What is meant by the term *nonuniform rational B-splines*?

Describe the characteristics of nonuniform rational B-splines.

What is meant by the term *digital mock-up*?

Describe the characteristics of a digital mock-up.

What is meant by the term *computer-aided engineering*?

Describe the characteristics of computer-aided engineering.

What is meant by the term *concurrent design*?

Describe the characteristics of a concurrent design.

What is meant by the term *simultaneous engineering*?

Describe the characteristics of a simultaneous engineering development process.

What is meant by the term *virtual reality*?

Describe the characteristics of the virtual reality method.

What is meant by the term *augmented reality*?

Describe the characteristics of the augmented reality method.

What is meant by the term *requirements engineering*?

Describe the characteristics of the requirements engineering method.

What is meant by the term *multiparameter optimization process*?

Describe the characteristics of the multiparameter optimization process.

What is meant by the term *modularization*?

Describe the characteristics of the modularization process.

What is meant by the term *platform*?

Describe the characteristics of a platform.

What is meant by the term *virtual product creation*?

Describe the characteristics of the virtual product creation process.

What is meant by the term *product data management*?

Describe the characteristics of a product data management system.

What is meant by the term *CAD pipeline*?

Describe the characteristics of a CAD pipeline.

What is meant by the term *STEP standard*?

Describe the characteristics of the STEP standard.

What is meant by the term *finite element analysis*?

Describe the characteristics of the finite element analysis.

What is meant by the term *systems integration*?

Describe the characteristics of systems integration architecture.

What is meant by middleware?

Describe the middleware characteristics.

What is meant by the term *product life cycle management*?

Describe the characteristics of product life cycle management.

What is meant by the term *product*?

Describe the characteristics of a product.

What is meant by the term *intangible product*?

Describe the characteristics of an intangible product.

What is meant by the term *physical, tangible product*?

Describe the characteristics of a physical, tangible product.

What is meant by the term *loss of control in life cycle management*?

Describe the characteristics of loss of control in life cycle management.

What is meant by the term *systems engineering*?

Describe the characteristics of systems engineering.

What is meant by the term *utilization phase*?

Describe the characteristics of the utilization phase.

What is meant by the term *evolution phase*?

Describe the characteristics of the evolution phase.

What is meant by the term *ultimate disposal phase*?

Describe the characteristics of the ultimate disposal phase.

What is meant by the term *product life cycle stages*?

Describe the characteristics of the product life cycle stages.

What is meant by the term *introduction stage*?

Describe the characteristics of the introduction stage.

What is meant by the term *growth stage*?

Describe the characteristics of the growth stage.

What is meant by the term *maturity stage*?

Describe the characteristics of the maturity stage.

What is meant by the term *decline stage*?

Describe the characteristics of the decline stage.

What is meant by the term *software life cycle process*?

Describe the characteristics of the software life cycle process.

What is meant by the term *ISO/IEC 12207:2008*?

Describe the characteristics of the ISO/IEC 12207:2008.

What is meant by the term *clauses*?

Describe the characteristics of a clause.

References and Further Reading

- (Anderl and Trippner 2000) Anderl, R., Trippner, D. (Ed.): STEP Standard for the Exchange of Product Model Data (in German). Vieweg and Teubner Publ., 2000
- (Becker and Zirpoli 2003) Becker, C. M., Zirpoli, F.: Organizing new Product Development: Knowledge Hollowing-out and Knowledge Integration. – The Fiat Auto Case. International Journal of Operations and Production Management, Vol. 23, No. 9, pp. 1033–1063, 2003
- (Bézier 1986) Bézier, P.: The Mathematical Basis of UNISURF CAD System. Butterworths Publ., 1986
- (Bilgic and Rock 1997) Product Data Management: State of the Art and the Future. Proceedings of DETC'97 ASME Design Engineering Technical Conferences, 1997
- (Buiga 2012) Buiga, A.: Investigating the role of MQB Platform in Volkswagen Group's Strategy and Automobile Industry. International Journal of Academic Research in Business and Social Sciences, Vol. 2, No. 9, pp. 391–399, 2012
- (Colotla et al. 2003) Colotla, I., Shi, Y., Gregory, M.: Operation and Performance of International Manufacturing Networks. International Journal of Operations and Production Management, Vol. 23, No. 10, pp. 1184–1206, 2003
- (Cooper 2017) Cooper, R. G.: Stage-Gate: Roadmap for New Product Development. Published by Product Development Institute, 2017. Available from: <http://www.prod-dev.com/stage-gate.php>
- (Cooper and Edgett 2005) Cooper, R. G., Edgett, S. J.: Lean, Rapid, and Profitable New Product Development. Published by Product Development Institute, 2005
- (Cusumano 2008) Cusumano, M. A.: Managing software development in globally distributed teams. Communications of ACM, Vol 51, Issue 2, p 15–17, Feb 2008
- (Cusumano and Nobeoka 1998): Cusumano, M., Nobeoka, K.: Thinking beyond Lean, The Free Press, New York, 1998
- (Davis 2011) Davis, A. M.: Requirements Bibliography; <http://www.reqbib.com/>
- (Dorfmann and Thayer 1990) Dorfmann, M., Thayer, R. H.: System and Software Requirements Engineering. IEEE Computer Society Press, 1990
- (DSMC 1990) Defense Systems Management College: Systems Engineering Management Guide, Washington, DC, U.S. Government Printing Office, 1990
- (EIA&IS/632/1998) Systems Engineering, Washington, D.C., Electronic Industries Association (EIA), 1994
- (Eigner and Stelzer 2013) Eigner, M., Stelzer, R.; Product Lifecycle Management – A Guide for Product Development and Life Cycle Management (in German), 2nd ed, Springer, Berlin Heidelberg, 2009
- (Faulconbridge and Ryan 2014) Faulconbridge, I., Ryan, M. J.: Systems Engineering Practice. Argos Press 2014
- (Gausemeier and Plass 2013) Gausemeier, J., Plass, C.: Future-oriented company design - strategies, business processes and IT systems for the production of tomorrow (in German). Carl Hanser Publ. 2013
- (Gulati et al. 2000) Gulati, R., Nohria, N., Zaheer, A.: Strategic Networks. Strategic Management Journal, Vol. 21, pp. 203–215, 2000
- (Gusig and Kruse 2010) Gusig, L.-O., Kruse, A. (Eds): Vehicle Development in the Automotive Industry - Current Tools for Practical Use (in German). Carl Hanser Publ., 2010
- (Grieb 2010) Grieb, P.: Digital Prototyping – Virtual Product Development in Mechanical Engineering (in German). Carl Hanser Publ., 2010
- (Grieves 2006) Grieves, M.: Product Lifecycle Management. Tata McGraw-Hill, 2006
- (Haas 2000) Haas, R.: Engineering Knowledge Management - Current status and future challenges. Proceed. ICE Conference, Toulouse, France, 2000
- (Haas and Sinha 2004) Haas, R., Sinha, M.: Concurrent Engineering at Airbus – A Case Study. Internat. J. of Manufacturing Technology and Management (IJMTM) Vol 6, No 3, 2004.
- (Haskins 2006) Haskins, C.: Systems Engineering Handbook – Version 3. International Council of Systems Engineering, 2006

- (Holweg 2008) Holweg, M.: The Evolution of Competition in the Automotive Industry. In: Build to Order, pp. 13–34, Eds. Perry, G., Graves, S. Springer Publ., 2008
- (ISO/IEC 29148 2011) ISO/IEC 29148 FDIS Systems and Software Engineering– Life Cycle Processes–Requirements Engineering, 2011
- (Körber and Möller 2003) Körber, C., Möller, D. P. F.: Dynamic Depth Triangulation of Large NURBS Surfaces in Real-Time and its Application to Geoscience. In: Proceed. 4th Mathmod Conf., pp. 618–622. Eds.: I. Troch, F. Breitenecker, ARGESIM Report, Vol. 24, 2003
- (Korth 2003) Korth, K.: Platform reductions versus demands for specialization. *Automotive Design and Production*, Vol. 115, No. 10, pp. 14–16. 2003
- (Kotonya and Sommerville 1998) Kotonya, G., Sommerville, I.: Requirements Engineering. Wiley & Sons, 1998
- (Lake 1996) Lake, J.: Unraveling the Systems Engineering Lexicon. Proceedings of the INCOSE Symposium, 1996
- (Lampón et al. 2015) Lampón, J., Cabanelas, P., Benito, J. G.: The Impact of Implementation of a Modular Platform Strategy in Automobile Manufacturing Networks. Governance and Economics Research Network Working Paper B, 2015
- (Lampón and Cabanelas, 2014) Lampón, J. F., Cabanelas, P.: La Estrategia de Plataformas Modulares “Una Nueva Revolución en la Organización de la Producción en la Sector del Automóvil”. *University Business Review*, Vol. 42, pp. 14–31, 2014
- (Lee et al. 2011) Lee, C., Leem, C. S., Hwang, I.: PDM and ERP Integration Methodology using Digital Manufacturing to Support Global Manufacturing. *Int J Adv Manuf Technol.*, Vol. 53, No. 1, pp. 399–409, 2011. doi:<https://doi.org/10.1007/s00170-010-2833-x>
- (Miltenberg 2009) Miltenburg, J.: Setting Manufacturing Strategy for a Company’s International Manufacturing Network. *Internat. J. of Production Research*, Vol. 47, No. 22, pp. 6179–6203, 2009
- (Möller 2000) Möller, D. P. F.: Virtual Reality: A Methodology for Advanced Modeling and Simulation of Complex Dynamic Systems. In: 3rd Mathmod, pp. 505–508, Eds.: I. Troch, F. Breitenecker, ARGESIM Publ., 2000
- (Möller 2004) Möller, D. P. F.: Virtual Reality Framework for Surface Reconstruction. In: Networked Simulation and Simulated Networks, pp. 428–430, Ed. G. Horton, SCS Publ. House, 2004
- (Möller 2016) Möller, D. P. F.: Guide to Computing Fundamentals in Cyber-Physical Systems – Concepts, Design Methods, and Applications. Springer Publ. 2016
- (Muffatto 1999) Muffatto, M.: Introducing a Platform Strategy in Product Development. *Internat. J. of Production Economics*, Vol. 60/61, pp. 145–153, 1999
- (Muffatto and Roveda 1999) Muffatto, M., Rodeda, M.: Developing Product Platforms: Analysis of the Development Process. *Technovation*, Vol. 20, No. 11, pp. 617–630, 1999
- (Prasad 2003) Prasad, C. S. K.: Global Virtual Teams: A Capability Centric Model – Degree of Virtualness in Capabilities and Predictors. Indian Institute of Science (IISc), 2003
- (Patchong et al., 2003) Patchong, A., Lemoine, T., Kern, G.: Improving car body production at PSA Peugeot Citroen. *Interfaces*, Vol. 33, No. 1, pp. 36–49, 2003
- (Piegl and Tiller 1997) Piegl, L., Tiller, W.: The NURBS Book. Springer Publ. 1997
- (Prautzsch et al. 2002) Prautzsch, H., Boehm, W., Paluszny, M.: Bézier and B-Spline Techniques. Springer Publ. 2002
- (Robertson and Ulrich 1998) Robertson, D., Ulrich, K.: Planning for Product Platforms. In: *Sloan Management Review*, Vol. 39, Issue 4, pp 19ff, 1998
- (Rogers 2001) Rogers, D. F.: Introduction to Nurbs. Morgan Kaufmann Publ. 2001
- (Rudberga and Olhagerb 2003) Rudberga, M., Olhagerb, J.: Manufacturing Networks and Supply Chains an Operations Strategy Perspective. *Omega* Vol. 31, pp. 29–39, 2003
- (Saaksvuori and Immonen 2008) Saaksvuori, A., Immonen, A.: Product Lifecycle Management. Springer Publ. 2008
- (Schoenberg 1967) Schoenberg, I. J.: On Spline Functions. In: Sischa, O. (Ed.) In-equalities, pp. 255–291. Academic Press 1967

- (Schuh 2012) Schuh, G.: Innovation Management (in German) Ed. Schuh, G. Springer Publ. 22012
- (Schumaker 1981) Schumaker, L. L.: Spline functions: Basis Theory, Wiley Publ., 1981
- (SECMM-95-01) Systems Engineering Capability Maturity Model, Version 1.1, Carnegie Mellon University, Pittsburgh, P.A., Software Engineering Institute, 1995
- (Sehgal and Gorai 2012) Sehgal, B., Gorai, P.: Platform Strategy will shape future of OEMs. White Paper Evaluserve, 2012
- (Seiffert and Rainer 2008) Seiffert, U., Rainer, G. (Eds): Virtual product design for vehicle and drive in cars (in German) Vieweg and Teubner Publ., 2008
- (Sendler and Waver 2011) Sendler, U., Waver, U.: From PDM to PLM (in German), Carl Hanser Publ., 2011
- (Shi and Gregory 1998) Shi, Y., Gregory, M.: International Manufacturing Networks to Develop Global Competitive Capabilities. *Journal of Operations Management*, Vol. 16, pp. 195–214, 1998
- (Sinha and Haas 2006) K. Sinha, K., Haas, R., An Architecture for Integrated Simulation Driven Design, Industrial Simulation Conference (ISC) 2006, Palermo, Italy
- (Smith and Reinertsen 1998) Smith, P. G., Reinertsen, D. G.: Developing Products in Half the Time: New Rules, New Tools. John Wiley & Sons Inc., 1998
- (Sörensen 2006) Sörensen, D.: The Automotive Development Process – A Real Options Analysis. Deutscher Universitäts Verlag, 2006
- (Spath and Kern 2003) Spath, D., Kern, P. (Eds.): Office 21 - More performance in innovative work environments (in German). Egmont vgs Publ., 2003
- (Stark 2011) Stark, J.: Product Lifecycle Management. Springer Publ., 2011
- (Stark 2016) Stark, J.: Product Lifecycle Management, Volume 2. Springer Publ. 2016
- (Suk et al., 2007) Suk, E., de Weck, O., Kim, I. Y., Chang, D.: Flexible Platform Components Design under Uncertainty. *Journal of Intelligent Manufacturing*, Vol. 18, No. 1, pp. 115–126, 2007
- (Thayer and Dorfmann 1997) Thayer, R. H., Dorfmann, M.: Software Requirements Engineering. Wiley Publ., 1997
- (Tiller 1983) Tiller, W.: Rational B-Splines for Curve and Surface Representation. *IEEE Comput. Graph Appl.* Vol. 3, pp. 61–69, 1983
- (Valerdi and Wheaton 2005) Valerdi, R., Wheaton, M.: ANSI/EIA 632 As a Standard WBS for COSYSMO. Proceedings 5th Aviation, Technology, Integration, and Operations Conference (ATIO), Arlington, Virginia, 2005
- (Vereecke and Van Dierdonck 1999) Vereecke, A., Van Dierdonck, R.: Design and Management of International Plant Networks. Research Report Gent Academia Press, 1999
- (Verner et al. 2005) Verner, J. K., Cox, S., Bleistein, S., Cerpa, S.: Requirements Engineering and Software Project Success: An Industrial Survey in Australian and the U.S.. *Australian Journal of Information Systems*, Vol. 13, No. 1, 2005
- (Wilhelm 1997) Wilhelm, B.: Platform and modular concept at Volkswagen – Their effect on the assembly process. In: Transforming Auto Assembly. Shimokawa, K., Jurgens, U., Fujimoto, T. (eds.), Springer Publ., 1997
- (Woodward 1987) Woodward, C. D.: Blends in Geometric Modeling. In: Martin, R. R. (Ed.), Mathematical Methods of Surfaces II, pp 255–297. Oxford University Press, 1987

Links

- (URL1 2017) http://www.stage-gate.com/resources_stage-gate.php
- (URL2 2017) <http://wirtschaftslexikon.gabler.de/Archiv/14455/simultaneous-engineering-v6.html>
- (URL3 2017) http://en.wikipedia.org/wiki/Requirements_engineering#cite_note-5
- (URL4 2017) <http://productlifecyclesstages.com/>

- (URL5 2017) http://www.iso.org/iso/catalogue_detail?csnumber=43447
- (URL6 2017) <https://www.iso.org/obp/ui/#iso:std:iso-iec:12207:ed-2:v1:en>
- (URL7 2017) http://www.prostep.org/en/medialibrary/publications/?no_cache=1
- (URL8 2017) <http://www.daimler.com>
- (URL9 2017) <http://www.ashokleyland.com>



Automotive E/E and Automotive Software Technology

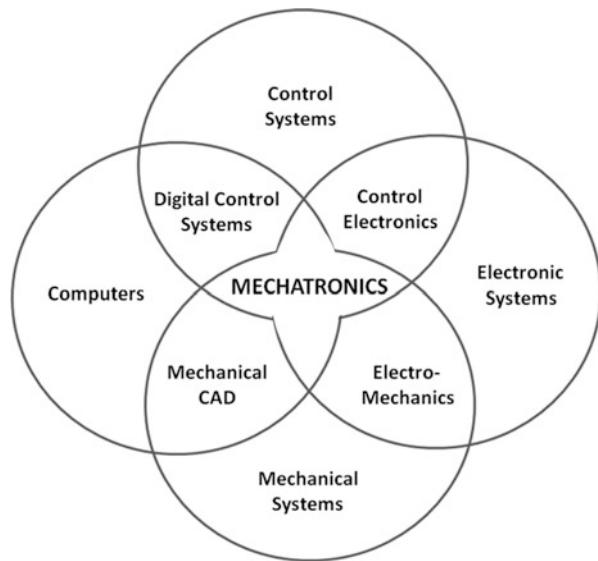
4

This chapter begins with an overview of mechatronic systems in the automotive domain in Sect. 4.1. Section 4.2 focuses on automotive electronics, taking into account body, chassis, comfort, driver assistance electronics, electronic control units (ECUs), and entertainment/infotainment electronics, as well as sensor technology. In Sect. 4.3, electrical and electronic (E/E) architectures and bus system requirements are introduced, with emphasis on disciplined approaches to their design. Section 4.4 discusses the concept of functional safety. Thereafter, Sect. 4.5 focuses on automotive software engineering, taking into account the increasing role of software content and product complexity, model-based software development, and hardware-in-the-loop (HIL) tests. Section 4.6 refers to the AUTomotive Open System Architecture (AUTOSAR) platform and Sect. 4.7 to the AUTOSAR Adaptive Platform. In Sect. 4.8, the GENIVI Alliance®, essential for telematic and infotainment components, is introduced. Section 4.9 provides examples of advanced driver assistance systems (ADASs) (see also Chap. 11), and Sect. 4.10 looks at future trends. Section 4.11 contains a comprehensive set of questions on automotive E/E and automotive software engineering, and finally followed by references and suggestions for further reading.

4.1 Mechatronic Systems in the Car

Mechatronics is an interdisciplinary engineering concept that synergistically combines mechanics, electronics, and software technologies in an integrated and optimized product design. The name itself was first used in Japan in the 1960s. Today, mechatronics is widely recognized as an engineering discipline with different application domains, as shown in Fig. 4.1. Mechatronics has huge potential in the automotive industry market as its products are upgraded and differentiated through innovation. For instance, mechatronic product features, such as power seats, electronic mirrors, automatic climate control, and others, facilitate memory functions

Fig. 4.1 Interdisciplinary disciplines of mechatronics



and automatic actuation in these products. Hence, mechatronics can be seen as the engineering answer to the manifold innovative demands of the automotive market which deal with the analysis, design, implementation, and test of electromechanical systems that are controlled by electronics and embedded software, such as antilock braking, power train control, door locking mechanism, suspension, and others.

As can be seen in Fig. 4.1, there are some fundamental trends that have significantly shaped the work of mechatronics engineers. These are primarily driven by advances in networking, modeling, simulation, and control, as well as embedded system design. The three major trends in mechatronics today are increasing software content, networked systems, and intelligent controls.

- *Rapid Increase of Software Content:* There is an exponential increase of embedded software in vehicles' ECUs. This stems from the increased complexity of the underlying control mechanisms but is also indicative of a trend of more development in software than in hardware, as shown in Fig. 4.2. This is possible because of tremendous improvement in the performance of microcontrollers. The advantage provided by this increase in software content is that software-based algorithms can be modified much easier, final parameterization can be done, and updates can be easily provided when necessary. The growth of software content, as shown in Fig. 4.2, has also increased the role of software engineering in mechatronics (Schäuffele and Zurawka 2016).
- *Intelligent Control:* Intelligent control provides adaptive and intelligent learning methods that supplement classical control techniques based on a mathematical model of the respective system. These techniques include a huge range of methodologies, from fuzzy controllers, neural networks, and expert-system-

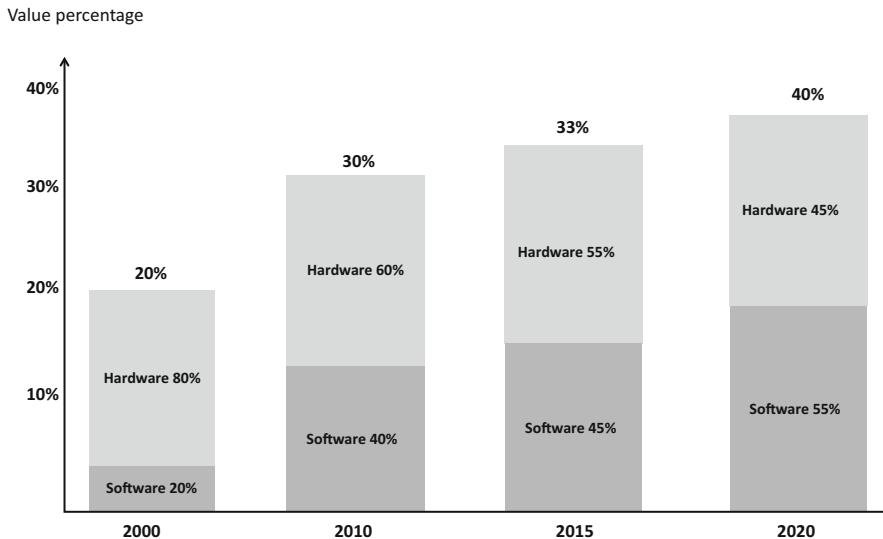


Fig. 4.2 Value percentage of Automotive E/E and Software in a vehicle (see also Mercer Management Consulting (2001) and (URL33 2017))

based control to hybrid and heterogeneous control techniques. Intelligent control is an important part of mechatronics as these systems often exhibit increased complexity. As the dynamics of a mechatronic system are often operating regime dependent, changes with external triggers sport a highly nonlinear behavior or need to adapt to various use case scenarios and nonstandard modeling techniques; and intelligent control methodologies become increasingly important.

- *Cyber-Physical Systems:* There is an increasing trend of connectivity in today's automotive systems which can be achieved by cyber-physical systems (see Sect. 5.1 in Chap. 5). These systems use computations and communication deeply embedded in and interacting with physical automotive systems by adding new capabilities to these systems. Cyber-physical systems (CPS) are networked systems which usually use open network technologies, such as the Internet. With the advent of the Internet of Things (IoT), sensors began to be networked, and mechatronics systems communicate with each other and backend systems. In this regard, computers and communication have become the universal system integrators that keep large systems together and which enable the composition of CPS infrastructure. Thus, CPS have an advanced and complex system architecture that connects computing, networking, and the physical and cyber, or virtual, environment within one paradigm, and therefore require a security design. CPS provide services such as:

- Control
- Information feedback
- Real-time monitoring

Most of the essential actions merge the interaction of the physical and the cyber worlds by integration and collaboration of computation, communication, and

control (3C) (Ning 2013; Möller 2016). Such networked systems can be easily monitored remotely; however, they are also exposed to cybersecurity risks (see Chap. 6).

4.2 Automotive Electronics

Automotive electronics deals with any electrical system or component used in vehicles. The number of these systems has increased rapidly and continuously during recent years. On the one hand, many new sensors and actuators and, therefore, new specific ECUs (Sects. 4.2.5) have been developed to make drivers and passengers feel safer. Electronic control units are modular devices, and a modern vehicle can have from 45 to more than 120 embedded ECUs. On the other hand, entertainment and navigation systems have made their way into vehicles to make travel more comfortable.

Automotive E/E systems and components are distributed according to different automotive E/E application domains which can be divided into the major categories of:

- Body electronics
- Communication and entertainment systems
- Power control
- Safety control

At present, most body electronic products have entered the mature or recession period of the product life cycle. For the power controls category, the gross domestic income (GDI) is still growing; and its penetration rate in the European market is expected to increase to around 40% by 2018. With regard to the safety controls category, the assistance systems for safe driving are experiencing a rapid development and growth phase as they are moving from high-end cars to the mid-range. Future large-scale adoption depends on the maturity of the technology and a decline in costs. However, due to the lack of mandatory rules of law, it is still in the introduction period in emerging markets such as China. Recently, original equipment manufacturers (OEMs) have focused on new communications and entertainment systems.

Embedded in-vehicle information systems have seen rapid development as Europe and the USA require new cars to be equipped with emergency in-vehicle information systems. In-vehicle information systems will be more widely adopted in the mass market in the future. In general, automotive electronics can be divided into the following domains:

- Body electronics
- Chassis electronics

- Comfort electronics
- Driver assistance and advanced driver assistance electronics
- Electronic control units
- Entertainment/infotainment
- Sensor technology

The requirement for increased computing performance for automotive electronic devices is also accompanied by the driving need for high bandwidth in the vehicle network. In addition, a vast number of sensors, actuators, and motors in a multitude of vehicle control applications are being deployed.

Sensors in the automotive domain are essential to measure gases such as:

- CO_x
- NO_x

They also measure:

- Speed
- Temperature
- Tire pressure
- Torque
- Vibration
- Yaw
- Other parameters helping to improve vehicle efficiency and safety

The actuators include:

- Drive pumps
- Electric motor drives controlled by engine control modules (ECMs)
- Fans
- Heating, ventilation, and air control (HVAC)
- Relays
- Solenoids
- Sunroofs
- Window lifts

All of these are essential to making vehicles more comfortable. The many electronic devices embedded in today's vehicles increase power consumption on one hand, based on the direct relationship of the weight of added hardware through the automotive electronic components and lower fuel economy. On the other hand, the need for more computing performance for these components, embedded memory capacity, and higher-bandwidth connectivity are each responsible for consuming additional power. In general, it can be stated that every 100 W of electrical power used requires approximately 0.1 l of gas/100 km or 0.1 l of gas/62 miles.

Every additional 50 kg of vehicle weight is responsible for approx. 0.15 l of gas/100 km or 0.15 l of gas/62 miles of increased fuel consumption. Thus, with the ever-growing levels of automotive electronic systems for comfort, efficiency, and safety, automakers and their electronic suppliers must cope with the conflicting demands of automotive electronics complexity versus vehicle power requirements and weight.

In addition to the greater computing and networking performance of automotive electronics, there is a major push for increased safety and security within the vehicle's body network to cope with the growing complexity of automotive electronics and the critical nature of the functions they enable.

- **Safety:** Safe comes from the Latin *salvus*, which means “uninjured” or “healthy.” Safety studies and the practice of designing safe vehicles and equipment and complying with regulations all help with the goal of minimizing the occurrence and consequences of traffic collisions.
- **Security:** Secure comes from the Latin *securus*, which means “with care.” Thus, security refers to a condition of being protected from or not exposed to danger. In the case of a vehicle, it offers collaborative opportunities for all stakeholders in the automotive industry to mitigate the risk of cyberattacks.

Nevertheless, the terms differ in connotation with regard to context when deciding which of them to use. However, as wireless communication for vehicles becomes more widespread, there is an ever-growing need for security in the automotive domain to prevent unauthorized access of the vehicle network for cyberattacks (see Chap. 6).

With regard to the advances in innovative automotive functionalities, the subtopics in automotive E/E can be summarized as shown in Fig. 4.3.

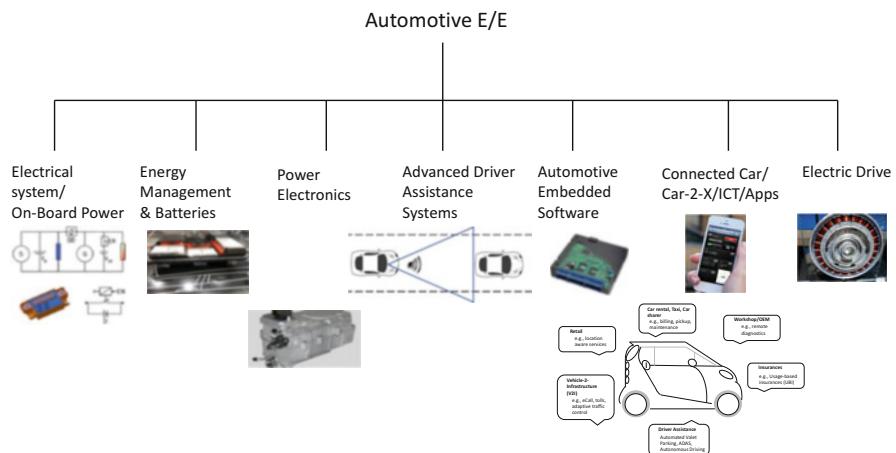


Fig. 4.3 Classification of subtopics in automotive E/E

4.2.1 Body Electronics

Body electronics are an important property for the acceptance of a vehicle with regard to functions such as:

- Central locking
- Lighting
- Window cleaning

Customers are very aware of all of these functions. Unfortunately, a well-engineered locking system does not persuade a customer to buy a particular vehicle brand; but a poorly designed one will definitely deter a customer from purchasing that brand. However, intuitive, logical, and clearly structured body electronic systems leave the customer with the impression of having a high-quality vehicle brand; but for the automaker, it is more difficult to develop systems with higher safety standards for an increasing number of vehicle variants and configurations. Hence, body electronics physical hardware is very important for customer because they are looking for new levels of comfort, efficiency, safety, and other features in their vehicle. Thus, body electronic systems are being developed for many different functionality, such as:

- *Body Control Module (BCM)*: The BCM is responsible for monitoring and controlling various automotive electronic accessories in today's vehicle body. A BCM controls a number of devices, such as:
 - Air conditioning
 - Central locking
 - Immobilizer system
 - Power mirrors
 - Power windows

Furthermore, the BCM communicates with system ECUs through the vehicle's controller area network (CAN) or local interconnect network (LIN) bus (see Sect. 4.3.3). Its main application is controlling load drivers and actuating relays that in turn perform functions in the vehicle, such as locking the doors, dimming the in-vehicle overhead lamp, and other essential features (URL1 2017).

- *Power Management Module (PMM)*: A vehicle PMM provides electronically switched power to the manifold E/E systems in a vehicle, such as data acquisition systems and automotive electronic devices, such as ECUs, lights, motors, solenoids, and others, and can be controlled through a combination of switch inputs, CAN messages (see Sects. 4.3.3 and 6.2.1), and logic functions. PMM reduces fuel consumption, maintenance costs, and repairs through:
 - Identification of degrading loads based on power usage or other parameters
 - Load prioritization/reconstitution/reduction/shedding
 - Power optimization of electrical loads or subsystems

Therefore, basic concepts of vehicle power management, as described in (Zhang and Mi 2011), include:

- Control device
 - Data acquisition device
 - Drive cycle selection device
 - Fuel consumption and performance device
 - Fuel demand in drive cycles device
 - Monitoring device
- *Power Window and Door Control (PWDC)*: The PWDC contains, within a vehicle's door, the automotive electronics used to drive all loads. It is connected via a LIN bus (see Sect. 4.3.3) to the dashboard, an easy-to-read, real-time user interface showing a graphical presentation of the current status and historical trends of the automotive electronic system, such as the vehicle's doors in conjunction with the vehicle's door module actuators, as shown in Table 4.1 for the ST Microelectronic microcontroller ST72F561, designed for midrange applications with CAN and LIN interfaces, based on an industry standard 8-bit microcontroller core, featuring an enhanced instruction set (ST Microelectronics 2013),

Table 4.1 Door module actuators

Actuators	
Door lock	1 DC motor
Mirror axis control	2 DC motors
Mirror defroster	1 DC motor
Mirror fold	1 grounded resistive load
Light bulbs	1 grounded resistive load
Window lift	1 DC motor

The following respective functions are realized through truth tables, an example of which is shown in Table 4.2 for the use case of door and window lock coding, as described in the *ST Microelectronics Application Note AN2334* (ST Microelectronics 2013):

- Light bulbs
- Mirrors
- Vehicle doors
- Windows

Vehicle doors are wired in many different ways, depending on which features are incorporated, such as the one that allows the driver to control all four windows on the vehicle and to lock out the controls on the other three individual windows. In this system, the power comes into the window switch control panel on the door and is distributed to a contact in the center of each of the four window switches. Two contacts, one on either side of the power contact, are connected to the vehicle ground and to the electric motor. The power also runs through the lockout switch to a similar window switch on each of the other doors.

Table 4.2 Window and door lock coding

Window				Door
Up left	Down left	Up right	Down right	Lock/Unlock
0	0	0	0	0
0	0	0	1	0
0	1	0	1	0
0	1	0	0	0
0	0	1	0	0
1	0	1	0	0
1	0	0	0	0
0	0	0	0	1
0	0	0	1	1
0	1	0	1	1
0	1	0	0	1
0	0	1	0	1
1	0	1	0	1
1	0	0	0	1

- *Remote Keyless Entry (RKE)*: Automotive electronic remote central locking controls access to a vehicle and is activated by a handheld device performing the function of a standard car key without physical contact. RKE can include two functions: (i) remote keyless entry system which unlocks the doors and (ii) remote keyless ignition system (RKI) which starts the engine. The keyless entry system was originally a lock controlled by a keypad located at or near the driver's door that required pressing a predetermined numeric code for entry. These systems have evolved into a hidden touch-activated keypad today (URL2 2017).
- *Seat Comfort (SC)*: Smooth ride quality is an important issue, since a rough or bouncy ride can exacerbate back pain. Therefore, SC should focus on multicontour driver and front passenger seats which contain inflatable air chambers enabling them to adapt to the user's individual anatomy to give real orthopedic support. For the most part, these seats are developed by orthopedic specialists to provide outstanding comfort and support driver fitness, especially on long journeys.
- *Smart Mirrors and Wipers (SMW)*: For information on smart mirrors, see PWDC in this section. Smart wipers that activate themselves when it rains and adjust their speed when rain gets heavier or lighter weren't always commonplace. Rain can be sensed by a module mounted on the inside of the windshield, behind the rearview mirror. It contains light-emitting diodes (LEDs) and a set of light collectors. When the weather is dry, the LED light bounces off the windshield and into the collectors. When a raindrop comes down in front of the module, some of the light is refracted away from the collectors; and the system triggers the wiper blades to swipe.
- *Sunroof (SHD)*: The SHD is controlled by the sunroof control module which contains all of the load circuits and is directly connected to the sunroof drive. The

module is allocated to the vehicle by means of encoding. It contains the following components:

- DC motor with attached step-down gear mechanism
- Two position-detecting hall effect sensors
- Electronic control switches

The two hall sensors register the number of DC motor revolutions and thus determine the position of the sunroof. The drive is switched off and on by reaching the relevant end position. The torque of the drive is constantly calculated from the pulses sent by the position sensors and the power consumption of the motor. Torque increasing to above a certain value is interpreted as trapping. Characteristic data for the antitrapping protection are defined in the coding data. The data are written into the control unit during the encoding procedure. Antitrapping protection is active in the close direction when the sunroof is open between > 4 and < 200 mm. This function is active during both normal closing and automatic operation and convenience closing of the sunroof. Antitrapping protection is deactivated in case of a fault by overpressing the switch in the close and hold direction. The closing procedure is interrupted if trapping is detected and the sunroof is opened for approximately 1 s ([URL3 2017](#)).

4.2.2 Chassis Electronics

The chassis system has a lot of subsystems which monitor various parameters and are actively controlled, such as:

- *Antilock Braking System (ABS)*: This safety system allows the vehicle's tires to maintain tractive contact with the road surface with regard to driver inputs while braking, preventing the wheels from locking up and avoiding uncontrolled sideslips. An ABS generally offers improved vehicle control and decreases stopping distances on dry and slippery surfaces; however, on loose gravel or snow-covered surfaces, ABS can significantly increase braking while still improving vehicle control. An ABS typically includes a central ECU, four wheel speed sensors, and two hydraulic valves within the brake hydraulics. The ECU constantly monitors the rotational speed of each wheel. If the ECU detects that a wheel is rotating significantly slower than the others, the ECU actuates the valves to reduce hydraulic pressure to the brake at the affected wheel, which reduces the braking force on that wheel and the wheel then turns faster. If the ECU detects that a wheel is turning significantly faster than the others, the brake hydraulic pressure to this wheel is increased so that the braking force is reapplied, slowing down the wheel. This process is repeated continuously. Thus, the wheels of cars equipped with ABS are practically impossible to lock even during panic braking in extreme conditions. Recent versions not only prevent wheel lock while braking but also electronically control the front-to-rear brake bias. This function,

depending on its specific capabilities and implementation, is known as electronic brake force distribution (EBD) traction control, emergency brake assist (EBA), or electronic stability program (ESP) (URL4 2017).

- *Electronic Brake Distribution:* This brake technology automatically varies the amount of force applied to each of a vehicle's wheels, based on road conditions, speed, loading, etc. It can apply more or less braking pressure to each wheel in order to maximize stopping power while maintaining vehicular control. Typically, the front end carries the most weight, and EBD distributes less braking pressure to the rear brakes so that the rear brakes do not lock up and cause a skid. In some systems, EBD distributes more braking pressure at the rear brakes during initial brake application before the effects of weight transfer become apparent (URL5 2017).
- *Electronic Stability Control (ESC)/Electronic Stability Program:* This electronic system improves a vehicle's stability by detecting and reducing loss of traction. If ESP detects a loss of steering control, it automatically applies the brakes, helping to steer the vehicle where the driver intends to go. Braking is automatically applied to wheels individually, such as the outer front tire to counter oversteering or the inner rear wheel to counter understeering. Some ESP systems also reduce engine power until control is regained. ESP does not improve a vehicle's cornering performance; instead, it helps to minimize the loss of control (URL6 2017) (see also Sect. 4.2.5).
- *Traction Control System (TCS):* The TCS is a function of the ESP to prevent loss of traction. A TCS is activated when throttle input and engine torque are mismatched with regard to road surface conditions. The traction control system splits up the electrohydraulic brake actuator and wheel speed sensors with ABS.

Another type of automotive chassis electronic subsystem are the passive safety systems (PSSs). These systems are always ready to react when there is a collision in progress or to prevent a collision when it identifies a critical or dangerous situation. To these systems belong:

- *Airbag Control System (ACS):* This safety system is designed to inflate the airbags very rapidly and then quickly deflate them during a collision in case of an impact with another object or a rapid sudden deceleration. The purpose of the ACS is to provide the driver and passengers with a soft cushioning and restraint during a crash event to prevent any large forces between the crashing driver and passengers and the interior of the vehicle. The airbag provides an energy-absorbing surface between the vehicle's driver and the steering wheel, the instrument panel, or the A-B-C structural body frame pillars, as well as the headliner and windshield/windscreen (URL7 2017).
- *Hill Descent Control (HDC):* The HDC system allows a smooth and controlled hill descent in rough terrain without the driver needing to touch the brake pedal. When the vehicle descends, the ABS system takes control of each wheel's speed. If the vehicle accelerates without driver input, the HDC will automatically apply the brakes to slow the vehicle down to the desired speed. With HDC, drivers can

be confident that even the ride down hills with slippery or rough terrain will be smooth and controlled and that they will be able to maintain control as long as sufficient traction exists (URL8 2017).

4.2.3 Comfort Electronics

Comfort electronics are automotive electronic systems that make a ride comfortable for the driver and the passengers like:

- *Automatic Climate Control*: Regulate cabin temperature and ventilation taking into account outdoor temperature, sun intensity, and cabin temperature with regard to the driver's or passengers' requests.
- *Electronic Seat Adjustment with Memory*: Keeps track of the user's seat position and stores the user's individual seat settings for the driver's seat and the front passenger's seat. Electronic seat adjustment with memory (ESAM) is available mostly in conjunction with memory exterior mirrors whose settings are also retrieved from the memory with regard to different drivers' or passengers' individual seat settings.
- *Automatic Wipers* (see Sect. 4.2.1): Activate themselves when it rains and adjust their speed when the rain gets heavier or lighter by detecting the amount of water on the windshield and controlling the wipers.
- *Automatic Headlamps—Adjusts Beam Automatically*: Activate through a photovoltaic sensor which is embedded into the instrument panel. The sensitivity of the sensor is either set by the automakers or the driver and is activated by the lighting conditions at dawn or dusk. The lights may switch off up to a couple of minutes after the engine has been turned off.
- *Automatic Cooling—Temperature Adjustment*: Maintain a constant temperature inside the vehicle. This requires the AC to regulate the inside temperature by an automatic control system using ambient air temperature sensors outside the passenger compartment. This type of sensor can be one or more in-vehicle air temperature sensors which also may include an infrared sensor that measures the actual body temperature of the driver and the passengers, a sunload sensor to compensate for sunlight entering the vehicle through the vehicle's windows, one or more electronic control modules, and electronic controls for the various heating, ventilation, and airflow control HVAC outlets. The controller keeps track of their position, running the motors full open and full closed and then counting the revolutions of the motor armature to determine their exact position (AA1Car 2016).

4.2.4 Driver Assistance Electronics

Driver assistance systems (DAS) are automotive electronic components developed to assist the driver in the driving process and to enhance vehicle systems for safety and better driving. Safety features are designed to avoid collisions and accidents by

offering technologies that alert the driver to potential critical situations or to avoid collisions by implementing safeguards and taking over control of the vehicle, in effect bailing out the driver. The advanced driver assistance systems (ADAS) are a collection of systems and subsystems that finally result in a fully automated vehicle (see Sect. 4.9). The benefits of ADAS are potentially considerable because of a significant decrease in driver suffering, economic cost, and pollution. However, there are also potential problems to be expected, since the task of driving an ordinary vehicle is changing and moving in the direction of supervising an automated moving vehicle.

There are many different kinds of DAS automotive electronic components available. Some of them are built into vehicles or are available as add-on packages. A DAS relies on input from multiple data sources, including automotive imaging, computer vision, image processing, in-vehicle networking, light imaging detection and ranging (LiDAR), and radar. LiDAR is a surveying technology that measures distance by illuminating a target through laser light. Future autonomous vehicles will use LiDAR for obstacle detection and safe avoidance navigation through environments using rotating laser beams.

DAS, as well as ADAS, are fast-growing segments in automotive electronics, with steadily increasing rates of adoption of industry-wide quality standards in vehicular safety systems following the ISO 26262 for functional safety of automotive E/E systems defined by the International Organization for Standards (ISO) in 2011.

Next generation DAS will increasingly leverage wireless network connectivity to offer improved value by using vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) data using wireless fidelity (Wi-Fi®) data network systems (URL9 2017). V2V is a communication technology allowing vehicles to communicate with each other by Wi-Fi. V2V is also known as the vehicular ad hoc network (VANET), a variation of the mobile ad hoc network (MANET). V2I communication, commonly called vehicle-to-X (V2X), is the wireless exchange of critical safety and operational data between vehicles and road infrastructure intended to avoid or mitigate vehicle accidents but also to enable a wide range of safety, mobility, and environmental benefits.

Wi-Fi is the name of a wireless network technology that provides high-speed Internet and network connections based on the IEEE 802.11 standards. In addition, Wi-Fi is a registered trademark of the Wi-Fi Alliance, an organization made up of leading wireless equipment and software providers, certifying all 802.11-based products for interoperability, promoting the term as a global brand name across all marked 802.11-based wireless local area network (LAN) products.

Despite progress in using DAS over the past three decades, drunk driving claims a huge number of lives each year. Thus, the Driver Alcohol Detection System for Safety (DADSS) feature has been developed as an additional DAS measure and based on a technology which automatically detects when a driver is intoxicated with a blood alcohol concentration (BAC) at a certain breath alcohol level above a prescribed amount and prevents the vehicle from moving. There are two systems available for DADSS, the breath-based system and the touch-based system

(DADSS 2016). DADSS is supported by a broad coalition of organizations including automakers, safety and child advocates, bipartisan leaders in the US Congress and other government entities, and members of the medical community.

The most used DAS and ADAS components are:

- *Adaptive Cruise Control (ACC)*: ACC is an intelligent form of cruise control that slows down and speeds up automatically to keep pace with the vehicle in front. The driver sets the driving speed; and the ACC radar sensor measures the speed ahead, monitors a vehicle in the lane, and instructs the following vehicle to stay a couple of seconds behind the vehicle ahead. The number of seconds to stay behind the vehicle ahead depends on the following driver's cruise control system settings. A schematic sketch is shown in Fig. 4.4.
- *Adaptive High Beam Assist (AHA)*: A headlight control strategy that continuously and automatically tailors the headlamp range so that the beam only reaches other vehicles ahead, AHA always ensures a maximum possible sight range without causing glare to other users on the road. The range of the beam can vary between 65 and 300 m, depending on traffic conditions.
- *Automated Parking Assist (APA)*: APA is a maneuvering system that moves a vehicle from a traffic lane into a parking spot. The parking maneuver is achieved by means of coordinated control of the steering wheel angle and speed taking into account the actual surrounding environment using various sensor-based methods to detect objects around the vehicle. A signal is emitted that is reflected back when an obstacle is encountered near the vehicle, ensuring collision-free maneuvering within the available space.
- *Automotive Navigation System (see Sect. 4.2.6), Global Positioning System (GPS), and Traffic Message Channel (TMC)*: GPS is a US-owned utility that provides

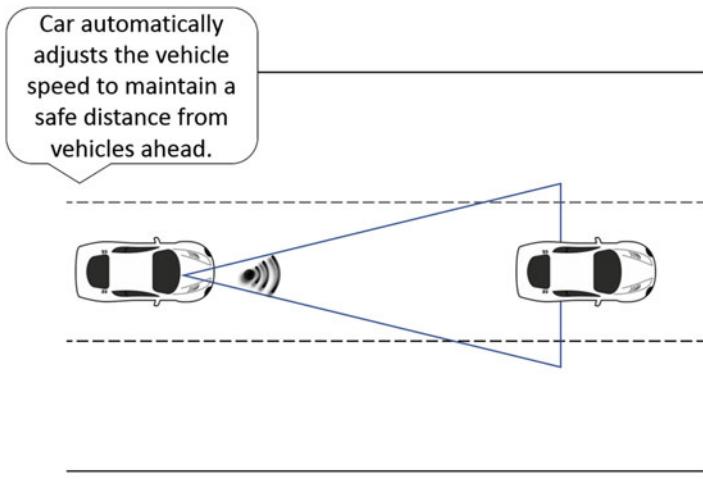


Fig. 4.4 Adaptive cruise control where the left vehicle in the bottom lane automatically follows the right vehicle at a safe distance

users with positioning, navigation, and timing services (PNT). It consists of three segments: space, control, and user. TMC delivers up-to-date traffic and travel information to vehicle drivers. It uses the ALERT C coding protocol to send messages via the Radio Data System (RDS) communication protocol standard embedding digital information in conventional FM radio broadcasts.

- *Collision Avoidance System (Pre-crash System)*: Once an imminent crash is detected, the system either warns the driver that a collision is imminent or takes action autonomously without any driver input, e.g., by braking or steering or both. Collision avoidance by braking is appropriate at low vehicle speeds, e.g., below 50 km/h/31 mph, while collision avoidance by steering is appropriate at higher vehicle speeds.
- *Crosswind Stabilization (CS)*: Sensor-based systems detect forces acting on vehicles through side-wind gusts, such as when driving across a bridge or overtaking trucks. A crosswind stabilization system response takes into account:
 - Steering characteristics of the driver
 - Vehicle load
 - Vehicle speedAdvanced crosswind stabilization regulates the suspension force according to the strength of the crosswind to reduce vehicle body oscillations.
- *Driver Drowsiness Detection (DDD)*: This system helps prevent accidents caused by the driver getting drowsy. Various technologies are used:
 - Monitors the steering pattern using steering input from the electric power steering system
 - Monitors the vehicle's position in a lane using lane monitoring camera data
 - Monitors the driver's eyes/face by using a camera to detect the driver's face and eyeblink data
 - Uses body sensors to physiologically measure parameters such as brain activity, heart rate, skin conductance, and muscle activity
- *Electric Vehicle Warning Sounds for Hybrids and Electric Vehicles (EVWS)*: This system is designed to alert pedestrians to the presence of electric vehicles traveling at low speed. Warning sound devices are necessary because electric vehicles produce less noise than traditional combustion engine vehicles which can make it more difficult for pedestrians, the visually impaired, cyclists, and others to be aware of their presence. Warning sounds may be driver-activated electric warning systems or automatic systems at low speeds.
- *Emergency Assist (EA)*: Emergency assist monitors driver behavior. In case of a medical emergency where the driver is no longer able to safely drive the vehicle, the vehicle takes control of the brakes and the steering until the vehicle comes to a complete stop.
- *Intersection Assistance (IA)*: The IA system monitors cross-traffic in an intersection/road junction. If the system detects a hazardous situation, it prompts the driver to start emergency braking by activating visual and acoustic warnings and automatically engaging the brakes.
- *Hill Descent Control (HDC)*: Hill descent control allows a smooth and controlled hill descent in rough terrain without the driver needing to touch the brake pedal (see Sect. 4.2.2).

- *Lane Departure Warning (LDW)*: This feature is designed to warn the driver when the vehicle begins to move out of its lane, unless a turn signal is on in that direction.
- *Parking Sensor (PS)*: A parking sensor system is designed to alert the driver to obstacles while parking.
- *Traffic Sign Recognition (TSR)*: This system is designed to recognize traffic signs on the road.
- *Vehicular Communication Systems (VCS)*: Vehicular communication systems are networks in which vehicles and roadside units are communication nodes providing each other with information, such as safety warnings and traffic information. VCS is part of intelligent transportation information systems (ITIS).
- *Wrong-Way Driving Warning (WWDW)*: This system emits an acoustic warning, together with a visual warning, which helps to prevent serious accidents caused by wrong-way drivers.

4.2.5 Electronic Control Units

An electric control unit (ECU) is basically made up of hardware and software where the software is a specific firmware based on the ECU's specific functionality. The hardware is mostly made up of various electronic components on a printed circuit board (PCB), with microcontroller chip(s) being the most important hardware component together with the following, depending on the ECU's properties:

- *Erasable Programmable Read-Only Memory (EPROM)*: Can retrieve stored data after its power supply has been turned off and back on; memory can be erased by exposure to a strong ultraviolet light source.
- *Flash Memory Chip*: Solid-state memory medium.
- *Electronic Solid-State Nonvolatile Storage Medium*: Can be electrically erased and reprogrammed.
- Other electronic hardware components.

The software (firmware) can be a set of lower- or higher-level codes that run the specific functionality on a microcontroller. An ECU can be characterized by its:

- Analog and/or digital inputs and outputs (I/O)
- Communication interface adapters
- Communication protocols
- Power device interface/control
- Switching matrices for low- and high-power signals

ECUs have different characterizing names based on their primary functionality. One of the most demanding ECUs in a vehicle is the engine control module (ECM). The main ECM function is to get information from sensors and, by running a certain

actuator with the computed sensor information, to adjust their settings. This allows the vehicle to run in accordance with the driver's behavior. Connected sensors which play an important role in the ECM are:

- Absolute pressure sensor
- Air temperature sensor
- Camshaft sensor
- Crankshaft sensor
- Engine coolant temperature sensor
- Idle air controller (an actuator not a sensor)
- Knock sensor
- Mass air flow sensor
- Oxygen sensor
 - Zirconia oxygen sensor
 - Titania oxygen sensor
- Throttle position sensor

Because of the many sensors that measure/monitor the following in real time, as well as other parameters at different points within the engine, the ECM control demand is one with the highest real-time constraints, the so-called hard real-time requirement. Sensors measure/monitor:

- Engine speed
- Flow
- Nitrogen oxide (NO_x) level
- Oxygen level
- Pressure
- Temperature

All sensor information is sent to the ECM, which has logic circuits for doing the actual controlling. The ECM output is connected to different actuators, e.g., the throttle valve; the exhaust gas recirculation (EGR) valve; the fuel injector, which uses a pulse-width modulated (PWM) signal, dosing the injector; and more. Of all of the automotive electronics in any vehicle, the computing power of the ECM, typically a 32-bit microprocessor is among the highest.

Another important ECU is the transmission control module (TCM), which monitors/controls the transmission system, mainly gear shifting for better shift comfort, and lowers torque that is interrupted while shifting. An automatic transmission uses controls for its operation. Many semiautomatic transmissions also have fully or semiautomatic clutches. The ECM and TCM exchange messages, sensor signals, and control signals for their operation.

The vehicle control module (VCM) is connected to various kinds of sensors to monitor/control several systems in the vehicle. The VCM receives inputs from crash

sensors (accelerometers) and sensors that detect the following to determine the force with which the frontal air bags should deploy:

- Seat belt use
- Seating position
- Seat position
- User's weight

Furthermore, the VCM take input from the following sensors to provide an output to the electronic stability control (ESC) for the safest driving situation.

- *Lateral Acceleration Sensors (LAS)*: Detect lateral acceleration of vehicles perpendicular to the direction of travel. It becomes noticeable as centrifugal force moves the vehicle to the outside of a curve when cornering. Lateral acceleration sensors are part of the electronic stabilization program.
- *Steering Wheel Angle Sensors (SWAS)*: Measure steering wheel angle and velocity through the entire range of the steering wheel; merges optical and magnetic principles with advanced underlying software representing a mechatronic component. Vital part of the electronic stability control system.
- *Wheel Speed Sensors (WSS)*: Provide essential wheel speed information for ABS and for traction control and stability control.
- *Yaw-Rate Sensors (YRS)*: Gyroscopic device that measures a vehicle's angular velocity around the vehicle's vertical axis.

Hence, the main systems the VCM takes care of are:

- *Adaptive Cruise Control*: Automatically adjusts a vehicle's speed to maintain a safe distance from the vehicle ahead, see Sect. 4.2.4.
- *Airbag Control System*: Control system that detects and evaluates a crash before activating the appropriate restraint systems based on the type of collision and its severity, see Sect. 4.2.2.
- *Electronic Power Steering (EPS)*: Helps drivers to steer by augmenting the steering force of the steering wheel by using an actuator and a hydraulic cylinder that is part of a servo system. An EPS with no mechanical connection is called "steer-by-wire." In this context, wire refers to electrical cables that carry power and data, not thin-wire-rope mechanical control cables (URL10 2017).
- *Electronic Stability Control, Also Referred to as Electronic Stability Program*: Improves vehicle stability by detecting and reducing loss of traction (see Sect. 4.2.2).

4.2.6 Entertainment/Infotainment Electronics

Entertainment/infotainment systems are manufacturer specific, designed automotive electronic components for which different tools are used for hardware and software

development. They are primarily developed by OEMs and/or third-party suppliers. The main types of entertainment/infotainment systems are:

- *Navigation Systems:* Are entirely on board a vehicle or located elsewhere and communicate via radio or other signals with a vehicle or use a combination of these methods:
 - Contain maps displayed in human readable format via text or in a graphical format
 - Determine vehicle's location via sensors, maps, or information from external sources
 - Provide suggested directions to a vehicle driver by text or voice
 - Provide directions directly to a connected car or an autonomous vehicle
 - Provide information on traffic conditions and suggest alternative directions
- *Vehicle Audio Systems:* Provide in-vehicle entertainment and information such as:
 - Navigation systems
 - Bluetooth® telephone integration
 - Smartphone controllers, such as CarPlay®, an Apple standard enabling a car radio or head unit to be a display and controller for an iPhone®, and Android™ Auto, a smartphone standard developed by Google allowing operation in vehicles through the dashboard's head unit

Operated from the dashboard, these systems can be controlled by the steering wheel controls of the vehicle's audio system and simple voice commands to initiate phone calls, select radio stations, or play music from an MP3 player or other embedded devices.

- *In-Vehicle Infotainment (IVI):* Hardware and software in vehicles that provide audio or video entertainment. In-vehicle entertainment originated with vehicle audio systems based on radio, cassette, and/or CD players and automotive navigation systems based on:
 - *Bluetooth and USB Connectivity:* Bluetooth was developed in the 1990s by the Bluetooth Special Interest Group as an industrial standard with regard to IEEE 802.15.1 for data transmission by radio waves between systems over a short distance. Bluetooth allows building up point-to-point, ad hoc, and piconet connections. A universal serial bus (USB) connects computers and peripherals and is an industrial standard developed by Compaq, Hewlett-Packard, Intel, Lucent, Microsoft, NEC, and Philips. It provides a ubiquitous link that can be used across a wide range of PC-to-telephone interconnections.
 - *Carputer:* Computer with specific features, such as compact size, low-power requirement, and some customized components.
 - *In-Vehicle Internet:* Provided by tethering the Internet connection of a phone or tablet with other devices, such as laptops, or with a mobile hotspot, a physical location where the driver may obtain Internet access, whether portable or built into the vehicle.
 - *Video Player:* Hardware device to watch online video or view video files saved locally. Notable brands are Windows® Media Player and VLC media player.

- *Wi-Fi*: Technology that allows automotive electronic devices to connect to a wireless local area network (WLAN). A WLAN is usually password protected but may be open, which allows any device within its range to access the resources of the WLAN network.

Once controlled by simple dashboard knobs and dials, IVI systems can include steering wheel audio controls and hands-free voice control.

4.2.7 Sensor Technology

A sensor is a device that generates a measurable signal in response to a stimulus. It is capable of converting any physical quantity to be measured into a signal which can be displayed, read, stored, or used to control some other quantity of interest. Sensors are used to measure a particular characteristic of any kind of object or device. With regard to its application domain, a sensor is developed based on the type of use. For example, a thermocouple sensor, an electrical device consisting of two dissimilar conductors forming electrical junctions at differing temperatures, can be used to sense heat energy (temperature) at one of its junctions and generate an equivalent output signal, e.g., a voltage, which can be read by a multimeter.

Sensors are classified based on the nature of the quantity they measure which results in different kinds of sensors and, thus, different kinds of signals, e.g., electric, mechanical, optical, and others. A lot of sensors generate electrical signals (usually voltages) which are analogous to the physical quantity to be measured. Often, the voltage is proportional to the measurand quantity. Then, the sensor voltage output generated can be described by:

$$V_{\text{sensor}} = K \cdot m$$

where V_{sensor} is the sensor voltage output generated by the sensor, K is the sensitivity constant of the sensor, and m is the measurand. The sensitivity of a sensor is an important characteristic indicating how much the sensor's output changes when the sensor's input quantity being measured changes. It is basically the slope:

$$\frac{\Delta y}{\Delta x}$$

with Δy as sensor output and Δx as sensor input for a linear characteristic.

The typical static characteristics a sensor should have are:

- *High Accuracy*: Indicates the correctness of the sensor output in comparison to the actual measured quantity.
- *High Precision*: Represents the capacity of a sensor system which gives the same reading for respective measures of a measurand under the same conditions; closely related to high repeatability which means that the sensor system generates the same response for successive measurements when all operative and environmental conditions remain constant.

- **High Resolution:** Indicates the smallest change in the sensor input signal that the sensor can detect.
- **High Sensitivity:** Indicates the ratio of an incremental change in the sensor's output, Δy , to the incremental change of the measurand in the sensor input signal, Δx . For example, if the sensor output voltage of a temperature sensor changes by 1 mV for every 1 °C change in temperature, then the sensitivity is 1 mV/°C.
An ideal sensor has a large and preferably constant sensitivity in its operating range. The operating range describes the measurement range of the sensor representing the minimum and maximum values of the measurement that can be measured with the sensor system. Outside the measurement range, the sensor can, e.g., reach a saturation state at which it can no longer respond to any changes
- **Less Noise and Disturbance:** The noise refers to the signal-to-noise ratio (S/N) comparing the level of a desired sensor signal to the level of its background noise. S/N is defined as the ratio of signal power to noise power as described in the following equation:

$$S/N = \frac{P_{\text{signal}}}{P_{\text{noise}}}$$

with P as average power. An S/N higher than 1:1 indicates a higher sensor signal than background noise. Both signals must be measured at the same or the equivalent measuring point in the system under test and within the same system bandwidth.

So far, the variance of the sensor signal and background noise is known; and if the signal is zero mean, then the following equation can be obtained:

$$S/N = \frac{\sigma_{\text{signal}}^2}{\sigma_{\text{noise}}^2}$$

$$S/N = \frac{P_{\text{signal}}}{P_{\text{noise}}} = \left(\frac{A_{\text{signal}}}{A_{\text{noise}}} \right)^2$$

If the sensor signal and the background noise are measured across the same impedance, then the S/N can be obtained by calculating the square of the amplitude ratio with A as root mean square (RMS) amplitude.

Disturbance refers to the impact to the measurement precision with regard to external or internal influences. Disturbance can be introduced as measurement error, defined as the difference between the true value of the quantity being measured and the actual value generated by the sensor. Hence, a measurement error can be defined as the real value of the sensor output of a measurement system minus the ideal value at the input of a measurement system according to:

$$e_{\text{ms}} = e_{\text{rum}} - e_{\text{itm}}$$

with e_{ms} as the measurement error, e_{rum} as the real untrue measurement value, and e_{itm} as the ideal true measurement value.

Since the measuring error can be caused by a variety of internal and external sources and is closely related to high accuracy, the absolute and relative error can be defined as follows:

$$\text{Absolute Error} = \text{Output} - \text{True Value}$$

$$\text{Relative Error} = \frac{\text{Output} - \text{True Error}}{\text{True Value}}$$

The absolute error has the same unit as the measured quantity; the relative error is unitless.

- *Less Power Consumption:* Minimized power source needs
- *Linearity:* Indicates that the sensor output signal changes linearly with the sensor input signal

Sensors are used to instrument and monitor a system or process, track assets through time and space, detect changes in the system or process which have been defined as being important, control a system or process with regard to being in closed vicinity within a defined range of change, and adapt services to improve their utility. Sensors are also used in everyday applications, such as touch-sensitive elevator buttons, lamps which dim or brighten by touching the base, and much more.

With advances in mechatronics and easy-to-use microcontroller platforms, the use of sensors has been expanded beyond the more traditional fields of measuring:

- Flow
- Pressure
- Temperature

Moreover, analog sensors, such as potentiometers, force-sensing resistors, and others are still widely used.

Sensors need to be designed to have little effect on the physical quantity measured, which requires that the sensor be manufactured to be very small and to consume less energy from the physical quantity measured to reduce measurement error. Sensors are hardware devices that range in scale from nano-sensors to macro-sensors. They act as data generators and pre- and/or post-processors of the data to be monitored. In the case of radio-frequency identification (RFID) sensors (see Sect. 5.2.2), the data processing is less complex as compared to other sensor types. The continuously produced analog signal x of the sensor is digitized into a proportional digital quantity by an analog-digital converter, and is sent to a microcontroller for further processing.

Sensors can be traditionally classified into the following categories:

- *Active Sensors:* Require continuous energy from a power source and sense data by actively probing the environment

- *Narrow-Beam Sensors*: Have a well-defined notion of the direction of measurement
- *Omnidirectional Sensors*: Have no notion of the direction involved in their measurements
- *Passive Sensors*: Self-powered and sense data without actually manipulating the environment by active probing

With the advent of new technologies, sensors are now manufactured on a microscopic scale, such as microelectromechanical systems (MEMS), a technology that in its most general form can be defined as miniaturized mechanical and electromechanical elements, devices, and structures, made using the techniques of microfabrication. They range in size from below 1 micron on the lower end of the dimensional spectrum to several millimeters and are fabricated as discrete devices or large arrays (Berlin and Gabriel 1997). MEMS devices can vary from relatively simple structures to extremely complex electromechanical systems under the control of integrated microelectronics. Moreover, MEMS perform two different types of functions: sensor and actuator. Both sensors and actuators act as transducers, converting one signal to another. Of specific interest are transducers that convert environmental information into digital signals and vice versa. Hence, MEMS sensors can convert environmental information, such as temperature, humidity, pressure, and more, into an electrical signal. In addition, MEMS actuators work in reverse to sensors; they convert an electrical signal into physical information to move or control devices, such as motors, hydraulic pistons, relays, and others. These MEMS components have high resonant frequencies leading to higher operating frequencies (Poslad 2009) and can be summarized as shown in Fig. 4.5 after (URL11 2017).

MEMS technology will enable a larger number of sensors to be used in connected cars and vehicles for autonomous driving, but this will require sophisticated networking to organize the real-time data transfer and fusion of the many sensors.

4.2.7.1 Sensor Nodes and Networks

Sensor nodes integrate sensors, actuators, computing elements, e.g., microcontrollers, memory, etc., communication systems; and a power source, e.g., a battery. A sensor node can also be a component of a larger network of sensors. Each sensor node in a sensor network is capable of performing processing, gathering sensor information, and communicating with other connected nodes in the network. They provide raw data to nodes responsible for sensor data fusion (SDF), which combines sensory data derived from distributed sources such that the resulting information has less uncertainty compared to the sources individually. Data sources

Fig. 4.5 Components of MEMS

MicroSensors	MicroActuators
MicroElectronics	MicroStructures

for sensor fusion are not necessarily specified to originate from identical sensors. Therefore, the sensor data fusion process can be:

- *Direct Fusion:* Based on a set of heterogeneous or homogeneous sensor outputs, on an algorithm where several measurements are processed together, or on historical values of sensor output data.
- *Indirect Fusion:* Uses information sources such as a priori knowledge about the environment and human input.

Raw data can also be processed by means of a sensor's computing capabilities, and the required output can be relayed to other sensor nodes.

Sensor networks usually interconnect many sensors or sensor nodes through wireless or wired connections (Golatowski et al. 2003). In data fusion in decentralized sensor networks with no central fusion system, no single sensor node has global knowledge of the network topology. Hence, algorithms are required which are able to locally process and assimilate data which finally yields a result identical to one obtained in a centralized system.

Wireless sensor networks (WSNs) consist of a large number of sensor nodes equipped with wireless network connections (WNCs) that can be deployed in the environment of the physical components of CPSs. In shared sensor and actuator networks (SANs), resource scheduling is an important feature for CPS operation.

The main components of a sensor node are, the microcontroller, the transceiver, the external memory, the power supply, and one or more sensors, resulting in a typical sensor node architecture as shown in Fig. 4.6.

The microcontroller component of the sensor node, shown in Fig. 4.6, performs specific tasks, processes data, and controls the functionality of other components in the sensor node. Microcontrollers are usually used because of their low cost, flexibility in connecting to other devices, easy programming, and low-power consumption.

Transceivers of sensor nodes represent a combination of a transmitter unit and a receiver unit into a single device. The operational states of transmitters are transmitting, receiving, idle, or sleep, which refers to their realization as state machines that

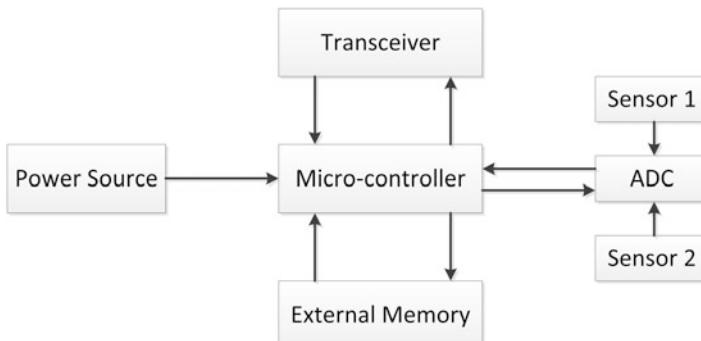


Fig. 4.6 Sensor node architecture

perform some operations automatically. A state machine is a mathematical model of computation used to design both computer programs and sequential logic circuits, acting on a set of inputs and computing a set of outputs. Thus, a finite state machine has a finite number of states to represent its state of processing. Its actions depend upon its internal state, and any inputs adhere to a specific syntax.

The memory requirements of sensor nodes depend on the specific application. There are two categories of memories usually used in sensor nodes:

- User memory to store application-related or personal data
- Program memory to program the device

The program memory can also contain the device's identification data.

An important issue in the development of wireless sensor nodes is ensuring that adequate energy is available to power the system because the sensor node requires power for sensing, communicating, and data processing. More of the required energy is used for data communication than for any other purpose. For example, the energy cost of transmitting 1 Kb a distance of 100 m (330 ft.) is approximately the same as that of executing 3 million instructions at 100 million instructions per second/W by a processor (URL12 2017). Power is stored either in batteries or capacitors. Batteries, both rechargeable and non-rechargeable, are the main source of power supplies in sensor nodes.

Wireless sensor nodes are typically very small electronic devices. They can only be equipped with a limited power source of less than 0.5–2.0 ampere hour and 1.2–3.7 volts (URL12 2017).

The energy efficiency for communication in a sensor network can be increased when using multi-hop topology (Zhao and Guibas 2004). In a N -hop network, overall transmission distance is Nd , where d is the average one-hop distance. The minimum receiving power is P_r , and the power at the transmission node is P_t . Thus, the power advantage P_A of a N -hop transmission versus a single-hop transmission over the same distance can be described as follows (Poslad 2009):

$$P_A = \frac{P_t(Nd)}{NP_t(d)} = \frac{(Nd)^a P_r}{Nd^a P_r} = \frac{N^a d^a P_r}{Nd^a P_r} = N^{a-1}$$

with a denoting the RF attenuation coefficient and $P_t \sim d^a P_r$.

Sensor nodes also play an important role in other domains, such as:

- Control systems
- Supervisory control and data acquisition (SCADA) systems
- Supervisory systems

For digital recording of analog quantities, analog-to-digital converters (ADC), shown in Fig. 4.7, are required (Möller 2003). The task of the ADC is to convert analog input variable X into a proportional output number. In many cases, time-dependent signals are digitized. For this purpose, the input quantity to be converted has to be sampled at a certain time and held. This task is performed by sample and

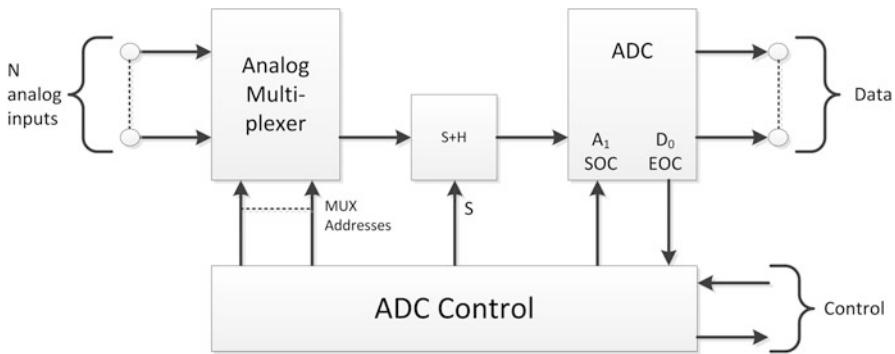


Fig. 4.7 Components of an analog-to-digital conversion system (Möller 2003)

hold (S/H) circuits, shown in Fig. 4.9. Very often, nonelectrical signals need to be digitally processed. Then, prior to the actual ADC conversion, the nonelectrical information needs to be converted into an electrical voltage. Sensors are used to detect nonelectrical information mapping an electrical voltage as output to the nonelectrical input.

In Fig. 4.7, the block structure of a conversion system converting analog inputs to digital data is shown. The control of the ADC is efficient for the following reasons:

- Several input channels are used; switching to a channel is controlled by using an analog multiplexer.
- After achieving the settling time, the S/H circuit switch is placed on hold; thus the conversion of a stable analog signal is possible. This does not affect the integration of the converter.
- Analog-to-digital conversion is started by the start-of-conversion (SOC) mode.
- After completion of the conversion, the analog-to-digital converter activates the end-of-conversion (EOC) mode.
- The ADC transfers the converted (digitized) measurements to the data processing unit (not shown in Fig. 4.7).

ADCs have a characteristic transmission curve in common, as shown in Fig. 4.8, with respect to the:

- Continuous abscissa pool y
- Discrete ordinate pool a

The intervals of variable Y can be mapped to a corresponding binary number a . In n -digit binary number $N = 2^n$, intervals are distinguished; the symmetrical ones are arranged around the abscissa values $0, Y, 2Y, \dots, iY, \dots, (N-1)Y$, as shown in Fig. 4.8. Hence, the values of the input voltage are on average in accordance with the converted binary number.

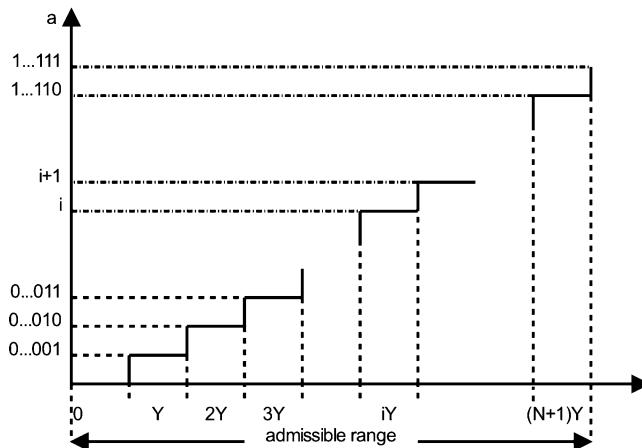
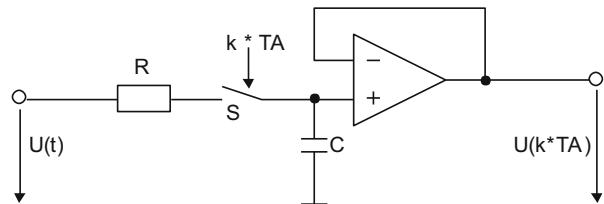


Fig. 4.8 ADC resolution characteristic (Möller 2003, 2016)

Fig. 4.9 S/H circuit of an ADC (Möller 2003, 2016)



Since the conversion process of analog-digital converters takes more than one clock cycle, the input signal U_{in} must remain constant throughout the conversion time period, which is achieved by an upstream S/H circuit. For this purpose, a hold signal is generated in the control logic of the converter, by which it is possible to define whether the input signal U_{in} should be held or should follow the real course of the analog input signal U_{in} . The S/H circuit consists of an operational amplifier connected as a voltage follower and a capacitor with low leakage current, as shown in Fig. 4.9.

4.3 E/E Architectures and Topologies

The number of electrical and electronic components in vehicles has increased rapidly and continuously during recent years. On the one hand, many new sensors and actuators and, therefore, new ECUs have been developed making passengers feel safer. On the other hand, entertainment and navigation systems have made their way into cars to make travel more comfortable.

The E/E architectures represent a consistent, vehicle-wide architecture of all E/E systems and E/E components on the hardware, software, electrical systems, cable

assembly, and topology levels. They ensure that all components which are safe to operate are reliably and efficiently supplied with electrical energy. An innovative solution for E/E systems communication networking and control includes the adequate design of the E/E system, optimal integration of onboard electrical components, and intelligent energy management, as well as testing and validation of the entire E/E system. Thus, E/E architecture development has to be done with the following constraints:

- Modularization in vehicle development with the goal of scaling effects through the reuse of E/E system components or modules
- Protection of the E/E architecture in the concept phase with the goal of developing the E/E architecture prior to setting systems/components

Bus systems are the connecting part for communication purposes.

4.3.1 Objectives

E/E systems have soared these systems are increasingly important for efficient, comfortable, and safe vehicle systems functions. But the more technology that goes into a vehicle, the more important it is to optimize the E/E architecture. Additional functions, such as brake assistance, skidding control, parking aid, and others, have been embedded, which in turn leads to an increase in the number of ECUs and their communication links. Thus, developing E/E architectures is rigorous up-front work that involves the overall E/E system and subsystem design, the physical and functional partitioning, and the physical layout of subsystems within the vehicle. Therefore, E/E system development is the most important building block in designing the ever more complex technologies in automotive systems and products in a powerful and cost-effective manner.

With the respective system knowledge of E/E, networking, systems functions, and management, adequate conditions exist for successfully developing and implementing E/E systems and products. Starting with functions and the corresponding function chains, the design of components, data buses, and line connections exists and takes into account important boundary conditions, such as:

- Bus load
- Installation space
- Topology

As a result, the design of E/E architectures is a core requirement in the automotive domain. With regard to increasing complexity, many communication links and dependencies between composite functions prevent a simple decomposition of the E/E architecture design task into independent subtasks, which in turn consist of a number of individual components that communicate by signals with each other. Such components can either be sensors, processing components, actuators, or others.

Hence, E/E is a rapidly evolving field, driven by different requirement profiles, such as consumer demand, governmental regulation, increased E/E content, and others. Therefore, suitable networking systems are essential to exchange data quickly and securely. Examples are the bus systems, CAN, LIN, or FlexRay™, in the automotive field or the various Ethernet derivatives in the automotive industry, as well as in industrial and automation technology. Using these communication systems, the best possible interpretation is required. The transmission of all relevant data must also be ensured in critical situations in order to ensure the secure and perfect functionality of the overall system.

4.3.2 Architectures and Topologies

Modern automotive E/E architectures are growing in complexity to the point where it is difficult for the designer to predict the effects of the design decisions precisely. Thus, in addition to applying an architecture reference model to decompose the architecture, the design also requires domain-specific tools for synthesizing and evaluating the architecture during the design process. However, the complexity of E/E systems makes their optimization complicated, too, because multiple design goals refer to dependability and other non-functional requirements are crucial to take into account, such as:

- Adaptability
- Costs
- Maintainability
- Performance
- Power consumption
- Reliability
- Response time
- Safety

Based on a set of mechatronic subsystems with their sensors and actuators, a respective topology has to be defined in terms of data/signal integrity; however, several challenging issues have to be considered beyond the number of subsystems (ECUs). The required topology can be understood as a structure consisting of nodes and connections showing which nodes are interconnected. Different topologies are in use in vehicles that incorporate the E/E architecture-specific function into the vehicle, whereby its components can be assigned to ECUs. The ECUs, in turn, must be placed in an appropriate location in the vehicle and assigned with the respective hardware, depending on the components' requirements. Thus, ECUs have to be assigned to a topological structure such as:

- *Star Topology*: There is a central hub at which all ECUs are connected. Each ECU has its own line. If the central hub fails, the entire communication breaks down.

- **Bus Topology:** ECUs are connected by short branch lines to a main line. Every communication flows over this main line. If this main line is interrupted, two segments are formed which normally continue functioning. This topology is also called linear bus topology. Advantages of the linear bus topology are:
 - Bus topology is inexpensive.
 - The cable length required for this topology is the shortest compared to other networks.
 - It is easy to set up and to extend the bus network.
 - The linear bus network is used mostly in small networks.
 Disadvantages of the linear bus topology are:
 - A dependency on central cable has its disadvantages; if the main cable encounters a problem, the whole network breaks down.
 - It is difficult to detect and troubleshoot a fault at an individual station.
 - The efficiency of the bus network reduces as the number of devices connected to it increases.
 - The central cable length and the number of nodes that can be connected are limited.
 - Maintenance costs can get higher over time.
 - It is not suitable for networks with heavy traffic.
 - Termination is required to dump signals, and use of terminators is required.
 - Security, generally, is low because all of the computers receive signals sent from the source.
- **Ring Topology:** Point-to-point connection between ECUs is representative of ring topology. All connections are arranged in a closed chain. The communication can be done in only one direction. If a section of the line fails, the entire system no longer functions.

Figure 4.10 is an example of an abstract E/E architecture design introduced by (Moritz et al. 2011), in which gray circles with a black center point indicate sensors, gray circles with a white center point indicate actuators, the gray circle itself represents ECUs, and dashed lines refer to digital bus systems.

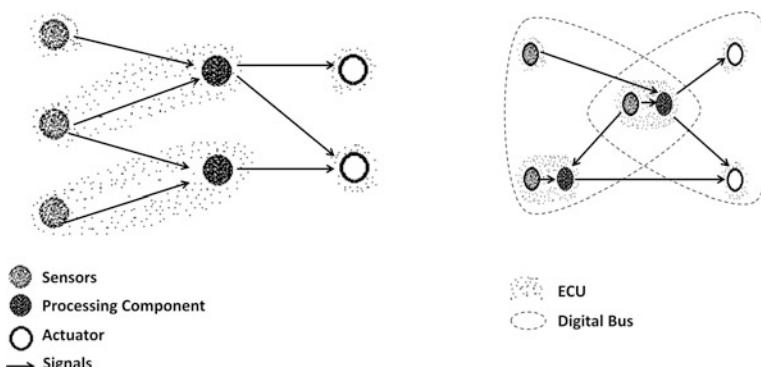


Fig. 4.10 E/E architecture design examples (Moritz et al. 2011) (for details, see text)

From Fig. 4.10, it can be seen that the E/E architecture of a given function contains several processing components, sensors, and actuators with corresponding ECUs assigned to digital buses, where the middle ECU acts as a gateway between the buses chosen. The arrows indicate the signal flow directions. The intrabus topology is not shown in Fig. 4.10.

The resulting architectures are evaluated according to two objectives, which are to be minimized (Moritz et al. 2011). The first one is cost, which is governed by cable and ECU cost. Cables in turn depend on the signals that have to be routed between the ECUs and on the communication structure used, as well as on the placement of the ECUs. The second objective is ECU complexity, defined as the average number of different functions assigned to an ECU. As ECU complexity decreases, the reliability of a vehicle, as to the number of functions which are affected by a single ECU failure, increases.

Optimization of the E/E architecture design can be based on the method of evolutionary algorithms which are primarily used for problem classes where classical standard methods fail or can hardly be applied (Yu and Gen 2010). In order to make the optimization procedure efficient, the evolutionary algorithm (EA) has to be adapted to the solution under test. Therefore, solution-specific knowledge is required for the evolutionary algorithm through (1) embedded local heuristics, (2) appropriate representation of solutions, and (3) corresponding variation operators. The use of local heuristics reduces the search space size of the evolutionary algorithm, but the available diversity of solutions in the population might be lost (Grosan and Abraham 2007). As described in (Moritz et al. 2011), the EA-optimized decisions are:

- Task 1: Assignment of components to ECUs
- Task 2: Physical placement of ECUs
- Task 3: Assignment of ECUs to digital buses

The decisions made by local heuristics are:

- Intelligent semiconductor selection for each ECU (simply take the cheapest one that satisfies the memory requirements of the processing components of that ECU).
- Bus type selection (choose the cheapest type that satisfies the data-rate requirements of the signals that have to be routed over the bus).
- Intrabus topology (choose such that the cable cost is minimized).

With regards to the aforementioned, the number of tasks that have to be optimized by the evolutionary algorithm has been reduced to the design of a suitable representation to optimize these tasks simultaneously. The assignment of components to ECUs corresponds to a partitioning of the components into clusters, whereby the number of clusters is the parameter to be minimized. The resulting E/E architectural representation of the hierarchical partitioning is shown in Fig. 4.11 (Moritz et al. 2011).

In Fig. 4.11, small gray circles indicate sensors or components; and small white circles indicate actuators. The arrows represent the data flow from sensors,

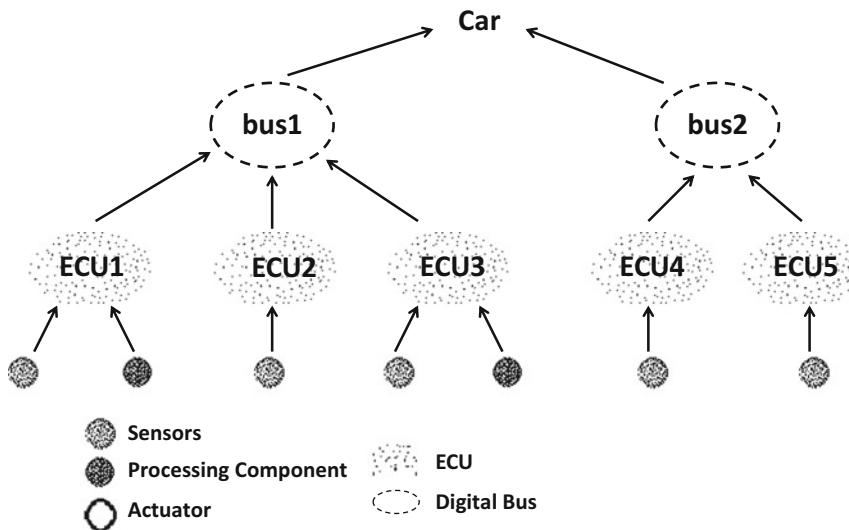


Fig. 4.11 E/E architecture design example representation after hierarchical partitioning (Moritz et al. 2011)

components, and actuators to ECUs, from ECUs to the bus systems, bus1 and bus2, and from the bus systems to the vehicle (car).

4.3.3 Bus Systems and ISO Standards

Modern vehicles have a large number of different kinds of ECUs through which specific vehicle functions are provided, as introduced in Sect. 4.2.5. One group of these ECUs can be stand-alone or distributed among the vehicles' EE multifunctional components. The majority are connected to one or more bus systems to control/monitor a broad range of vehicles. Another group of ECUs are those that have external interfaces, such as infotainment electronics, navigation systems, and others.

ECUs are interconnected through specialized internal communication networks inside vehicle bus systems. To meet the design challenges due to the different requirements, such as capacitance, real-time operation, and cost, several bus systems have been developed. The most important bus systems currently used in vehicles are:

- *Controller Area Network*: Inexpensive low-speed serial bus protocol for interconnecting automotive electronic components, such as microcontrollers and other devices, to communicate with each other in applications without a host computer, invented in 1986 by Robert Bosch GmbH and focused on safety, i.e., reliability. Bus nodes are all connected to the same shared bus line. A CAN bus is wired such that a 0-signal is dominant over a recessive 1-signal, as shown

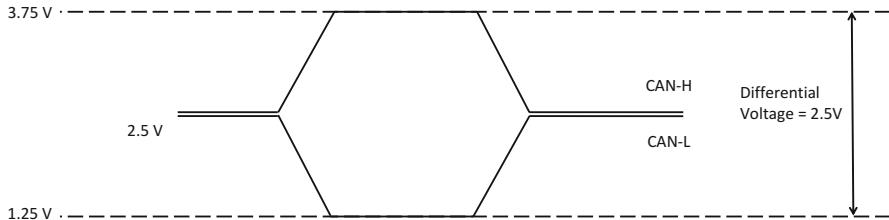


Fig. 4.12 CAN bus output signal

in Fig. 4.12. These dominant and recessive signals are used for a carrier-sense multiple access with collision avoidance (CSMA/CA) protocol that operates in the data link layer (layer 2) of the open systems interconnection (OSI) model that characterizes and standardizes the communication functions of a computing system without regard to their underlying internal structure and technology. Its goal is the interoperability of diverse communication systems with standard protocols. The model partitions a communication system into seven layers. An arbitration scheme using priority resolution is used to decide which node is allowed to transmit data over the bus. The lower signal is its ID (and thus the more dominant 0s it sends during bus arbitration, the higher its priority). Using this scheme allows the CAN bus to communicate in real time.

Originally there was no built-in possibility to enforce security, such as encryption or authentication. Due to this original architecture, vehicles were vulnerable to cyberattacks, being hacked remotely and immobilized as a result of the hacking. The intrusion point for cyberattacks can be, for example, through the wireless tire pressure sensor (TPS) or directly, i.e., by physically accessing the CAN bus through the vehicle's entertainment/infotainment system. A solution to this problem is the CAN bus firewall (CBF) which separates each of the externally accessible ECUs from the shared bus by filtering the messages being sent from the ECUs and making sure that no hostile messages are going through and attacking the vehicle. It can also ensure that no unauthorized ECUs are connected to the vehicle. More details are discussed in Sects. 6.2.1 and 6.3.2.

- *Local Interconnect Network:* Broadcasting serial network protocol used for communication between components in vehicles comprised of 16 nodes. In the late 1990s, the LIN Consortium was founded by five automakers: Audi Group, BMW, Mercedes-Benz, Volvo, and VW Group, with networking technologies from Volcano Automotive Group and Motorola. The first fully implemented LIN specification was published in November 2002. The current LIN application combines the low-cost efficiency of LIN and simple sensors to create small networks. These subsystems can be connected by a backbone network, such as CAN, in vehicles.
- *FlexRay:* Automotive network communication protocol to govern onboard computing is a deterministic, error-tolerant, high-speed bus system developed by a consortium of leading automakers and Tier 1 suppliers with networking

technologies from National Instruments. It supports the essential needs required for drive-by-wire, steer-by-wire, and brake-by-wire applications. The consortium disbanded in 2009; but the FlexRay standard is now a set of ISO standards, ISO 17458-1–17458-5.

- *Ethernet for Control Automation Technology (EtherCAT)*: Initiated by Beckhoff Automation as a real-time Ethernet, disclosed as International Electrotechnical Commission (IEC) standard 61158 protocols. EtherCAT is suitable for hard and soft real-time requirements in automation technology. The EthernetCAT Technology Group has 3905 members worldwide.
- *MOST Bus*: High-speed multimedia network technology optimized by the automotive industry and used for applications inside or outside the vehicle.

The standard multi master serial CAN bus (ISO 11898-1:2003) was originally specified as link layer protocol for the physical layer, for example, asserting the use of a medium with multiple access at the bit level through use of dominant and recessive states. The complexity of the ECU can range from a simple I/O device up to an embedded computer with a CAN interface and sophisticated software. The ECU may also be a gateway allowing a standard computer to communicate over a USB or Ethernet port to the devices on a CAN network. All ECU nodes are connected to each other through a twisted two-wire bus, CAN high (CAN-H) and CAN low (CAN-L). If a signal on the CAN-H wire goes from 2.5 to 3.75 V, the corresponding signal on the CAN-L wire goes from 2.5 to 1.25 V as shown in Fig. 4.12.

The CAN network, ISO 11898-2, is called a high-speed CAN using a linear bus terminated at each end with $120\ \Omega$ resistors. The electrical aspects of the physical layer, such as voltage, current, number of conductors, were specified by the International Organization for Standardization (ISO) as ISO 11898-2:2003. However, mechanical aspects of the physical layer, such as connector type and number, colors, labels, and pin-outs, have to be formally specified, too. As a result, an automotive ECU will typically have a particular connector with various sorts of cables, of which two are the CAN bus lines. The most common mechanical connector for the CAN bus is the 9-pin D-sub type male connector with the following pin-out:

- Pin 2: CAN-L (CAN–)
- Pin 3: GND (Ground)
- Pin 7: CAN-H (CAN+)
- Pin 9: CAN V+ (Power)

CAN is a serial bus protocol to connect individual sensors, ECUs, and systems as an alternative to conventional multiwire looms allowing automotive components to communicate on a single- or dual-wire networked data bus up to 1 megabit per second (Mbps). They are designed to allow microcontrollers and other devices (actuators, sensors, etc.) to communicate with each other in applications without a

host computer. A CAN system allows the use of a single command station to control diagnostic systems and receive varied information, such as:

- Brake and transmission temperature
- Emissions levels
- Fuel efficiency
- Tire pressure

The CAN bus system is characterized by the following:

- All messages are broadcasted.
- Any node is allowed to broadcast a message.
- Each message contains an identification (ID) that identifies the source or content of a message.
- Each receiver decides to process or ignore each message.

There are four areas of application for serial communication (CAN) in automotive deployment which are subject to different requirements and objectives.

- *Connecting ECUs for Controlling Engine, Transmission, Suspension, and Brakes:* Data transfer rates are in the typical range for real-time applications ranging from 200 kilobit per second (kbps) to 1 megabit per second (Mbps).
- *Mobile Communication:* Connects components, such as vehicle radios, vehicle phones, navigation devices, etc., with a central ergonomically designed control unit.
- *Networking Components of Body and Convenience Electronics:* Multiplex applications, such as air control, air conditioning, central locking, and seat and mirror adjustment. Particular attention is paid to the cost of components and wiring. Typical data rates are in the vicinity of 50 kbps.
- *Onboard diagnostics (OBD):* Hard-wired communication link to the ECU through which is allowed access to read and reset a vehicle's fault code. Fault codes are also known as diagnostic trouble codes typically made up of a letter followed by four numbers. So each code has a total of five characters, for example, B32XX. The first character, B, shows the identification for body systems, such as airbags, climate control, lighting, etc. The second character, 3, refers to a manufacturer-specific code, while the third character, 2, refers to the secondary air injection system. The fourth and fifth characters XX refer to the actual component that the ECU has identified with a fault. Once the OBD connector has access to the CAN bus, it is possible to monitor every component connected to it.

The CAN system bus data message structure, shown in Fig. 4.13, is the same for both the standard and the extended version.

SF	Message Identifier	Control	Data	CRC	ACK	EF
1bit	11 or 29 bits	6 bits	upto 64 bits	16 bits	2 bits	7 bits

Fig. 4.13 CAN bus data message structure

In Fig. 4.13, the acronyms have the following meaning:

- *Start Field (SF)*: Marks the start of the data protocol. A bit with 3.75 V (depending on the system used) is sent over the CAN-H line, and a bit with 1.25 V is sent over the CAN-L line, i.e., the differential voltage is 2.5 V, as shown in Fig. 4.12.
- *Message Identifier*: Defines the priority level of the data protocol. If two CAN nodes try to transmit a message onto the CAN bus at the same time, the node with the highest priority (lowest arbitration ID) gets bus access. Depending on the standard being used, the length of the frames can be in two formats: standard, which uses an 11-bit arbitration ID, and extended, which uses a 29-bit arbitration ID, as indicated in Fig. 4.13
- *Control or Check Field*: Displays the number of items of information contained in the data field. This field allows any receiver to check if it has received all of the information transferred to it.
- *Data Field*: In this field, information is transferred to other CAN nodes.
- *Cyclic Redundancy Check or Safety Field (CRC)*: Contains a 15-bit cyclic redundancy check code and a recessive delimiter bit. The CRC field is used to transfer fault detection.
- *Acknowledge Field or Confirmation Field (ACK)*: Receivers send signal to transmitter that the data protocol has been correctly received. If an error is detected, the receivers notify the transmitter immediately. The transmitter then sends the data protocol again.
- *End Field (F)*: Marks the end of the data protocol. The last possibility to indicate errors which lead to a repeat transfer.

CAN, LIN, and FlexRay are mainly used for control systems, whereas Media Oriented Systems Transport (MOST) is used for telematic applications. A bus system overview with regard to specific features is shown in Table 4.3.

Furthermore, intelligent and highly integrated actuator and sensor nodes in vehicles communicate via their adequate system network. Most body electronics applications use CAN or LIN communication interfaces. Application requirements, such as latency and bandwidth, as well as cost, influence the selection of a specific interface. Since the actual communication physical layer is driven mostly by electric and electromagnetic requirements, such as electromagnetic compatibility (EMC), electromagnetic interference (EMI), and electromagnetic discharge (EMD) standards, it is not a negligible portion of the device area. Normally, either a CAN physical layer or a LIN is integrated according to the respective application needs. In addition to CAN and LIN protocols, other communication interfaces like the ones for

Table 4.3 Bus system overview

	CAN	LIN	FlexRay	MOST
Application	Soft real-time systems	Low-level communication systems	Hard real-time systems	Multimedia, telematics
Bandwidth	500 kBit/s	19.6 kBit/s	10 Mbit/s	24.8 Mbit/s
Bus access	CSMA/CA	Polling	TDMA/FTDMA	TDM/CSMA
Control	Multi master	Single master	Multi master	Timing master
Data bytes per frame	0–8	0–8	0–254	0–60
Physical Layer	Electrical (twisted pair)	Electrical (single wire)	Optical, electrical	Mainly optical
Redundant channel	Not supported	Not supported	Two channels	Not supported

the power train or chassis domain, such as Single-Edge Nibble Transmission (SENT) or Peripheral Sensor Interface 5 (PSI5), are gaining interest for use in further reducing network costs. For example, the use of PSI5 instead of LIN reduces the number of wires and connector pins from three (LIN, VBAT (battery voltage)), GND (ground)) down to only two (supply, GND). Even though PSI5 needs to modulate the data onto the supply line, the savings on the harness and connector side may be sufficient to account for the higher requirements on the E/E side.

In spite of the continued development of lower-cost protocols, the overall trend to CAN- and LIN-based nodes has been observed for many years in automotive sensor and actuator networks, especially in body electronics applications. According to strategic analytics, it is expected that in 2018, the number of CAN nodes will exceed the mark of $2 \cdot 10^9$; and the number of LIN nodes will exceed $1 \cdot 10^9$. In this regard, the average number of nodes per vehicle will be around 20 CAN nodes and approximately 10 LIN nodes. With a 17% compound annual growth rate, the expected market growth for LIN nodes is significantly higher compared to CAN nodes, with a 13% compound annual growth rate. This shows that simple functions are increasingly implemented in LIN nodes.

Ethernet, a family of computer technologies commonly used in LANs and metropolitan area networks, has been refined to support higher bit rates and longer link distances and will be the protocol of choice forming the backbone of the domain network. Also, a new super high-speed CAN bus called, CAN with Flexible Data-Rate (CAN FD), was developed by Bosch. CAN FD is positioned between the classic high-speed CAN and FlexRay and offers very high bandwidths. Many next generation ECUs will have a CAN FD interface. FlexRay and LIN will provide connectivity to intelligent nodes within a vehicle subdomain. But embedding powerful domain controllers will require an adequate support of this highly interconnected architecture. All cars sold in the USA since 1996 are required to have an onboard diagnostics (OBD) connector for access to vehicles' ECUs. Onboard diagnostics refers to a vehicle's self-diagnostic and reporting capability.

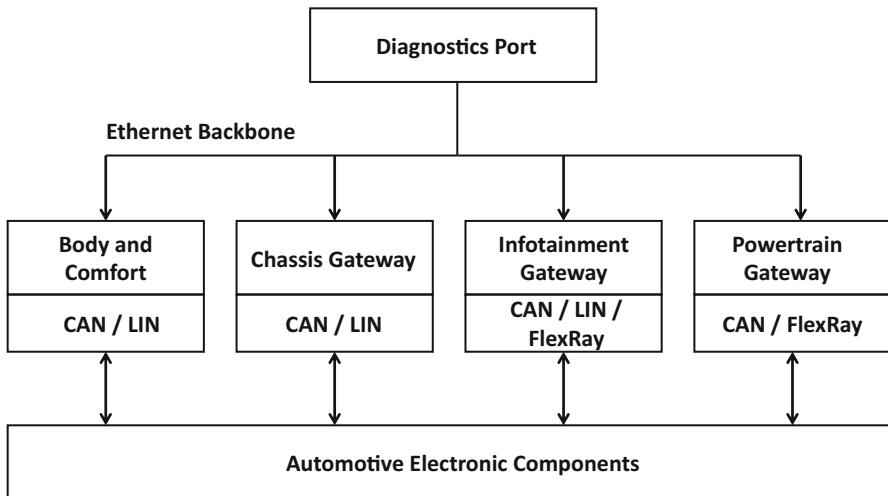


Fig. 4.14 Vehicle networks with regard to their application domains (URL1 2013)

In Fig. 4.14, an example of a vehicle network partitioned into separate application domains with associated domain controllers is shown. These domain controllers require significant amounts of processing power coupled with real-time performance and a plethora of communications peripherals.

The IEC 61508 standard, generally applicable to E/E programmable safety-related products, is only partially adequate for automotive electronics development requirements. Consequently, for the automotive industry, this standard is replaced by the existing ISO 26262, currently released as a Final Draft International Standard (FDIS). ISO/DIS 26262 describes the entire product life cycle (PLC) of safety-related EE components for vehicles. In this regard, PLC is the process of managing the entire life cycle of an automotive electronic component from inception, through engineering design and manufacturing, to service and disposal of manufactured components. It was published as an international standard in its final version in November 2011.

The implementation of this standard will result in modifications and various innovations in the automobile electronics development process, as it covers the complete product life cycle from the concept phase to decommissioning. It also has a safe assure solution, developed in accordance with the automotive functional safety standard (ISO 26262) and is targeted at specific safety functions of at least an Automotive Safety Integrity Level (ASIL) B rating. The safety integrity level (SIL) is defined as a relative level of risk reduction provided by a safety function or to specify a target level of risk reduction. The European functional safety standard is based on the IEC 61508 standard which defines four SILs where SIL 4 is the most dependable and SIL 1 is the least dependable, as shown in Table 4.4 (URL14 2017).

The SIL requirements for hardware safety integrity, shown in Table 4.4, are based on a probabilistic analysis of devices represented as a function of probability of failure on demand (PFD) and risk reduction factor (RRF) of low demand device

Table 4.4 SIL as defined in IEC 61508 (for details see text)

SIL	PFD	PFD (power)	RRF
1	0.1–0.01	10^{-1} – 10^{-2}	10–100
2	0.01–0.001	10^{-2} – 10^{-3}	100–1000
3	0.001–0.0001	10^{-3} – 10^{-4}	1000–10.000
4	0.0001–0.00001	10^{-4} – 10^{-5}	10.000–100.000

operation. The PFD is a measure of the effectiveness of a safety function, expressing the likelihood that the system will not perform the required safety function. For example, the likelihood that a SIL-3 system does not shut down a process when required is better than 1 in 1000 or 0.1%, as shown in Table 4.4. In other words, the availability of the safety function is better than 99.9%. Alternatively, it may help to think of a reduction of risk by a factor of 1000 ([URL15 2017](#)). The PFD of a one-channel system can be calculated by using a Markov model.

To achieve a given SIL, the hardware device must meet targets for the maximum probability of dangerous failure and a minimum safe failure fraction. Hence, a SIL is determined based on a number of quantitative factors in combination with qualitative factors, such as the development process and safety life cycle management.

4.4 Functional Safety

Functional safety is part of the overall safety of a vehicle system, or a component of it, that depends on the CPS and its components operating correctly in response to inputs, including the safe management of likely operator errors, hardware failures, and environmental changes. Functional safety is intrinsically end-to-end in scope as it has to treat the function of a system, subsystem, or component as part of the function of the whole system. This means that while functional safety standards focus on electrical, electronic, and programmable electronic (E/E/PE) systems, the end-to-end scope means that in practice functional safety methods have to extend to the non-E/E/PE parts of the system that the E/E/PE actuates, controls, or monitors. Therefore, the aim of functional safety is to bring risk down to a tolerable level and to reduce its negative impact; however, there is no such thing as zero risk. Functional safety measures risk by how likely it is that a given event will occur and how severe it would be or in other words, the amount of harm it could cause.

Thus, functional safety is achieved when every specified safety function is carried out and the level of performance required of each safety function is met. This is normally achieved by a process that includes the following steps as a minimum ([URL1 2013](#)); ([URL13 2017](#)).

- *Identify the Required Safety Functions:* This means hazards and safety functions have to be known or can be identified.
- *Assess the Risk Reduction Required by Safety Functions:* This involves a SIL, or performance level (PL), or other quantification assessment. A SIL applies to an end-to-end safety function of the safety-related system, not just to a component or part of the system.

- *Automotive Safety Integrity Level:* A risk classification scheme defined by the ISO 26262 standard, *Road Vehicles—Functional Safety*, which is an adaptation of the SIL used in IEC 61508 for the automotive industry. This classification helps define the safety requirements necessary to comply with the ISO 26262 standard. An ASIL is established by performing a risk analysis of a potential hazard by looking at the severity, exposure, and controllability of the vehicle operating scenario. The safety goal for that hazard, in turn, carries the ASIL requirements. There are four ASILs identified by the standard: ASIL A is comparable to SIL-1, ASIL B/C is comparable to SIL-2, and ASIL D is comparable to SIL-3. There is no ASIL comparable to SIL-4. ASIL D dictates the highest integrity requirements for a product and ASIL A the lowest. However, ISO 26262 does not provide normative nor informative mapping of ASIL to SIL. ASIL is a qualitative measurement of risk; SIL is quantitatively defined as a probability or frequency of dangerous failures depending on the type of safety function. Thus, in IEC 61508, higher risk applications require greater robustness to dangerous failures.

In general, IEC 61508-1:2010 covers aspects to be considered when E/E/PE systems are used to carry out safety functions. A major objective of this standard is to facilitate the development of product and application sector international standards by the technical committees responsible for the product or application sector. This will allow all of the relevant factors associated with the product or application to be fully taken into account and thereby meet the specific needs of users of the product and the application sector. A second objective of this standard is to enable the development of E/E/PE safety-related systems where product or application sector international standards do not exist.

Furthermore, IEC 61508-3:2010 applies to any software forming part of a safety-related system or used to develop a safety-related system within the scope of IEC 61508-1 and IEC 61508-2. It provides:

- Specific requirements applicable to support tools used to develop and configure a safety-related system within the scope of IEC 61508-1 and IEC 61508-2
- Requires that the software safety functions and software systematic capability are specified
- Establishes requirements for safety life cycle phases and activities which will be applied during the design and development of the safety-related software These requirements include:
 - Applying measures and techniques which are graded against the required systematic capability for the avoidance of and control of faults and failures in the software.
 - Providing requirements for information relating to the software aspects of system safety validation to be passed to the organization carrying out the E/E/PE system integration
 - Providing requirements for the preparation of information and procedures concerning software needed by the user for the operation and maintenance of the E/E/PE safety-related system
 - Providing requirements to be met by the organization carrying out modifications to safety-related software

- Providing, in conjunction with IEC 61508-1 and IEC 61508-2, requirements for support tools, such as development and design tools, language translators, testing and debugging tools, and configuration management tools

The second edition cancels and replaces the first edition published in 1998. This edition constitutes a technical revision. It has been subject to a thorough review and incorporates many comments received at the various revision stages ([URL16 2017](#)).

- *Ensure Safety Function Performs to the Design Intent:* This includes under conditions of incorrect operator input and failure modes. It also involves having the design and life cycle managed by qualified and competent engineers carrying out processes to comply with a recognized functional safety standard. In Europe, that standard is IEC EN 61508, one of the industry-specific standards derived from IEC EN 61508, or some other standard, such as ISO 13849.
- *Verify That the System Meets the Assigned SIL (ASIL, PL, or Agricultural Performance Level (agPL)):* This can be accomplished by determining the mean time between failures (MTBF) and the safe failure fraction (SFF), along with appropriate tests. SFF is the probability of the system failing in a safe state. The critical or dangerous state is identified from a failure mode and effects analysis (FMEA) or failure, mode and effects and critical analysis (FMECA) of the system under test.
 - *Mean Time Between Failure (MTBF):* The predicted elapsed time between inherent failures of a system during operation which can be calculated as the **arithmetic mean** time between **failures** of a system using the following equation:

$$\text{MTBF} = \frac{\sum (\text{start of downtime} - \text{start of uptime})}{\text{number of failures}}$$

- *Mean Time to Dangerous Failure (MTTF_d):* The MTTF_d-value should primarily be provided by the system manufacturer. If the manufacturer cannot provide the required values, they can be taken from ISO 13849-1 tables or can be calculated using the B_{10d}-value, (average number of cycles until 10% of the components have a dangerous failure). To calculate the MTTF_d, it is also important to know the average number of cycles per year the component will execute.

A B_{10d} = 2·10⁶ results in MTTF_d = 1,141 year which corresponds to the level MTTF_d = high.

$$\text{MTTF}_d = \frac{B_{10d}}{0.1 \cdot n_{op}}$$

where

$$n_{op} = \frac{d_{op} \cdot h_{op} \cdot 3600}{t_{cycle}}$$

with

n_{op} = Number of cycles per year

d_{op} = Operation days per year

h_{op} = Operation hours per day

t_{cycle} = Cycle time in seconds

- *Safe Failure Fraction:* Takes into account any inherent tendency to fail toward a safe state. An SFF is the sum of the rate of safe failures plus the rate of detected dangerous failures divided by the overall failure rate:

$$SFF = \frac{\sum \lambda_S + \sum \lambda_{DD}}{\sum \lambda_S + \sum \lambda_D}$$

It is important to realize that the only types of failures to be considered are those which could have some effect on the safety function.

where

λ_S : Rate of safe failure

$\left(\sum \lambda_S + \sum \lambda_D \right)$: Overall failure rate

λ_{DD} : Rate of detected dangerous failure

λ_D : Rate of dangerous failure

- *Failure Mode and Effects Analysis (FMEA):* The first step of a system reliability study involving reviewing as many components, assemblies, and subsystems as possible to identify failure modes and their causes and effects. For each component, the failure modes and their resulting effects on the rest of the system are recorded in a specific FMEA worksheet. FMEA can be a qualitative analysis but may be put on a quantitative basis when mathematical failure rate models are combined with a statistical failure mode ratio database ([URL17 2017](#)).

By early dealing with possible sources of error, a strategy of error avoidance is followed instead of elaborate error correction. The FMEA is, therefore, particularly suitable for new developments and changes to products and processes. The risk assessment enables critical components to be found and priorities to be set in the prevention of errors.

- *Failure Mode and Effects and Criticality Analysis (FMECA):* Extended FMEA indicates that criticality analysis is also performed. For each failure mode, the ability of the system to detect and report the failure in question is analyzed. One of the following will be entered on each row of the FMECA matrix:

- *Normal:* The system correctly indicates a safe condition to the designer.

- *Abnormal*: The system correctly indicates a malfunction requiring design action.
- *Incorrect*: The system erroneously indicates a safe condition in the event of malfunction or alerts the designer to a malfunction that does not exist (false alarm).

Failure mode criticality assessment may be qualitative or quantitative. For qualitative assessment, a mishap probability code or number is assigned and entered on the matrix. For example, MIL-STD-882 uses five probability levels:

Level	Description
A	Frequent
B	Probably
C	Occasional
D	Remote
E	Improbably

The criticality numbers are computed as:

$$C_m = \lambda_p \frac{\alpha}{\beta' t}$$

and

$$C_\tau = \sum_{n=1}^N (C_m)_n$$

with

λ_p = Basic failure rate

α = Failure mode ratio

β' = Conditional probability

t = Mission phase duration

- *Conduct Functional Safety Audits*: Examine and assess the evidence that the appropriate safety life cycle management techniques were applied consistently and thoroughly in the relevant life cycle stages.

Neither safety nor functional safety can be determined without considering the vehicle cyber-physical system as a whole and the environment with which it interacts. Functional safety is inherently end-to-end in scope.

4.5 Automotive Software Engineering

The automotive industry faces global competition where speed, cost-efficiency, and innovative power are decisive factors for securing the future of automakers. Current challenges for automakers, such as innovative drive concepts, Car IT, and driver assistance systems increase the demand for automotive software engineering with regard to information technology (IT) and E/E. This trend is being reinforced by the networking of vehicles for assistance and piloting services.

The key challenge today is that almost all functions of a vehicle are electronically controlled or monitored. The realization of vehicle functions controlled by software offers unique degrees of freedom in the design process. But the main difference between automotive software and other types of software, such as personal computers and telecommunication systems, is in the essential requirements for reliability. In a complex ECU network, automotive software must be exceptionally reliable over the full vehicle life cycle. Therefore, in vehicle development, main boundary conditions, such as high reliability and functional safety requirements, comparatively long product life cycles, complexity of software functions, limited costs, shorter development times due to shorter innovation cycles, and an increasing variety of vehicle variants, have to be considered. The number of software functions of a motor control unit has now reached the three-digit range.

However, powerful software functions must be implemented in vehicles which interact internally but also have numerous interfaces to functions in the chassis or body area, for example, to the drive slip control or automatic climate control, and other features. Typical is the high number of parameters, such as characteristic values, characteristic curves, and key fields, which are used to coordinate the software functions for the respective systems' or components' undisturbed functionality, such as engine, gearbox, vehicle variant, and other important functions (Schäuffele and Zurawka 2016).

Another essential requirement for automotive software engineering is that the ECUs are highly interconnected by in-vehicle networks. ECUs communicate via standardized bus systems, such as CAN, LIN, FlexRay, and MOST. In contrast to Ethernet, well-known from PC connectivity, CAN and LIN bus systems are rather slow. In vehicles information must be processed within milliseconds. However, the increasing number of connected ECUs has led to more elaborate structures. Thus, in-vehicle networks require specific E/E architecture designs. To master the complexity of these networks, vehicle ECUs are partitioned into domains, such as power train, chassis, body/interior, infotainment, and others, whereby each domain has different requirements. For example, the power train domain requires extremely precise timing, closed-loop control, and real-time behavior, whereas infotainment needs optimal presentation of information. For example, the power train domain requires extremely precise timing, closed-loop control, and real-time behavior, whereas infotainment needs optimal presentation of information.

4.5.1 Increasing Software Content and Product Complexity

Today's E/E systems carry out many functions in modern vehicles, including driver assistance functions, vehicle dynamics control, active/passive safety systems, and other features, the functionality of which is embedded through software content. Automotive software has several main requirements with regard to:

- *Functional Safety*: Functions such as antilock braking, ESC, and others require fail-safe operation, which puts high demands on software development processes and the software functionality itself.
- *Minimized Resource Consumption*: Additional computational power and memory capacity must be minimized as the need for them continues to grow.
- *Real-Time Behavior*: Defined fast reaction on external incidents requires optimized operating systems and specific software content design.
- *Reliability*: Automotive software must be exceptionally reliable within the complex and multifaceted ECU networks over the full vehicle life cycle.

These requirements result in increased complexity of ECU software functions and therefore increased product complexity with regard to the number of lines of code.

The methods, standards, and processes used for automotive software development and tests are:

- *Agile Software Development*: Principles for software development based on which requirements and solutions evolve through the collaborative effort of self-organizing, cross-functional developer teams.
- *Automotive SPICE®*: Industry-specific standard derived from ISO 15504 for software process assessments, published by the Special Interest Group Automotive in 2005. It has two dimensions: its own process reference model (PRM) and process assessment model (PAM).
- *AUTOSAR*: A worldwide development partnership founded in 2003 (see Sect. 4.6).
- *Diagnostic Standard of Association for Standardization of Automation and Manufacturing (ASAM) Systems*: Provides standards for data models, interfaces, and syntax specifications for a great number of applications, examples, evaluations, and simulations.
- *HIL*: Technique used in developing and testing complex real-time systems by using simulation (see Sect. 4.5.3).
- *Model-Based Development*: Model-based method addressing problems associated with developing complex control systems (see Sect. 4.5.2).
- *Safety Standards*: According to ISO 26262 and IEC 61508.
- *State-of-the-Art Requirements Engineering*.

The standards used for network connection are:

- CAN (see Sect. 4.3.3)
- Ethernet (see Sect. 4.3.3)
- FlexRay (see Sect. 4.3.3)
- LIN (see Sect. 4.3.3)
- MOST (see Sect. 4.3.3)

The increasing complexity of software applications requires efficient development of high-quality software code. Thus, software engineering, can be regarded as a study and application of engineering to the design, development, and maintenance of software, becomes very important. In this regard, agile software development describes a set of principles for software development under which requirements and solutions evolve through the collaborative effort of self-organizing cross-functional teams. It advocates adaptive planning, evolutionary development, early delivery, and continuous improvement; and it encourages rapid and flexible response to change. These principles support the definition and continuing evolution of many software development methods.

Twelve principles of agile software development (ASD) have been defined by the Agile Alliance to supplement the Manifesto for agile software development (see Sect. 7.2). They are as follows (Holtz and Möller 2017):

- The highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business people and developers must work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- Continuous attention to technical excellence and good design enhances agility.
- Simplicity—the art of maximizing the amount of work not done—is essential.
- The best architectures, requirements, and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective and then tunes and adjusts its behavior accordingly.

Well-known ASD methods include (Holtz and Möller 2017):

- Agile modeling
- Agile unified process (AUP)
- Dynamic systems development method (DSDM)
- Essential unified process (EssUP)
- Extreme programming (XP)
- Feature-driven development (FDD)
- Open unified process (OpenUP)
- Scrum
- Velocity tracking

Automotive SPICE is a framework established in ECU development to improve and evaluate processes. The industry-specific standard was developed in 2005 based on ISO/IEC 15504 for software process assessments and adapted for requirements in the automotive sector. Automotive SPICE® V3.0 was presented to the public at the VDA Automotive Systems 2015 conference as a new standard. With this revision, Working Group 13, commissioned by the German Association of the Automotive Industry, ensured that Automotive SPICE is in harmony with the new ISO/IEC 12207 and ISO 15504-5. Therefore, in version 3.0, the previous engineering processes are divided into system and software processes. In addition, there are many improvements in the model. The German automotive manufacturers, Audi, BMW, Daimler, and Volkswagen, have agreed on a minimum subset of 16 processes (Hersteller Initiative Software or HIS Scope), which are assessed by each member in the software initiative. The HIS Scope of Automotive SPICE, as the minimum requirement of the processes to be considered in an assessment, is also used in other branches of industry as a starting point for process improvement and a focus for assessments.

ASAM MCD-2 (Association for Standardization of Automation and Measuring Systems) allows the data-oriented specification of vehicle diagnostics. The standard defines a data model for the description of diagnostics capabilities of ECUs needed throughout the life cycle of a vehicle from development to testing, production, aftersales, and service. The standard facilitates the exchange of diagnostic information between partners in the development process, e.g., between OEM and Tier 1 suppliers or between OEMs in a cooperation project. In detail, ASAM MCD-2 covers the description of:

- Diagnostic communication via requests and responses, diagnostic trouble codes, parameters, and other diagnostic data
- Communication parameters for different diagnostic protocols
- ECU memory programming
- ECU variant coding
- Function-oriented diagnostics

The standard defines a data model and description format which is independent of specific vendors, buses, or protocols and which has well-defined semantics for all specification elements. ODX allows the user to store diagnostic data in a central location and to efficiently distribute the data to all involved parties from a single source. ODX data is serialized in machine-readable Extended Markup Language (XML) format. Therefore, the standard enables the complete reuse of diagnostic data throughout all development phases of an ECU, e.g., for the design of diagnostic communication, the development of the ECU kernel and application software, configuration of diagnostic testers, and the generation of the diagnostics documentation for the vehicle. Since the data originates from one source, ODX helps to prevent inconsistencies, errors, and repetitive efforts (URL18 2017).

4.5.2 Model-Based Development

The complexity of automotive systems operations, especially safety functions, makes predicting safety performance extremely difficult in the product development process, which can be supported by the model-based development approach shown in Fig. 4.15. This approach focuses on the model of the vehicle systems and its corresponding control algorithms. A lot of care has to be taken to capture all relevant requirements and features of the vehicle systems, and extensive simulations are done to validate the overall closed control loop system functionality.

In a classical, nonmodel-based development scenario, the parameter of the controller defines the variables and constants for a piece of C-code that implements the control algorithm for the chosen target architecture. In model-based development, this step is automated. The dynamic model of the controller provides the input to software called an auto-code generator that directly produces the target C-code.

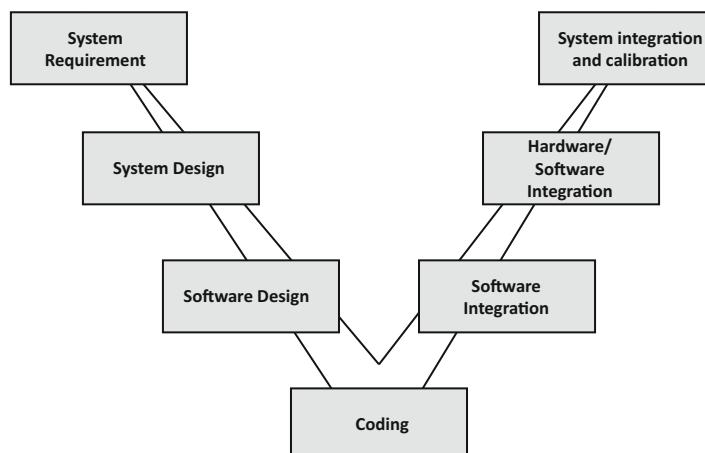


Fig. 4.15 Model-based design process

As the auto-code generator has been tested extensively, the resulting code is of excellent quality and minimizes standard programming errors, such as buffer overflow and others, which are often found in manually generated code.

- *Algorithm Design:* Simulation using tools such as MATLAB®/Simulink®, Advanced Simulation and Control Engineering Tool (ASCET), and LabVIEW
 - *MATLAB/Simulink:* MATLAB is the core product for numerical calculations. It is used in almost all industries and sciences for calculation and data evaluation. Simulink is the quasi-standard for the modeling of dynamic systems.
 - *ASCET:* ASCET provides software tools needed to successfully develop model-based application software and generate C-code. The notations, which ASCET uses for modeling, allow engineers to capture application software designs quickly and effectively. Through a combination of static analysis and testing, designs can be validated efficiently early in the development life cycle. The tool also supports model-based development, validation, measurement, and calibration of AUTOSAR-based ECU software.
 - *LabVIEW:* A graphical programming platform supporting engineers in scaling from design to test and from small to large systems. Integrating all tools that engineers need to build a wide range of applications in less time, LabVIEW is a development environment for problem solving, accelerated productivity, and continuous innovation.
- *Modeling and Simulation, Auto-Code Generation:* As a result of automated code generation technology, another application of these models has become viable which can serve as input to an automatic embedded code generation process. The code generated from these models is highly efficient, readable, testable, and suitable for use in safety-critical applications.
- *Rapid Prototyping With Porting Optimization:* A method that helps in validating numerical analysis results and shortening the development cycle, whereby it is necessary to choose the optimization parameters, which usually refer to cost, for each optimization cycle.
- *Model-in-the-Loop Testing:* The first representative of in-the-loop tests. For this test, a model of the ECU environment to be developed is required. The environmental model can have a high degree of abstraction, since, for example, sensors and actuators do not have to be completely modeled, their input and output behavior can be directly implemented.

Model-based development is a proven approach for efficiently developing solutions to complex engineering problems. It is a method for developing complex systems using mathematical models of system components and their interactions with the surrounding environment. These models have many applications in the development process, including system simulation, stability analysis, and control algorithm design specific to ECUs. Several off-the-shelf software tools for model-based development support the specification, design, and validation of high-reliable

ECU models for a wide range of system design in automotive applications. The resulting models can be as detailed as necessary, assuming that sufficient information is available to support the construction and validation of the model. One important application for high-fidelity ECU models is simulation. A model of the ECU for a high-fidelity vehicle is developed and integrated into the vehicle system simulation. By working with a system-level simulation that combines the vehicle and its respective ECU, it is possible to thoroughly test the control system design and rapidly make changes and improvements as needed.

Model-based development takes into account ISO 26262 a functional safety standard intended to be applied to the development of software for E/E systems in the automotive domain. ISO 26262 is an adaptation of the broader IEC 61508 safety standard. ISO 26262 provides an automotive safety life cycle, including management, development, production, operation, service, and decommissioning, and supports tailoring the necessary activities during these life cycle phases. Furthermore, it covers functional safety aspects of the entire development process, including such activities as requirements specification, design, implementation, integration, verification, validation, and configuration. ISO 26262 also provides an automotive-specific, risk-based approach for determining ASIL risk classes. It uses ASIL to specify an item's necessary safety requirements for achieving acceptable residual risk and provides requirements for validation and confirmation measures to ensure that a sufficient and acceptable level of safety is being achieved.

Today's software development, however, has big challenges in balancing the requirements for shortened total vehicle development time with longer development times for software, more stringent safety requirements, and, especially, growing complexity due to the rising number of functions and the increasing interaction between them. To master these challenges, automakers and suppliers are driving a paradigm change in software development from hand-coded to model-based development. The model-based development process is specifically attractive for automotive software as development in this domain is driven by two strong forces. On one side is the evolutionary development of automotive control systems, dealing with the iterated integration of new functions into a substantial amount of existing/legacy functionality from previous system versions, and on the other side the platform-independent development that substantially reduces the amount of reengineering/maintenance caused by fast-changing hardware generation. As a result, using a model-based approach in the development process in its early phases is being pursued to enable a shift in focus from function-based to code-based engineering of automotive systems. Model-based development is used by several automakers and suppliers even though no major empirical investigations into costs and benefits of model-based development have been conducted. Thus, the criteria for optimizing the costs and benefits of model-based development have to be identified first to determine the potential for further model-based development in development phases such as requirements engineering and E/E architecture design.

4.5.3 Hardware-in-the-Loop Tests

The functional and spatial distribution of ECUs (see Sect. 4.2.5) in today's vehicles has resulted in multifaceted innovations, such as DAS (see Sect. 4.2.4), ADAS (see Sect. 4.9 and Chap. 11), and other essential vehicle systems. A variety of tests ensures the reliability of these complex, real-time, networked vehicle systems to prevent potential malfunctions. With regard to complexity and real-time behavior, hardware-in-the-loop (HIL) test systems are used in the development and test of these complex real-time vehicle systems and have been established as a method for quality assurance of vehicle systems or components. HIL was introduced as a measure to improve the test possibilities in the automotive sector. Therefore, it can be seen as a method for testing and securing vehicle ECUs or mechatronic components and systems during development and early commissioning, providing an effective platform for the complexity of the vehicle ECUs or mechatronic components under test by adding a mathematical representation of the respective dynamic systems. The mathematical representation is referred to as the vehicle system simulation. In the automotive domain, HIL is applied in two main forms for the test:

- Adaptation of an ECU to a HIL simulator as a so-called component or module test bed
- Adaptation of several ECUs to one or several coupled HIL simulators as a so-called integration test bed

When performing HIL tests, the manually conducted tests in the initial phase are replaced by automatic test sequences performed with HIL test equipment (see Sect. 4.5.3.1). HIL includes electrical emulation of real hardware components from the vehicle for testing in a simulated sensor/actuator environment. These electrical emulations act as an interface between the real vehicle system and the vehicle system model representation under test. Hence, the scope of HIL tests is primarily aimed at establishing logically functional errors of the control software. Therefore, the value of each electrically emulated vehicle component is controlled by the system simulation and is read by the embedded system model under test. The overall test range at the HIL can be roughly divided into three categories:

- *Component Tests*: Deals with the function of a single ECU. These tests primarily show the functional specification of an ECU.
- *Integration Tests*: ECUs are tested in the network. The test focus is primarily on communication between the ECUs. Global functions are tested, the sub-functions of which are implemented and distributed over several ECUs.
- *Diagnostic Tests of Functions Implemented in the ECUs*: By generating faulty or implausible conditions, it is possible to test whether the ECU or the ECU interconnector detects these states and responds to them in accordance with the design specifications.

Apart from the pure connection of ECUs to a HIL environment, there is also a variant for the mechatronic components available. In this case, a part of the mechanics is integrated into the control loop. This approach is often used in conjunction with electronic steering systems, whereby a part of the steering rod is coupled as a real mechanic to the HIL environment.

In contrast to the test in the vehicle, the ECUs in the HIL test facility are fully embedded in a virtually simulated environment. Almost all parameters of this simulation environment can be changed, meaning that the test situation in which the vehicle is currently located does not depend on the actual external conditions, as in a driving test, but can be specified. Moreover, the vehicle characteristics are part of the simulation environment so they can be determined by changing the virtual vehicle parameters, such as weight, motorization, and other essential features, which has an influence on the HIL test results which can simply be checked. In case the HIL test system also permits a change in the ECU device coding, which depends on the chosen HIL test system, a set of different equipment variants can be tested with the same HIL test setup without great effort.

As an example of a HIL test, the platform for the development of a vehicle antilock braking system (see Sect. 4.2.2) has embedded mathematical representations for each of the following subsystems for the system simulation:

- Dynamics of the brake system's hydraulic components
- Road characteristics
- Vehicle dynamics, such as suspension, wheels, tires, roll, pitch, and yaw

The value of each electrically emulated vehicle component is controlled by the system simulation and is read by the embedded system model under test.

4.5.3.1 HIL Test System Architectures

As described in a white paper by National Instruments (NI) (URL19 2017), a HIL test system consists of three primary components:

- *Real-Time Processor*: The core of the HIL test system. It provides deterministic execution of most of the HIL test system components, such as hardware I/O communication, data logging, stimulus generation, and model execution. A real-time system is typically necessary to provide an accurate simulation of the parts of the system that are not physically present as part of the test.
- *I/O Interfaces*: Analog, digital, and bus signals that interact with the unit under test. They are used to produce stimulus signals, acquire data for logging and analysis, and provide the sensor/actuator interactions between the ECU being tested and the virtual environment being simulated by the model.
- *Operator Interface*: Communicates with the real-time processor to provide test commands and visualization. Often, this component also provides configuration management, test automation, analysis, and reporting tasks.

Many HIL test systems use hardware fault insertion to create signal faults between the ECU and the rest of the system to test, characterize, or validate the behavior of the device under these conditions. To accomplish this, fault insertion units (FIUs) inserted between the I/O interfaces and the ECU, as shown in Fig. 4.16, allow the HIL test system to switch the interface signals between normal operation and fault conditions, such as a short-to-ground or open-circuit constraint.

Some vehicle systems use multiple ECUs that are often networked together to function cohesively. Although each of these ECUs may initially be tested independently, a system's integration HIL test system, such as a full vehicle simulator, is often used to provide more complete virtual testing, as shown in Fig. 4.17.

Even with the latest multicore processing power, some vehicle systems require more processing power than what is available in a single HIL environment. To address this challenge, distributed processing techniques are used to meet the

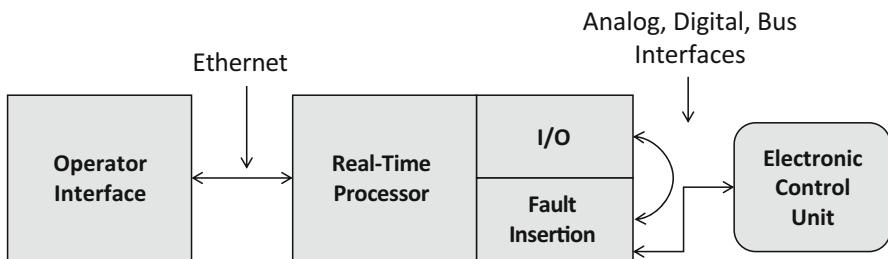


Fig. 4.16 Hardware fault insertion to test the behavior of the ECU during signal faults

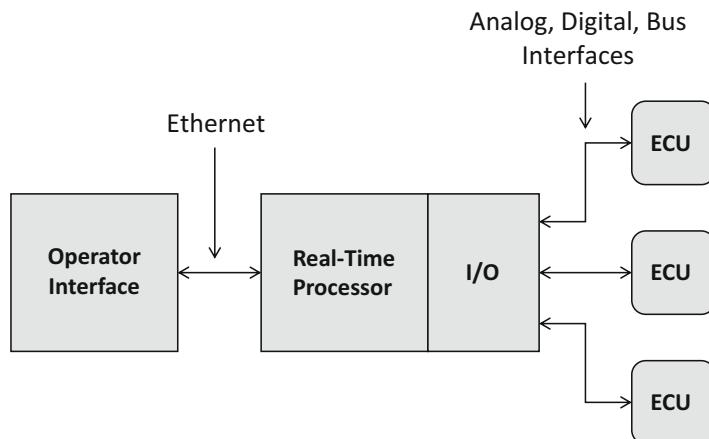


Fig. 4.17 Multiple ECU tests

performance requirements of these systems. In very high-channel-count systems, the need is more than simply additional processing power; additional I/O is also necessary. In contrast, systems using large, processor-hungry models often use additional real-time processors only for the extra processing power, allowing those processors to remain dedicated to a single task for greater efficiency. Depending on how the simulator tasks are distributed, it may be necessary to provide shared trigger and timing signals between the real-time processors as well as deterministic data mirroring to allow them to operate cohesively, as shown in Fig. 4.18.

Implementing and maintaining wiring for high-channel-count systems can pose costly and time-consuming challenges. These systems can require hundreds to thousands of signals be connected between the ECU and the HIL test system, often spanning many meters to compensate for space requirements.

Fortunately, deterministic distributed I/O technologies can help tame these wiring complexities and provide modular connectivity to ECUs, which allows for efficient system configuration modifications. Instead of routing all connections back to a single rack containing one or more real-time processing facilities instrumented with I/O interfaces, deterministic distributed I/O can be used to provide modular I/O interfaces located in close proximity to each ECU without sacrificing the high-speed determinism necessary for accurate simulation of the virtual parts of the system.

This approach greatly reduces the cost and complexity of HIL test system wiring by making it possible for the connections between the ECU and the I/O interfaces to be made locally (spanning less than a meter), while a single bus cable is used to span

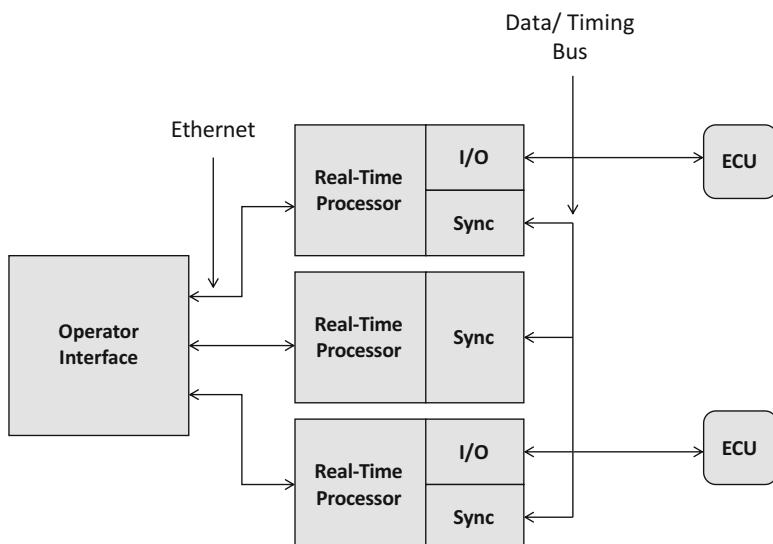


Fig. 4.18 Multiple real-time processors for additional processing power

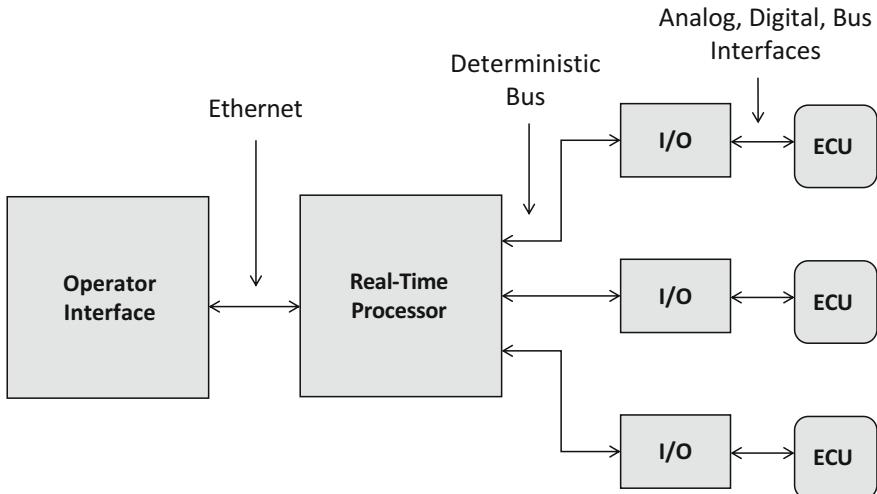


Fig. 4.19 Deterministic distributed I/O interfaces reduce HIL test system wiring cost and complexity because connections between ECU and I/O interfaces can be made locally

the additional distance to the real-time processing chassis. Additionally, with the modular nature of this approach, HIL test systems can easily scale incrementally, from a multi-ECU test system, in which all but one of the ECUs are simulated, to a complete systems integration HIL test system where none of the ECUs are simulated. The architecture behind this approach is shown in Fig. 4.19.

Once the appropriate architecture for the HIL test system is selected, the first step in creating a HIL test system is to select the components that best meet the vehicle development requirements. National Instruments (NI) provides a wide variety of real-time processing and I/O options for implementing HIL test systems. Because they are all based on open industry standards, users are assured that NI always delivers the latest advances in PC technology for the HIL test system and always meets future test system requirements. The NI HIL platform is open and extensible, which means that it can adapt to changing system requirements. Because of its modular architecture, the NI HIL platform can be easily upgraded with additional functionality, which helps in future proof test systems and meets the requirements of the most demanding embedded software testing applications. In addition to the widest range of I/O on the market, NI offers software tools that

- Automate HIL tests
- Perform post-processing and report generation
- Map test results to requirements.

with regard to

- Real-time Processing
- Analog/Digital I/O
- Fault Insertion
- Bus Interfaces
- Instrument Grade and RF I/O Vision/Motion.

These tools help to perform a wider range of tests earlier in the software development process, which reduces overall development cost while improving product quality ([URL19 2017](#)).

4.5.3.2 HIL Test System

dSPACE offers a wide range of HILs, HIL-specific hardware, and related software for ECU testing, such as HIL Simulator Full-Size and HIL Simulator Mid-Size, offering an easy way to update existing HIL systems. The Simulator Full-Size is a very versatile HIL, offering a comprehensive range of adaption and configuration possibilities to meet customer-specific requirements. It consists of one or multiple racks, is up to 41 height units tall, and uses standard processor and I/O cards, making adaptions easy. dSPACE Simulator Full-Size can be used for any application up to simulating a complete virtual vehicle and is shown in Fig. 4.20.

Fig. 4.20 dSPACE Full-Size HIL



Typical fields of application of dSPACE Full-Size HIL:

- Battery management systems
- Comprehensive closed-loop tests on ECUs, release/acceptance tests
- Electric motor simulation for hybrid or electric power trains and electric steering system
- Engine, power train, chassis, and body
- Mechanical test benches
- Networked ECUs
- Racing applications (Formula One, rally)
- Special requirements, e.g., with high system flexibility or high-current applications
- Truck applications

dSPACE Simulator Mid-Size generates and measures I/O signals via integrated dSPACE I/O boards. The function range is complemented by load and failure simulation. Typical fields of application of dSPACE Simulator Mid-Size HIL are:

- Automated testing
- Electric drives applications (combined with DS5202 Electric Motor HIL Solution or DS5203 FPGA Board)
- Engine, transmission, vehicle dynamics, and body electronics HIL
- Function integration tests, release tests, and ECU diagnostics tests
- Open-loop or closed-loop environment
- Realistic unit tests
- Real-time simulation

A standard dSPACE HIL Simulator Mid-Size also supports electrical failure simulation on all ECU output pins connected to the HIL I/O board. A hardware extension allows electrical failures to be simulated on ECU inputs as well. The host PC controls both types of failure simulation via an RS232 interface:

- Broken wire simulation (open circuit)
- ECU inputs optional by DS793/DS794 FIUs
- ECU outputs per load/FIUs
- Remote-controlled with ControlDesk® failure simulation and automated with AutomationDesk
- Simulation of cross-wired short circuits between ECU pins via common fail planes
- Simulation of short circuits: from ECU pins to ground or battery voltages
- Simultaneous activation of multiple failures (latch mode)

Another HIL simulation environment is SCALEXIO[®], the main dSPACE HIL simulator ranging from small to large systems and providing very high processing power. It is configured entirely by software, which makes adapting to changing requirements easy and simple. SCALEXIO can also be coupled with other dSPACE HIL systems, such as the HIL Simulator Full-Size and the HIL Simulator Mid-Size, offering an easy way to update existing HIL systems.

dSPACE SCALEXIO is a very versatile technology that provides highly flexible channels that can be extended to any required size and is completely software-configurable. Its application range covers all test domains, including the test of ECUs of electric drives. With regard to the SCALEXIO multiprocessing feature, the simulator can be coupled with existing SCALEXIO-based or DS100x-processor-board-based systems, allowing users to expand their existing test setups to meet growing project needs. The key benefits are:

- Easily resizable to fit specific test tasks because component test systems and network systems are both built with the same standardized hardware components and connections
- Graphical configuration of channels
- Support of different workflows and user roles by separating I/O configuration, modeling, and code generation
- Support of functional mock-up interface (FMI)
- Test of different ECU variants and types on a single system with minimal configuration effort
- Use of virtual ECUs (V-ECUs) for HIL tests if the real ECU prototype is not available yet

Furthermore, SCALEXIO contains an FIU consisting of several components:

- An onboard failure routing unit (FRU) on the I/O channels prepares failure simulation by switching the I/O channels to fail rails. The FRU is available for each channel on the MultiCompact and HighFlex boards and uses relays to provide the features of the central failure simulation unit to each channel.
- Depending on their properties, the channels are connected to the failure simulation system by the high-current (up to 80 A) or the low-capacitance (up to 1 A) fail rail. The low-capacitance fail rail for an optimized signal quality connects signal generation channels and bus channels to the central FIU. The high-current fail rail connects signal measurement channels to the central FIU.
- The central FIU is located on either the DS2642 FIU and Power Switch Board or the DS2680 I/O unit. The central FIU uses semiconductor switches for switching the failures. It switches very fast (pulsed switching), which makes it possible to simulate loose contacts or insert faults for a very precise duration.
- The fail rail segment switch is used to switch selected segments into the fail rails for failure simulation. This way, the conducting capacity can be minimized to avoid signal corruption, even for large simulation systems that have a high number of inputs/outputs or that are distributed across several cabinets.

The SCALEXIO FIU concept with an example of HighFlex I/O boards is shown in Fig. 4.21. The available failure types of the SCALEXIO FIU are listed in Table 4.5.

The dSPACE tool chain also allows rapid prototyping and ECU validation with virtual test drives covering the following applications:

- Rapid control prototyping
 - Predictive drivetrain control for commercial vehicles
 - Autonomous emergency braking based on radar and camera data
 - Automatic windshield wiper control and rain sensor

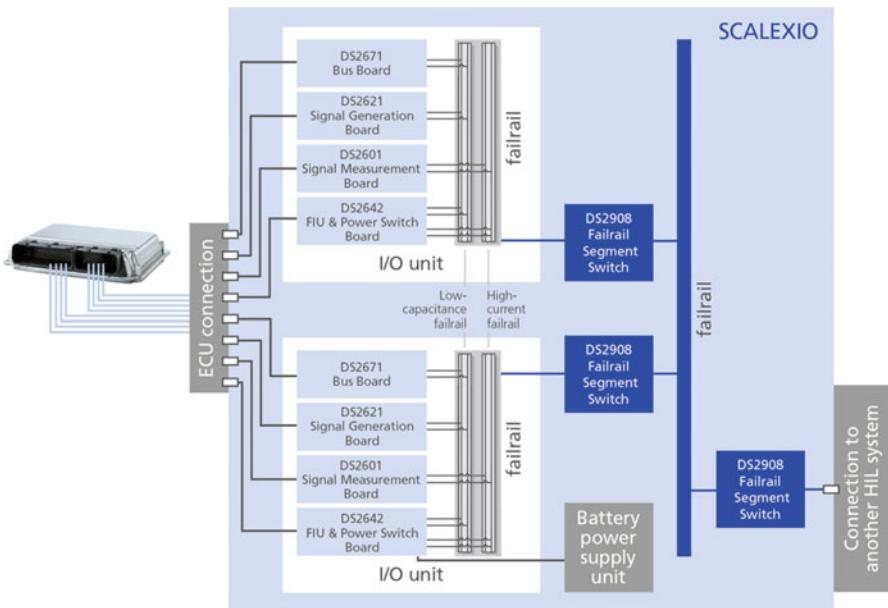


Fig. 4.21 SCALEXIO FIU concept with a selection of HighFlex I/O boards

Table 4.5 SCALEXIO failure types

Failure type	Failure on single signals	Failure on several signals
Open circuit	1 channel	All channels ^a
Short circuit to ground or U_{BAT}	2 channels	Up to 10 channels ^{a,b}
Short between channels	2 channels	Up to 10 channels ^{a,b}
Failure with pulsed switching	✓	

^aRequires the option “Activation by FRU relay” and is only possible on I/O channels without current enhancement

^bDepending on the ampacity of the fail rail

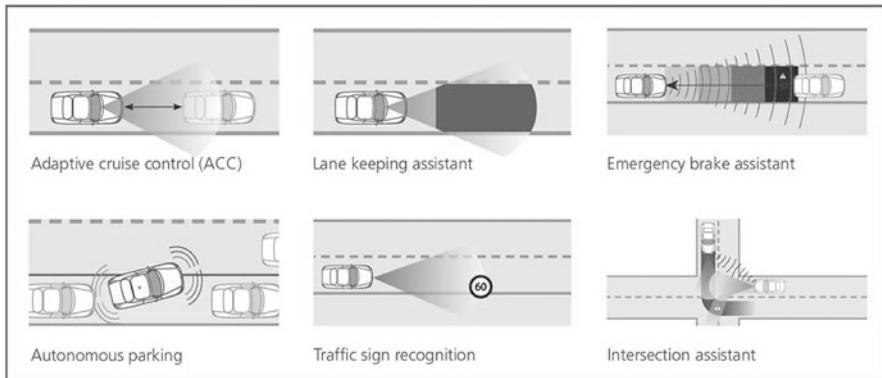


Fig. 4.22 dSPACE practical examples for virtual validation and hardware-in-the-loop simulation

- Virtual validation and hardware-in-the-loop simulation
 - Adaptive cruise control
 - Lane keeping assistants
 - Pedestrian detection
 - Traffic sign recognition
 - Intersection/cross-traffic assistant

These use case examples are illustrated in Fig. 4.22 used with the permission of dSPACE ([URL20 2017](#)).

4.6 AUTOSAR

The electric/electronic (E/E) architecture landscape in the automotive industry was characterized by proprietary solutions in the past, which seldom allowed the exchange of applications between automakers, and Tier 1 suppliers. With regard to the continued exponential growth in complexity and functionality of E/E components and systems, further proliferation of proprietary solutions consumes more and more resources and becomes difficult to control. This has resulted in an industry-wide initiative to manage the complexity of emerging automotive E/E architectures, the so-called AUTomotive Open System ARchitecture (AUTOSAR). AUTOSAR is a worldwide joint initiative of several major industries which was formed in mid-2003 to create and establish standardized software architecture for automotive ECUs. It serves as a platform upon which future vehicle applications can be implemented and also serves to minimize barriers between functional domains. Development goals include scalability to different vehicle and platform variants, transferability of software, consideration of availability and safety requirements, collaboration between various partners, sustainable utilization of natural resources, maintainability throughout the whole product life cycle, and process management

during the entire life cycle of a product from inception, through engineering design and manufacturing, to service and disposal of manufactured products. AUTOSAR is driven by the advent of innovative vehicle applications, contemporary automotive E/E architecture that has reached a level of complexity requiring a technological breakthrough in order to manage it satisfactorily and fulfill the heightened passenger and legal requirements. This need is important for vehicle manufacturers and their leading Tier 1 suppliers who are faced with often conflicting requirements from:

- *Driver Assistance and Dynamic Drive Aspects*: Key items include detection and suppression of critical dynamic vehicle states and navigation in high-density traffic surroundings.
- *Legal Enforcement*: Key items include environmental aspects and safety requirements.
- *Passenger Convenience and Service Requirements*: Comfort and entertainment functional domains.

Leading OEMs and Tier 1 suppliers, having recognized this industry-wide challenge, decided to work together to meet the challenge. Their common objective is to create a development base for industry collaboration on basic functions while providing a platform which continues to encourage competition on innovative functions. To this end, a development partnership called AUTOSAR was formed, including all vehicle domains with the goals of ([URL21 2017](#)):

- Collaboration between various partners
- Definition of an open architecture
- Development of highly dependable systems
- Scalability to different vehicle and platform variants
- Standardization of basic software functionality of automotive ECUs
- Support of different functional domains
- Support of applicable automotive international standards and state-of-the-art technologies
- Transferability of software

The AUTOSAR standard serves as a platform upon which future vehicle applications will be embedded and also serves to minimize the current barriers between functional domains. It will, therefore, be possible to map functions and functional networks to different control nodes in the system, almost independently from the associated hardware. The technical goals of AUTOSAR:

- Modularity of automotive software elements to enable tailoring of software according to the individual requirements of ECUs and their tasks.
- Reusability of functions to help improve product quality and reliability and to reinforce corporate brand image across product lines.

- Scalability of function to ensure the adaptability of common software modules to different vehicle platforms and prohibit proliferation of software with similar functionality.
- Transferability of functions to optimize the use of resources available throughout a vehicle's electronic architecture.

This helps to provide a common software infrastructure for automotive systems of all vehicle domains based on standardized interfaces for the different layers, as shown in Fig. 4.23. This common infrastructure encompasses the following elements:

- *Electronic Control Unit (ECU)*: The physical hardware.
- *Runtime Environment (RTE)*: All communication between software components and basic software including the operating systems (OS) and communication services is carried out through the RTE layer.
- *Main Software*: A combination of:
 - *Basic Software*: Builds on RTE to offer some general utilities which provide the overall functionality of the AUTOSAR infrastructure (software components and RTE on an ECU). Basic software is essential for running the functional part of the software; however, it does not fulfill any functional job itself. The software components do that.

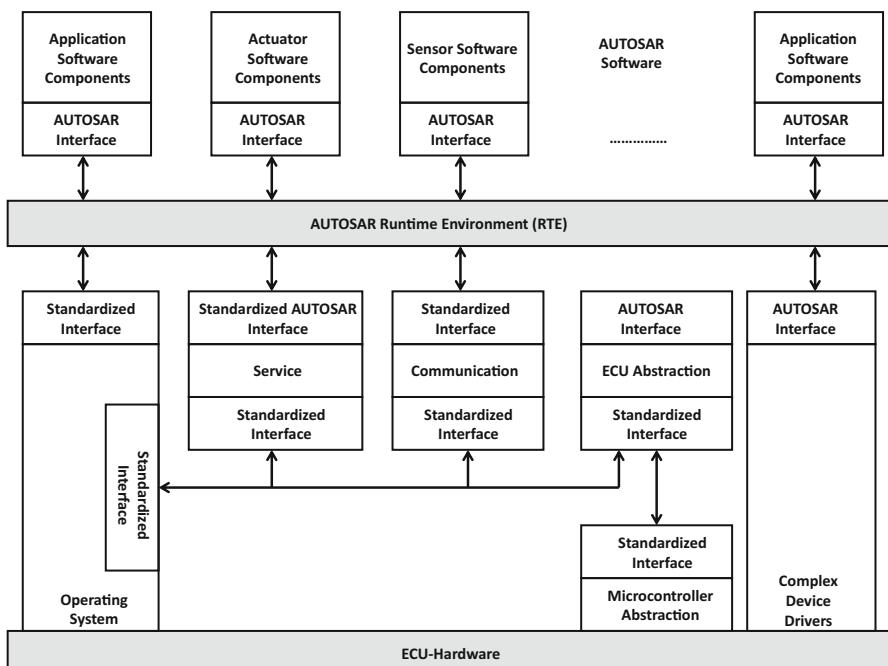


Fig. 4.23 AUTOSAR ECU software architecture (modified after ([URL21 2017](#)))

- *Software Components:* Building blocks for software systems, either custom made or purchased off-the-shelf, each supporting and implementing a dedicated set of functionalities, and, in conjunction, providing the overall functionality of the software application. Software components are the fundamental building blocks of AUTOSAR systems. Types of software components are:
 - Application software components
 - Actuators/sensor software components
- *Complementary Software:* Manufacturer- and model-specific software.

The AUTOSAR RTE is the central connecting element in an AUTOSAR ECU architecture. It realizes the interfaces to enable interaction between any kind of AUTOSAR software components. Each ECU component has its own customized RTE implementation. Depending on the location of each component, the former virtual interaction can then be mapped to a real interaction implementation. Components that are mapped onto one ECU communicate through intra-ECU mechanisms. Since the RTE source code is generated, it can be tailored by the generator to implement the communication paths required by its connected AUTOSAR components. Thus the RTE can be interpreted as a static implementation of specialized communication topologies.

The ECU abstraction layer provides a unified interface for AUTOSAR software components to access electrical values of the underlying ECU independently of the actual ECU hardware architecture. The ECU abstraction itself is closely coupled to the microcontroller abstraction layer that provides access to the actual physical signals of the microcontroller. The microcontroller abstraction layer is a hardware-specific component available on each standard microcontroller and provides the basic software to access hardware information without directly accessing the microcontroller's registers. Among others, MCAL provides access to digital I/O, analog/digital converter, flash, electrically erasable programmable read-only memory (EEPROM), and others.

Hence, standardization of functional interfaces across automakers and suppliers and standardization of the interfaces between the different software layers is seen as a basis for achieving the technical goals of AUTOSAR. AUTOSAR provides a standard description format for the interfaces as well as other aspects needed for the integration of the AUTOSAR software components. Key ECU automotive software elements are:

- *Operating System:* Task scheduler (event, take place on a regular basis, etc.)
- *Application:* Supports normal power train operations; diagnostic, calibration
- *Network:* Communication, data transfer, OEM network strategy (Ford Network Operating System, General Motor Local Area Network, etc.), and data transfer

Some principal classical challenges and solutions suggested by AUTOSAR, together with their implied benefits, are listed in Table 4.6 (Heinecke et al. 2003).

Table 4.6 Challenges, solutions, benefits of AUTOSAR

Challenges	Solutions	Benefits
Non-competitive functions have to be adapted to OEM-specific environments	Standardized interfaces	Reduction/avoidance of interface proliferation within and across OEMs and suppliers
Tiny little innovations cannot be implemented at reasonable effort as provision of interfaces from other components requires a lot of effort		Eases implementation of hardware-independent software functionality by using generic interface catalogs
Missing clear interfaces between basic software and code generated from models		Simplifies model-based development and makes it usable for standardized AUTOSAR code generation tools Reusability of modules across OEM Exchangeability of components from different suppliers
Effort wasted on layout and optimization of components which add no value recognized by customer	Basis software core	Enhanced software quality Concentration on functions with competitive value
Obsolescence of hardware (μ C, circuits, . . .) causes huge efforts in adapting existing software	Microcontroller abstraction	Part of the hardware can be exchanged without need for adaptation of higher software/functions/applications
Extended needs for microcontroller performance (caused by new functions) causes need for upgrade, i.e., redesign effort		
Large effort when relocating functions between ECUs	Runtime environment	Encapsulation of functions creates independence of communication technology
Large effort when reusing functions		Communication easier through standardized mechanisms Partitioning and relocatability of functions possible
Immature processes because of acting in ad hoc mode/missing traceability of functional requirements	Software component template	Improvement in specification (format and content)
Lack of compatible tooling (supplier, OEM)	Exchange formats	Opportunity for a seamless tool chain
OEM buys black box and is not able to extend/integrate new functionality in an ECU (e.g., integration of tire guard functionality)	Technical integration of software of multiple suppliers	Eased process of integration of different software components allows optimization of hardware costs
Lack of guidelines for use/buy of software components	Conformance test process	Integration of third-party software components
Unclear legal situation	License agreement	Common understanding between suppliers and OEMs

4.7 AUTOSAR Adaptive Platform

In the near future, domain controllers will be enhanced with multicore processors in vehicles used for computing-intensive vehicle applications. Moreover, the vision for autonomous driving also requires such domain controllers which results in more sophisticated designs. Adaptive AUTOSAR, the next generation of AUTOSAR, is software on which such designs can be based.

The AUTOSAR Adaptive Platform is designed to support software engineers creating more flexible E/E architectures. For this reason, the AUTOSAR Adaptive Platform will provide a software framework for more complex vehicle systems. Engineers will be supported by an increase in bandwidth, the result of implementing Ethernet networking technology “to provide an optimal standardized software framework for new applications, especially in the fields of connectivity and highly automated and autonomous driving,” said Stefan Rathgeber, spokesperson for the AUTOSAR development partnership ([URL22 2017](#)). Thus, classic and adaptive applications can be seamlessly combined using an Ethernet connection.

Therefore, the new standard will probably be first applied in an ADAS. However, “highly automated driving systems must be dependable and have fail-safe operational capabilities,” Rathgeber explained. This can only be accomplished with features such as high data processing capacities, service-oriented communication, and over-the-air updates. The driverless vehicle is where the new platform’s strengths will ultimately provide the greatest benefits, being a key enabler on the way to a self-driving vehicle by making the new platform accessible to as many manufacturers, suppliers, and developers as possible. It can also aid infotainment system development by providing a more seamless integration into a standard operating system with more connectivity and graphics computing power.

Among the development committee’s goals is to create a dynamic system that includes middleware and supports complex operating systems using a POSIX interface and multicore microprocessors. Its main communication approach is based on service-oriented communication and IP/Ethernet. The platform will be capable of supporting adaptive software deployment while interacting with non-AUTOSAR systems, as shown in Fig. [4.24](#).

4.8 GENIVI

Compared to AUTOSAR, the non-profit GENIVI Alliance is committed to driving the broad adoption of specified, open source, IVI software. Therefore, GENIVI provides automakers with four unique approaches to meeting today’s challenges:

1. *Define*: Allows flexible definition of IVI systems that fit customers’ latest needs
2. *Partner*: Supports business model evolution and networking across the supply chain
3. *Leverage*: Provides standard, open source architectures, tools, and software components
4. *Reuse*: Allows reuse of components and redeployment of solutions with no royalty fees

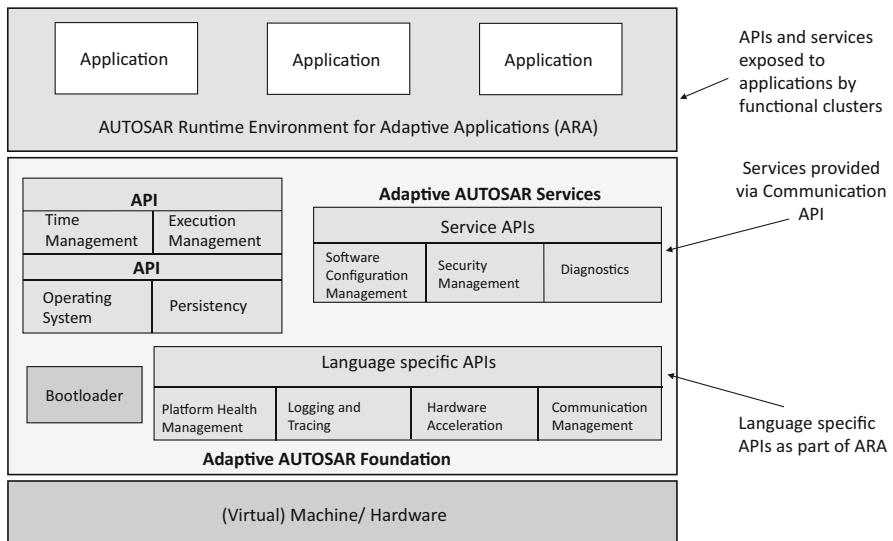


Fig. 4.24 Adaptive AUTOSAR platform

Automakers and their suppliers face at least three significant challenges in developing and delivering IVI functionality to their customers:

- **Responding to Consumers:** Consumers want IVI functionality that is the same or similar to that found in consumer electronic devices, such as smartphones and tablets. New devices with the latest features are typically launched in the market on an 8- to 18-month cycle versus the 2–5 year cycle for most in-vehicle software. As a result, consumers have introduced a new competitive measure that automakers must use: the time from consumer request to in-vehicle availability.
 - GENIVI's open software approach better aligns consumer electronics and automotive development cycles.
 - GENIVI's individual software components and reusable platform provide automakers and their suppliers with the tools to perform rapid prototyping and to quickly develop and deliver IVI systems that fulfill consumer requests.
- **Complexity and Cost:** Consumer functionality requests push the amount of software in a typical IVI system to over several million lines of code. Hence, automakers have to deal with the increasing complexity and cost of developing, validating, and maintaining software. Many automakers are shifting away from the historic black box approach and are taking more ownership of the design and development process, including maximizing the reuse of legacy code to reduce costs and deploying a software platform on multiple hardware platforms based on the needs of their various models.
 - GENIVI's technical deliverables and open approach promotes a wide range of supplier models based on the preferences of the automaker.

- Automakers can launch a single reusable software platform that, with limited integration, can run on a wide range of automotive boards, from low- to high-end performance.
- *Customer Ownership:* Automakers are keen to keep their customer relationships sustainable. Large technology companies, such as Apple and Google, have entered the automotive market, introducing demands for user experience, branding, and data usage that limit the automaker-driver relationship. Automakers have their own business model; some prefer a single Tier 1 supplier, while others prefer multiple suppliers taking ownership of certain pieces of the overall system.
 - GENIVI's approach allows automakers to maintain their independence from technology titans by pushing their own business models in the automotive industry.
 - GENIVI's flexible architecture and pick-and-mix model gives automakers the freedom to include preferred, best-in-class software from multiple suppliers.

GENIVI's technical deliverables consist of:

- Flexible technical architecture
- Individual software components
- Pre-integrated, reusable IVI platform
- Standard interfaces/application programming interfaces (APIs)

that are essential to overcoming the IVI challenges faced by every automaker. Thus, GENIVI technologies is at the forefront of a new generation of IVI solutions. As one of the many GENIVI use cases, BMW has moved from its traditional approach to IVI software development to where it is today, the first automaker to deliver a complete infotainment product, the so-called entry media and navigation system (EMNS). EMNS rolled off the assembly line in the fall of 2013 and is now in the MINI and 1, 3, and 5 BMW series product lines based on the GENIVI Linux platform. Since then, other automakers have selected products with GENIVI solutions resulting in vehicles on the road in four continents around the world. Furthermore, several additional automakers will release GENIVI-equipped systems in vehicles during the next 2 years.

4.9 Example: Advanced Driver Assistance System

Advanced driver assistance systems (ADAS) (see also Chap. 11) support vehicle drivers in the driving process by enhancing it for safety and better driving conditions. Therefore, safety features are implemented to avoid collisions or accidents by embedding intelligent safeguard devices and taking over control of the vehicle in critical driving or traffic situations. In this regard, the ADAS development process

began with the definition and specification of functional requirements in terms of the desired safety functions being embedded ride comfort, and operational restrictions. The primary functionality of ADAS is to facilitate the task performance of drivers by providing them with:

- Instructions
- Real-time advice
- Warnings

This type of driver support operates in different kinds of modes (Rosengren 1995), such as:

- Advisory mode
- Automatic mode
- Semiautomatic mode.

All of them have different consequences for the driving task and hence on vehicle and traffic safety. Thus, the purpose of ADAS is to reduce or even eliminate driver errors, resulting in an enhanced efficiency in driving the vehicle. Therefore, the benefits of ADAS are high because of a significant decrease in human suffering, economical costs, and less pollution because:

- Drivers using ADAS will be safe and efficient drivers.
- Driving safety will be considerably enhanced.
- High-performance driving is possible without regard to vision, weather, and environmental constraints.
- More vehicles will be accommodated on regular highways but especially in dedicated lanes.
- Other essential features of safe driving.

In this regard ADASs are safety-critical systems that require a high degree of:

- *Fault Tolerance*: Enables a system to continue its intended operation in the event of a failure of one or some of its components, possibly at a reduced level, rather than failing completely.
- *Real-Time Behavior*: A real-time system that executes tasks to completion in a guaranteed amount of time.
- *Reliability*: The ability of a system or component to perform its required functions under stated conditions for a specified time.
- *Security*: The degree of resistance to, or protection from, harm.

Functional and safety requirements can be represented by the system specification in order to define the exact operation of safe system functionality. Therefore, the

system specification represents the basis for the top-level design of the system architecture, followed by a detailed module design of:

- Actuators
- Controller
- Driver interfaces
- Human-machine interface (HMI)
- Sensors
- Other essential components

After implementation of the various hardware and software modules, the system will be carried out by integrating the individual modules, bringing overall system functionality and safety together. In each integration step, verification is performed to determine whether or not the output of a step meets the design specifications because ADAS relies on inputs from multiple data sources. However, additional inputs are possible from other sources separate from the primary vehicle platform, such as other vehicles, referred to as V2V or V2I, and the vehicle-to-X (V2X) systems.

More in general, ADAS (see Chap. 11) is one of the fastest-growing segments of automotive electronics with industry-wide standards in vehicular safety systems, such as ISO 26262, and the developing technology specific standards, such as IEEE 2020 for image sensor quality, and communication protocols, such as the vehicle information API, as reported in ([URL23 2017](#)). In general, an API is a set of routines, protocols, and tools for building software-based applications, specifying how software application programs should interact.

ADAS design can be done through so-called model-in-the-loop (MIL) testing and simulations to abstract the ADAS behavior in a way that the developed ADAS model can be used for testing, simulating, and verifying. Using an industry standard, such as Simulink, for model definition enables the engineer to test and refine the model within a desktop environment, allowing a complex system to be developed efficiently ([URL24 2017](#); [URL25 2017](#)). The code can subsequently be used with software-in-the-loop (SIL) simulations verified by the remaining hardware components, vehicle dynamics, and simulation of the real-time environment. Finally the hardware can then be tested by a real-time (HIL) simulation (see Sect. 4.5.3).

4.9.1 ADAS Functionalities

With regard to:

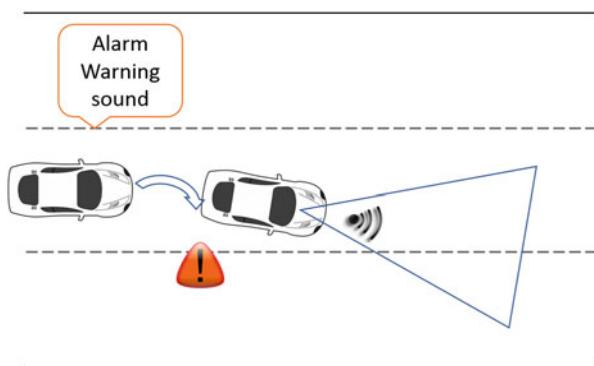
- Advanced functions
- HMI
- Information distributions
- Sensors and actuators
- Systems software and hardware

ADAS can cover a variety of functionalities, and many similar systems in different parts of the world often exist with a slightly different name (see Chap. 11). The systems functionalities mentioned on the website of the ADASE2 cluster (URL26 2017) and some additional ones are listed alphabetically below:

- *ACC/Stop and Go and Foresight (ACC/SaG)*: Ensures that during a stop and go traffic situation, the longitudinal control of a vehicle will be partly carried out by detecting the traffic in front of the vehicle in the near field. In extension to an ACC system, the detection of the near-field area is necessary to react to other vehicles swerving into the near field as they react to the more far away traffic situation of the vehicles ahead. Hence, the near-field communication at the end of a traffic jam can be included into the longitudinal control of a vehicle before the driver is able to see it.
- *Automatic Parking (AP)*: Ensures that a driver entering into a parking slot in a parallel maneuver is supported by automatically taking over the steering and engine control
- *Autonomous Driving (AD)*: Ensures that driving is safely controlled in every situation by an algorithm.
- *Autonomous Emergency Breaking (AEB)*: System that avoids a collision with another vehicle or a pedestrian.
- *Blind Spot Detection (BSD)*: Sensors monitor the road area behind and next to the vehicle and warn if the driver tries to pull out despite there being no gap, taking much of the strain off the driver and avoiding hazardous situations.
- *Intelligent Headlight Control (IHC)*: Ensures optimum illumination of the road. The improved vision makes driving at night much safer and more comfortable. The system uses a video camera to measure the ambient brightness and to estimate the distance from vehicles in front and oncoming traffic. This data is used to implement a variety of light functions. The high beam activation function enables drivers to use their high beam lights as often as possible without having to manually switch them on and off. If the function does not detect any other vehicles, it activates the high beam lights. However, if a vehicle is detected, the high beam light is switched off again. The adaptive high beam control function enables variable adjustment of the high beam range between the low beam and high beam levels. The area between the vehicle in front and an oncoming vehicle is better illuminated since the headlight cone is continuously adapted to this distance. With the continuous high beam control function, the driver can travel with permanent high beam headlights. If the camera detects other vehicles, the headlights are tilted horizontally or vertically, independently if necessary. This produces cones of light in which other road users are blocked out. The light distribution from the high beam lights remains virtually unchanged, while the driver's visual range is increased considerably.
- *Intersection Support (IS)*: Ensures certain tasks are supported, such as approaching stop signs, traffic lights, cross-traffic, and other tasks which result in IS system complexity due to the manifold scenarios which have to be supported by the system making detection and interpretation very difficult.

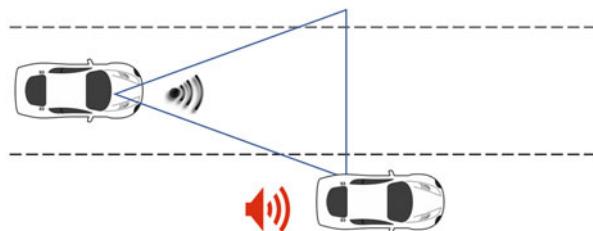
- *Lane Change Assistant (LCA)*: Ensures the driver is warned before and during a dangerous lane change. Through an acoustic and visual warning system, as well as a haptic feedback from the steering wheel, the driver is supported following a lane change trajectory. This requires detection of all other vehicles around the vehicle as well as the detection of the lane. LCA consists of several combined systems, such as LDW and BSD. Functional limits of LCA systems are apparent in the case of fast approaching vehicles.
- *Lane Departure Warning (LDW)*: This Continental-developed driver assistance system alerts the driver with acoustical or haptic warnings before the vehicle is about to leave the lane. According to a study carried out on behalf of the German Federal Ministry of Education and Research, LDW could prevent about half of the accidents caused in this way, as reported in (URL27 2017). In Fig. 4.25, the LDW system function is depicted, based on the example given in (URL27 2017).
- *Lane Keeping Assistant (LKA)*: Responds through a gentle intervention in the steering, which the driver can counteract at any time. This can save additional reaction time in cases where each and every second counts.
- *Local Hazard Warning (LHW)*: In a case where a hazard occurs too far in front of the vehicle for the driver to see it, the LHW system will warn the driver by communicating hazard information, e.g., an accident far down the road is passed along over long distances by using ad hoc networks.
- *Near-Field Collision Warning (NFCW)*: Detects vehicles in the near field, such as in the blind spot area (see BSD). The detection area is very close to the vehicle in the near field. Driver warning can be acoustical, haptical, or optical.
- *Night Vision Plus (NVP)*: Offers the driver a true-to-life image of the road ahead and provides valuable information about the course of the road, vulnerable road users, and obstacles on and alongside the road. When the system identifies pedestrians, they are highlighted clearly in the night vision image. Night vision plus directs the attention of the driver to potential risks, allowing the driver to take appropriate action.
- *Obstacle and Collision Warning (OCW)*: The driver will be warned if a potential collision is detected with, e.g., another vehicle or obstacle. The warning can be acoustic or visual. Complex scenarios, such as evading, can be included as well as warning breaking, which is a very short brake in order to give a kinesthetic feedback.

Fig. 4.25 LDW driver assistance system detecting vehicle's lane-following behavior (URL27 2017)



- *Platooning*: Ensures that several vehicles, e.g., trucks, that are following one after the other in a platoon in order to save space are connected electronically by means of communication.
- *Pre-crash Collision and Mitigation System (PCCMS)*: Ensures damage reduction from an accident by acting on the pretensioner of the safety belts before the accident occurs and automatically starting to brake when the system detects an upcoming collision which cannot be avoided.
- *Rear View System (RVS)*: Many new vehicle models allow drivers to see very little of the vehicle's immediate environment. As aerodynamics and pedestrian protection issues exert an ever greater influence on vehicle contours, while side and rear windows shrink in size, it is becoming almost impossible to maneuver cars safely and precisely. The Bosch rear view system supports drivers as they reverse their vehicle. The camera image is displayed via the radio or radio navigation system and shows the area behind the vehicle.
- *Road Departure Protection (RDP)*: The Continental RDP avoids roadway departure crashes, which currently are not completely covered by today's lateral guidance ADAS. The Continental base system uses a forward-looking mono camera to detect roadway boundaries, monitor the driver's steering angle and vehicle path through existing ESC sensors, and use chassis motion sensors to identify if the vehicle is crossing the road boundary. It then uses the existing ESC system to apply the individual wheel brakes to automatically steer the vehicle back on the road while simultaneously warning the driver and reducing the speed of the vehicle for safety reasons. This active intervention is signaled when the vehicle senses it is departing from the road. The system is designed with a driver intention recognition feature in the event that the driver does intend to leave the roadway for any reason (URL28 2017). In Fig. 4.26, the LDW system function is shown, based on the example given in (URL28 2017).
- *Road Sign Recognition (RSR)*: Drivers should always be aware of the current speed limit. With road sign recognition, currently applicable road signs are in view. When a video camera identifies a road sign signaling the beginning or end of a speed limit as well as any special instructions, such as slippery when wet, the function displays this sign in the form of a symbol in the cockpit. The speed limit on variable message signs as well as restrictions on overtaking and the end of such restrictions can also be detected. If the driver fails to observe the speed limit, he

Fig. 4.26 RDP driver assistance system detecting vehicle's lane-following behavior (URL28 2017)



can, for example, be warned by an audible signal. In the future, it will also be possible to detect other road signs in addition to speed limits.

- *Rural Drive Assistance (RDA)*: Ensures that systems developed for use on highways also work on rural roads which require the extension of some system functionalities of the highway systems.
- *Traffic Sign Recognition (TSR)*: Has a display on the instrument panel to remind drivers of the current speed limit. This is achieved through multiple uses of the same camera which is also used for the lane departure warning system. When combined with high-performance software, it can also recognize speed limit signs. Digitized speed limit information of the onboard navigation system will be incorporated to prepare for roads without assigned speed limit signs.

It has to be mentioned that the previous alphabetic list of functionalities of ADAS is, in some cases, related to a research roadmap. Obstacle and collision avoidance, platooning, and autonomous driving are still being researched and will be developed for use in the near future in conjunction with the development of the required sensors for the aforementioned advanced functionalities. In addition, ADAS and pedestrian protection systems (PPSs) have become an active research area aimed at improving traffic safety. In this regard, the major challenge of PPSs is the development of reliable onboard pedestrian detection systems. Due to the varying appearance of pedestrians with regard to the following, it is very difficult to cope with the robustness needed for this kind of protection system (Gironimo et al. 2009).

- Aspect ratio
- Different sizes
- Different types of clothing
- Dynamic shape
- Unstructured environment

Thus, the problems arising in this research area are the lack of public benchmarks and the difficulty in reproducing many of the proposed methods, which makes it difficult to compare the approaches. Hence, a more convenient strategy for surveying the different approaches will be dividing the problem of detecting pedestrians from images into different processing steps, each with attached responsibilities. Then, the different proposed methods can be analyzed and classified with respect to each processing stage, favoring a comparative viewpoint.

4.9.2 ADAS Sensor Types

Sensors are designed for specific application domains in which they work over a specific range. The design range is usually determined with regard to the application, ensuring safe and precise measures. The reason is that if the measuring range is exceeded, the sensor may be permanently damaged or destroyed. More in general, a

sensor is a device that generates a measurable signal in response to a stimulus received from the following:

- Components
- Objects
- Systems

The characteristics of a sensor can be classified as being either static or dynamic, which is important in high-fidelity mapping of output versus input signals. Static characteristics are those measured after all transient effects have stabilized to their final or steady state. In contrast, dynamic characteristics describe the sensor's transient properties.

ADAS require different types of sensors such as vision and range sensors to accurately determine situational assessment and action implementation. Common sensor technologies for ADASs, which are being increasingly integrated by OEMs and Tier 1 suppliers, are:

- Infrared camera
- Lidar
- Radar
- Ultrasonic
- Video

According to market analysts, the use of forward-looking cameras will go up from 30 million units in 2014 to nearly 100 million by 2019. However, range sensors, which are based on radar and, more recently, on LiDAR technologies, are projected to be much more available in vehicles within the next 2 years. Furthermore, ongoing technology and integration developments in many ADAS application areas are increasing the design alternatives. Thus, OEMs and Tier 1 suppliers need to continually evaluate their systems, deciding how and when to integrate the newest technologies and latest advancements into their designs. In order to be able to use the sensor signals correctly, the operation of the respective sensor and the nature of the signals they generate must be clearly understood. With regard to this knowledge, engineers must be able to use the right approach for data acquisition from the sensor.

Looking at the manifold ADASs (see Sect. 4.9.1), a lot of different types of sensors are present, such as traditional technologies like cameras, radar, and ultrasound, which display various limitations in the context of ADAS applications, from sensitivity to weather conditions to the ability of reliable detection of objects. LiDAR, an advanced sensor technology for ADAS, remains cost prohibitive even at high volumes and may lack the robustness required for automotive applications.

Existing fixed-beam LiDARs are more robust than their scanning counterparts; however, they also entail major limitations in terms of distance range. A comparison of the main automotive detection and ranging technologies is shown in Table 4.7 (URL29 2017). To gain the respective knowledge about ADAS sensor operation and the nature of the signals they generate, the most important ones are discussed in more detail in the following subsections.

Table 4.7 Comparison of main automotive detection and ranging technologies

	Ultrasonic	Camera	Camera	Radar	LiDAR	Laser
Impact of lighting conditions	None	High		None	Low to medium	
Impact of weather conditions	High			Low	Medium	
Field of view type	Short and wide		Far and narrow		Short and wide	Far and wide
No moving part design	Yes			No		
Pedestrian detection	Limited	Yes		Limited	Yes	
Stationary object detection	Yes			Limited	Yes	

4.9.2.1 RADAR Sensor

Radar is the acronym for radio detection and ranging. A radar system operates in the ultrahigh frequency (UHF) or microwave part of the radio-frequency (RF) spectrum and is used to detect the position and/or movement of objects, such as vehicles. Radar waves are transmitted at defined intervals. The delay between the transmitted wave and the echo determines the radial position for each azimuth direction on the display used. The greater the echo delay from a particular object in space, the farther from the display center it appears. Radar systems in vehicles are responsible for the detection of potential collisions or hazardous situations. A positive detection can be used to warn/alert the driver or to intervene with the braking and other controls of the vehicle in order to prevent an accident. In its practical realization, a vehicle radar system contains one or more radar sensors to detect obstacles around the vehicle and their speeds relative to the vehicle. Based on the detection signals generated by the sensors, a processing unit determines the appropriate action needed to avoid the collision or to reduce the collateral damage.

Using the vehicle radar system as a decision-making unit, it can:

- Alert the driver about any potential danger
- Assist the driver in parking the vehicle
- Prevent collisions by intervening with the control of the vehicle in hazardous situations
- Take over partial control of the vehicle such as adaptive cruise control

The key performance parameters of a vehicle radar system are:

- Angular resolution
- Angular width of view
- Detection range
- Range precision
- Speed detection range
- Velocity precision

With regard to the aforementioned characteristics, vehicle radar systems can be divided into three subcategories:

- Short range
- Midrange
- Long range

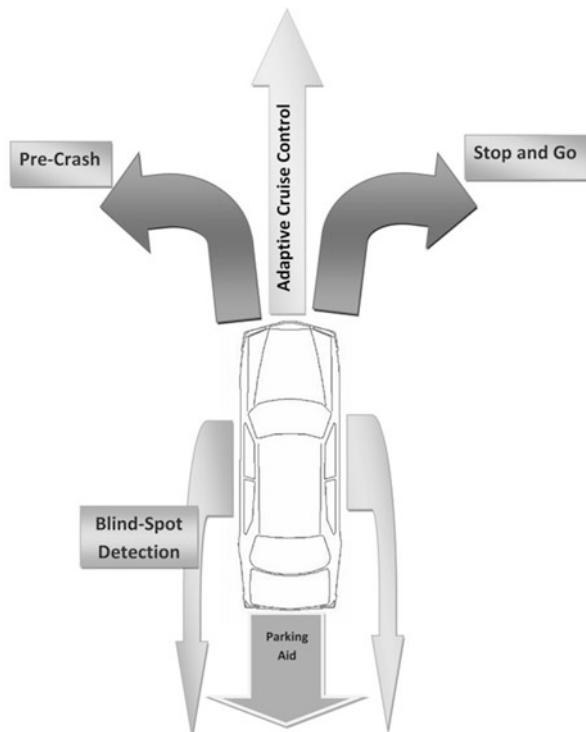
The main feature of short-range radars is range accuracy, while for midrange and long-range radar systems, the key performance feature is detection range. Short-range and midrange radar systems (a range of tens of meters) enable several ADAS applications, such as BSD, PCCMS, and LDW. They can also be used for implementation of SaG applications in city traffic. Long-range radars (hundreds of meters) are typically used for ACC systems. These systems can provide enough accuracy and resolution for even relatively high speeds of ~ 120 mph (URL30 2017).

The typical radar applications in ADAS are shown in Fig. 4.27.

4.9.2.2 LiDAR Sensor

LiDAR is applicable in a broad range of locating, profiling, and ranging applications. A LiDAR system consists of a laser capable of transmitting light (pulsed or

Fig. 4.27 Radar sensor applications in ADAS (URL30 2017)



continuous) over the required range of distance and a high-speed, low-noise receiver for reflected signal analysis. The transmitted light interacts with the target. A percentage of this light is reflected to the receiver according to the reflectivity of the target. Changes in the properties of the transmitted signal enable some properties of the target to be determined by providing clear-cut 3D snapshots of every object in the vehicle's vicinity. Along with its surveying feature, LiDAR has improved capabilities when it comes to the detection of objects, even in cases where there is a complete absence of light. LiDAR's features, which include ACC, BSD, and PCCMS and PPS, are not only better than the features of other sensors but are also far more consistent and reliable.

It is expected that LiDAR will become a central element of the autonomous vehicle's sensor suite, alongside existing technologies, ensuring robust sensing redundancy and increasing overall system reliability. The number of vendors in the automotive LiDAR sensor market is high, including companies such as Bosch, Continental, Denso, Hella, First Sensor, LeddarTech, Novariant, Phantom Intelligence, Quanergy, Teledyne Optech, Valeo, and Velodyne Lidar.

4.9.2.3 Laser Sensor

Laser light consists of light waves of the same wavelength which have a fixed phase relationship (coherence) resulting in an important feature of laser sensors: the almost parallel light beam. The resulting small divergence angle makes it possible to realize large ranges and topological mapping, both of which are required to build a sensor that provides high reliability for long distance measurements to ensure reliable vehicle guidance and measurements with regard to collision avoidance or the ability to narrow to a specific target. In this regard, blind spots can cause costly collisions. ADAS functions performed by laser sensor systems are LDW, LHW, NFCW, OCW, and PCCMS.

Weather conditions, such as fog or even dust, do not create a problem for laser sensors when they are embedded in a target discrimination mode. In this mode, the sensor performs a comparison and is able to distinguish the last target from all other reflections, which equates to dust or fog penetration and the ability of the sensor to see the road surface.

4.9.2.4 Camera Sensor

The front camera sensor is used in an ADAS machine vision system using images from a forward-facing camera to perform tasks such as lane-departure warning (LDW), obstacle and collision warning (OCW), traffic sign recognition (TSR), and distance measurement, among others. The output can either be a warning to the driver or direct control of certain vehicle functions, such as steering or braking.

The rear camera system uses images from a backward-facing camera to perform tasks such as automated parking (AP), object detection, and distance measurement, among others. The output can either be a warning to the driver or direct control of certain vehicle functions.

The surround-view-camera system is an ADAS technology that assists the driver in parking the vehicle safely by providing a top-down view of the 360° surroundings of the vehicle, as can be seen in Fig. 4.28 (URL31 2017).

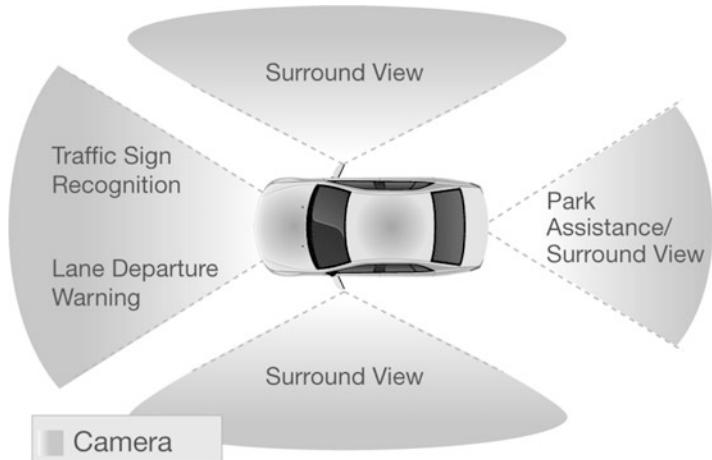


Fig. 4.28 Camera sensor applications in ADAS (URL31 2017)

4.9.2.5 Vision Sensor

Vision sensors use a camera to detect white lines and obstacles on the road ahead. A new sensor design by DENSO uses a pair of cameras placed side by side, which enable the distance to a target object to be measured more accurately and enhances the activation of autonomous emergency braking (AEB), lane-departure warning (LDW), and intelligent headlight system (IHS), a high beam system that automatically switches the headlights from high beam to low beam for better night vision.

Compact vehicles have limited space to install devices. Therefore, vision sensors are required to be small for easier installation. Generally, in stereo cameras, the longer the distance between the two camera lenses (baseline), the longer the measurable distance to the target, meaning that the camera body needs to be larger to extend the maximum measurable distance. A combination of highly accurate lens distortion correction and stereo matching technologies enable the new sensor to ensure that the maximum measurable distance is long enough while the baseline length is halved. Moreover, the new DENSO vision sensor, shown in Fig. 4.29, is integrated with and is controlled by an ECU (URL32 2017).

4.9.2.6 Ultrasonic Sensor

The ultrasonic sensor is used in ultrasonic-distance-ranging automotive applications, such as automated parking (AP) and blind spot detection (BSD), where ultrasonic waves transmitted by the sensor are reflected by objects that are close or in the near vicinity. The system receives the reflected wave, or echo, and compares the object's echo amplitude against a threshold to detect the object. The echo for objects that are closer to the system is stronger than that for objects that are farther from the system. Hence, it is relatively common for the threshold to be varied with time.

Fig. 4.29 DENSO vision sensor



Ultrasonic sensors are installed in the front and rear bumpers and wing mirrors of a vehicle to transmit ultrasonic waves and receive the ultrasonic waves reflected back by nearby objects. An ultrasonic wave's time of flight (TOF) is used to calculate the distance to the objects to assist the driver in parking the vehicle, identifying parking spots, or detecting objects in the driver's blind spot. Up to four sensors (transducers) are installed in the front and rear bumpers, and one sensor (transducer) is installed in each wing mirror.

Ultrasonic waves generated by the sensor are a series of sinusoid pulses at carrier frequency and are characterized by sound pressure level (*SPL*), which can be expressed by:

$$\text{SPL} = 20\log_{10} \cdot \left(\frac{P_{\text{RMS}}}{P_{\text{ref}}} \right)$$

where P_{RMS} is the *RMS*, and sound pressure P_{ref} is the reference sound pressure. The *SPL* of ultrasonic waves created by the transducer at an object depends on the object's distance from the sensor (transducer). Specifically, the pressure is inversely proportional to the distance:

$$p \sim \frac{1}{d}$$

where p is the pressure of the sound waves, and d is the distance of the object from the sensor (transducer). Tracking is a specific area of interest in the context of driver assistance systems based on ultrasound sensors such as lane change detection or blind spot surveillance systems. For given ultrasonic sensor specifications the *SPL* at arbitrary distance x from the sensor (transducer) can be calculated by using the distance law (URL32 2017).

4.9.3 Pros and Cons of the ADAS Sensor Suite

Today's ADASs (see Sect. 4.9.1 and Chap. 11) make use of the combination of different sensor types by combining their characteristics to achieve the most suitable performance for the respective application. Therefore, Table 4.8 shows several pros and cons of the different types of sensors used for ADAS (URL32 2017).

Table 4.8 Pros and cons of ADAS sensor types

Sensor type(s)	Pros	Cons	ADAS applications
Camera	Good lane detection	Traffic signs show very poor contrast	AP
	Usable under dark/night conditions		LDW OCW RVS PCCMS TSR
Laser	Good distance/speed detection	Poor lane detection	LDW LHW
	Small obstacles detectable	Poor vehicle or pedestrian detection	NFCW OCW PCCMS
LiDAR	Good obstacle detection	Poor vehicle/pedestrian detection	ACC BSD
	Good distance/speed detection	Smaller range than radar	PCCMS PPS
Radar	Accurate speed detection	Poor lane detection	ACC
	Good distance detection	Poor vehicle/pedestrian detection	BSD LDW
	Usable in environment with reflections	Beam blockage	PCCMS
	Usable for short- and long-range detection	Big size	SaG
	Waves transmitted are not affected by obstacles		
Ultrasonic	High angular range	Easily distorted by reflections on the road	ASP
	Short-range detection	No angular position	BSD
		No echo cancellation	
Vision	Good lane detection	Complex electronic system required for data processing	LCA LDW
	Good vehicle/pedestrian detection	Poor obstacle detection	LKA RDP
	Present images of reality Small sizes	Poor speed/distance detection	RSR RVS TSR

4.10 Trends

The automotive industry has been growing in the past decades and is an active sector with forecasts showing much more may come. Moreover, the globalization of the automotive industry has greatly accelerated during the last half of the 1990s due to the construction of important overseas facilities and mergers between giant multinational automakers. Global vehicle sales are expected to exceed 100 million units a year by 2020. There are some key trends which can be identified now:

- *Changes in Brand Loyalty:* Brand-loyal customers are rethinking their buying decisions as a result of surplus choices in the market. Impressing the customer remains harder than ever before.
- *Changes in Customer Demand:* Many customers are inclined to buy greener, fuel efficient, and sustainable vehicles. With the market launch of e-vehicles and alternative fuel, automakers became aware that the days are gone when design and style were the major decision-making factors. After Volkswagen's emission scam, customers became more cautious.
- *Changes in Mobility:* The only objective that counts is the efficient and inexpensive mobility when using a vehicle. Autonomous vehicles are not the only trend challenging the automotive industry. Views about mobility, what we can do with a vehicle, and about the status of owning a car are in transition; and the number of female buyers is increasing.
- *Resource Shortage:* According to recent growth figures, electric vehicle sales grew massively in 2016. This is roughly equivalent to the growth forecasted by Tesla Motors, where production is expected to increase from 50,000 in 2015 to 500,000 in 2020. Assuming Tesla can meet its forecasts and its current electric vehicle market share remains the same, and if each electric vehicle roughly displaces 15 barrels of oil a year, then the next oil crash will occur much later than in a pure combustion engine based scenario.
- *Technological Advances:* The global automotive industry has witnessed a lot of transformation in the last two decades with the digitization of vehicles. Linking mobile devices to the vehicle creates many options. For example, one can check how much fuel is left, the condition of the brakes, when maintenance is needed, and other features. A mobile device can also be used as a vehicle key or for applying personal settings in a rented vehicle. Thus, connecting vehicles is the next big platform for application developers. It is assumed that in 2020 approximately up to 15% of new vehicles sold could be fully autonomous, especially if one considers the activities in China in this domain by Bytons SUV business concept, in which the Byton electric car is a smart device on four wheels (URL1 2018).

The concept of connected vehicles (see Sect. 5.3), which focuses on connecting vehicles with the outside world and enhancing the onboard experience, combines telecommunication and informatics to provide various services, such as:

- Automatic parking/parking management
- Automatic toll transactions
- Live traffic updates
- Onboard entertainment
- Roadside assistance in case of accidents
- Smart routing and tracking

Therefore, the next step beyond connected vehicles will be self-driving vehicles, also called autonomous vehicles, which in the long term will revolutionize vehicle operation and the experience of driving. The most important developments on the path to self-driving cars are ADASs, which are making vehicles safer; and their gradual introduction is already improving road safety. Thus, ADAS features represent an essential evolutionary step in developing self-driving vehicles but the development and market launch of self-driving vehicles is an evolutionary step which spans a number of automotive generations. ADAS, by its very nature, perfects different essential aspects and features of automated control, one of the requirements for self-driving vehicles. It accomplishes this through independent subsystems with increasing levels of system integration, ultimately resulting in a vehicle that can drive itself.

The different kinds of ADAS are the driving force behind connected vehicles and self-driving vehicles and can be summarized by the following characteristics:

- Information and warning systems
- Function-specific automation systems
- Combined function automation systems
- Limited self-driving automation systems
- Full self-driving automation systems

Security is becoming a major concern as vehicles are beginning to communicate with each other and with the road infrastructure installations as well as with traffic signs and traffic lights. Thus, connected and self-driving vehicles will need protection from malicious intrusion, i.e., vehicle hacking, which is described in detail in Chap. 6. In this regard the design and manufacturing of vehicular components and systems as well as vehicles itself require to follow a new design and manufacturing paradigm, which can be stated as security by design, as it was introduced by the German Industry 4.0 Platform (URL2 2018).

4.11 Exercises

What is meant by the term *mechatronics*?

Describe the characteristics of mechatronic systems.

What is meant by the term *intelligent control*?

Describe the characteristics of intelligent control.

What is meant by the term *cyber-physical systems*?

Describe the characteristics of cyber-physical systems.

What is meant by the term *automotive electronics*?

Describe the characteristics of automotive electronics.

What is meant by the term *body electronics*?

Describe the characteristics of body electronics.

What is meant by the term *chassis electronics*?

Describe the characteristics of chassis electronics.

What is meant by the term *comfort electronics*?

Describe the characteristics of comfort electronics.

What is meant by the term *driver assistance electronics*?

Describe the characteristics of driver assistance electronics.

What is meant by the term *electronic control unit*?

Describe the characteristics of an electronic control unit.

What is meant by the term *entertainment/infotainment electronics*?

Describe the characteristics of entertainment/infotainment electronics.

What is meant by the term *passive safety electronics*?

Describe the characteristics of passive safety electronics.

What is meant by the term *bus system*?

Describe the characteristics of bus systems.

What is meant by the term *entertainment electronics*?

Describe the characteristics of entertainment electronics.

What is meant by the term *infotainment electronics*?

Describe the characteristics of infotainment electronics.

What is meant by the term *sensor technology*?

Describe the characteristics of sensor technologies.

What is meant by the term *signal-to-noise ratio*?

Describe the signal-to-noise ratio mathematically and explain it.

What is meant by the term *active sensor*?

Describe the characteristics of an active sensor.

What is meant by the term *microelectromechanical systems*?

Describe the characteristics of microelectromechanical systems.

What is meant by the term *sensor node*?

Describe the characteristics of sensor nodes.

What is meant by the term *sensor data fusion*?

Describe the characteristics of sensor data fusion.

What is meant by the term *sensor network*?

Describe the characteristics of sensor networks.

What is meant by the term *analog-to-digital conversion*?

Describe the analog-to-digital conversion process.

What is meant by the term *ADC resolution characteristic*?

Describe the characteristics of ADC resolution qualitatively.

What is meant by the term *bus system*?

Describe the characteristics of bus systems.

What is meant by the term *CAN bus system*?

Describe the characteristics of the CAN bus systems.

What is meant by the term *LIN bus system*?

Describe the characteristics of the LIN bus systems.

What is meant by the term *FlexRay bus system*?

Describe the characteristics of the FlexRay bus systems.

What is meant by the term *functional safety*?

Describe the characteristics of functional safety.

What is meant by the term *safe failure fraction*?

Describe the characteristics of the safe failure fraction.

What is meant by the term *failure mode and effects and criticality analysis (FMECA)*?

Describe the characteristics of the failure mode and effects and criticality analysis.

What is meant by the term *agile software development*?

Describe the characteristics of the agile software development.

What is meant by the term *automotive spice*?

Describe the characteristics of the automotive spice.

What is meant by the term *ASAM*?

Describe the characteristics of ASAM.

What is meant by the term *model-based development*?

Describe the characteristics of model-based development.

What is meant by the term *rapid prototyping*?

Describe the characteristics of rapid prototyping.

What is meant by the term *model-in-the-loop test*?

Describe the characteristics of the model-in-the-loop test.

What is meant by the term *hardware-in-the-loop test*?

Describe the characteristics of the hardware-in-the-loop test.

What is meant by the term *AUTOSAR*?

Describe the characteristics of the AUTOSAR.

What is meant by the term *adaptive AUTOSAR platform*?

Describe the characteristics of the adaptive AUTOSAR platform.

What is meant by the term *GENIVI*?

Describe the characteristics of GENIVI.

What is meant by the term *advanced driver assistance system*?

Describe the characteristics of advanced driver assistance systems.

What is meant by the term *advanced driver assistance system (ADAS)*?

Describe the characteristics of advanced driver assistance systems functionalities.

What is meant by the term *advanced driver assistance system sensors*?

Describe the characteristics of advanced driver assistance systems sensors.

What is meant by the term *original equipment manufacturer (OEM)*?

Describe the characteristics of OEM.

What is meant by the term *Tier 1 supplier*?

Describe the characteristics of Tier 1 supplier.

What is meant by the term *connected car*?

Describe the characteristics of connected cars.

What is meant by the term *autonomous vehicle*?

Describe the characteristics of autonomous vehicles.

What is meant by the term *connected car gateway*?

Describe the characteristics of the connected car gateway.

References and Further Reading

- (AA1Car 2016) Troubleshoot Automatic Climate Control Systems. 2016 http://www.aalcar.com/library/automatic_climate_control.htm
- (Barr 2004) Barr, D.: Supervisory Control and Data Acquisition (SCADA) Systems. NCS Technical Information Bulletin 04-1, 2004
- (Bechmann et al. 2015) Bechmann, R., Scherk, M., Heimann, R., Schäfer, R.: Trend Analysis: Connected Car 2015. MBtech Consulting GmbH, 2015. Available from: <https://www.yumpu.com/en/document/view/10955202/trend-analysis-connected-car-2015-mbtech-group>
- (Berlin and Gabriel 1997) Berlin, A. A., Gabriel, K. J.: Distributed MEMS: New Challenges for Computation. In: IEEE Computational Science and Engineering, Vol. 4, No. 1, pp. 12–16, 1997
- (Biddlecombe 2005) Biddlecombe, E.: BBC News 17.11.2005
- (Chaouchi 2010) Chaouchi, H.: The Internet of Things – Connecting Objects to the Web. J. Wiley Publ., 2010
- (DADSS 2016) Inventing a world without drunk driving; www.dadss.org
- (Gironimo et al. 2009) Gironimo, D., Lopez, A. M., Sappa, A. D., Graf, T.: Survey of Pedestrian Detection for Advanced Driver Assist Systems. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 32, Issue 7, pp. 1239–1258, 2009
- (Golatowski et al. 2003) Golatowski, F., Blumenthal, J., Handy, M., Haase, M., Burchardt, H., Timmermann, D.: Service-Oriented Software Architecture for Sensor Networks. In: Proc. of Internat. Workshop on Mobile Computing (IMC), pp. 93–98, 2003
- (Grosan and Abraham 2007) Grosan, C., Abraham, A.: Hybrid Evolutionary Algorithms: Methodologies, Architectures, and Reviews. In: Hybrid Evolutionary Algorithms, Studies in Computational Intelligence, Springer Publ. 2007
- (Heinecke et al. 2003) Heinecke, H., Schnelle, K.-P., Fennel, H., Bortolazzi, J., Lundh, L., Leflour, J., Mate, J.-L., Nishikawa, K., Scharnhorst, T.: AUTomotive Open System ARchitecture – An Industry-Wide Initiative to Manage the Complexity of Emerging Automotive E/E-Architectures. Convergence Transportation Electronics Association, 2004
- (Holtz and Möller 2017) Holtz, S., Möller, D. P. F.: Agile Software Development Use Case Example: Interface Renewal for Automated Loading of Tank Trucks. In: Proceed. IEEE/EIT Conference 2017, pp.115–120
- (Mercer Management Consulting 2001) Mercer Management Consulting and Hypovereinsbank. Study - Automotive Technology 2010 (in German), Munich, 2001
- (Möller 2003) Möller D. P. F.: Computer Structures – Fundamentals of Computer Science (in German), Springer Publ. 2003
- (Möller 2016) Möller, D. P. F.: Guide to Computing Fundamentals in Cyber-Physical Systems – Concepts, Design Methods, and Applications. Springer Publ., 2016
- (Moonen et al. 2005) Moonen, A., von den Berg, R., Bekooij, M., Bhullar, H., van Meerbergen, J.: A Multi-Core Architecture for In-Car Digital Entertainment. <http://www.es.ele.tue.nl/epicurus/publications/gspx05.pdf>
- (Moritz et al. 2011) Moritz, R., Ulrich, T., Thiele, L.: Evolutionary Exploration of E/E-Architectures in Automotive Design, pp. 361–365. In: Operations Research Proceedings, 2011, Eds.: Klatte, D., Lüth, H.-J., Schmedders, K., Springer Publ., 2011
- (Ning 2013) Ning, H.: Unit and Ubiquitous Internet of Things, CRC Press, 2013

- (Pala and Inanc 2007) Pala, Z., Inanc, N.: Smart Parking Applications Using RFID Technology. In: RFID Eurasia. 1st Annual Conference, 2007 DOI: <https://doi.org/10.1109/RFIDEURASIA.2007.4368108>
- (Poslad 2009) Poslad, S.: Ubiquitous Computing – Smart Devices, Environments and Interactions. John Wiley and Sons Publ., 2009
- (Rosengren 1995) Rosengren, L. G.: Driver assistance and co-operative driving. In: Proceedings of the First World Congress on Advanced Transport Telematics and Intelligent Vehicle Highway Systems, pp. 1613-1622, Artech House Publ., 1995
- (Schäuffele and Zurawka 2016) Schäuffele, J., Zurawka, T.: Automotive Software Engineering: Efficient Use of Basics, Processes, Methods and Tools (in German), Springer Publ. 2016
- (ST Microelectronics 2013) Complete car door module – AN 2334 Application Note. ST Microelectronics, 2013
- (Vembo 2016) Vembo, D.: Connected Cars – Architecture, Challenges and Way Forward. Whitepaper Sasken Communication Technologies Pvt. Ltd. 2016. Available from: https://www.sasken.com/sites/default/files/files/white_paper/Sasken-Whitepaper-Connected%20Cars%20Challenges.pdf
- (Vemuri 2012) Vemuri, A. T.: Using a fixed threshold in ultrasonic distance-ranging automotive applications. In: Analog Applications Journal, Vol 30, pp. 19-23. 2012
- (Yu and Gen 2010) Yu, X., Gen, M.: Introduction to Evolutionary Algorithms. Springer Publ. 2010
- (Zhang and Mi 2011) Zhang, X., Mi, C.: Vehicle Power Management – Modeling, Control and Optimization. Springer Publ. 2011
- (Zhao and Guibas 2004) Zhao F., Guibas, L.: Wireless Sensor Networks: An Information Processing Approach. Morgan Kaufmann Publ. 2004

Links

- (URL1 2013) <https://www.nxp.com/docs/en/white-paper/BODYDELECTRWP.pdf>
- (URL1 2017) https://en.wikipedia.org/wiki/Body_control_module
- (URL2 2017) https://en.wikipedia.org/wiki/Remote_keyless_system
- (URL3 2017) http://www.meecknet.co.uk/e38/E38_Sunroof_Description.pdf
- (URL4 2017) https://en.wikipedia.org/wiki/Anti-lock_braking_system
- (URL5 2017) https://en.wikipedia.org/wiki/Electronic_brakeforce_distribution
- (URL6 2017) https://en.wikipedia.org/wiki/Electronic_stability_control
- (URL7 2017) <https://en.wikipedia.org/wiki/Airbag>
- (URL8 2017) https://en.wikipedia.org/wiki/Hill_Descent_Control_system
- (URL9 2017) https://en.wikipedia.org/wiki/Advanced_driver_assistance_systems
- (URL10 2017) https://en.wikipedia.org/wiki/Power_steering
- (URL11 2017) <https://www.mems-exchange.org/MEMS/what-is.html>
- (URL12 2017) https://en.wikipedia.org/wiki/Sensor_node
- (URL13 2017) <https://community.nxp.com/docs/DOC-335306>
- (URL14 2017) https://en.wikipedia.org/wiki/Safety_integrity_level
- (URL15 2017) <https://www.powerspex.nl/en/wat-is/sil/>
- (URL16 2017) <https://webstore.iec.ch/publication/5517>
- (URL17 2017) https://en.wikipedia.org/wiki/Failure_mode_and_effects_analysis
- (URL18 2017) <https://wiki.asam.net/display/STANDARDS/ASAM+MCD-2+D>
- (URL19 2017) <http://www.ni.com/white-paper/10343/en/>
- (URL20 2017) www.dSPACE.org
- (URL21 2017) www.autosar.org
- (URL22 2017) <https://www.autosar.org/standards/adaptive-platform/>
- (URL23 2017) en.m.wikipedia.org/wiki/Advanced_driver_assistance_systems
- (URL24 2017) <https://in.mathworks.com/solutions/automotive/advanced-driver-assistance-systems.html>

- (URL25 2017) www.add2.co.uk/application/model-in-the-loop-testing-application
- (URL26 2017) <https://trimis.ec.europa.eu/project/advanced-driver-assistance-systems-europe-thematic-network-cluster-project>
- (URL27 2017) http://www.continentalautomotive.com/www/automotive_de_en/themes/passenger_cars/chassis_safety/adas/ldw_en.html
- (URL28 2017) http://www.continentalautomotive.com/www/automotive_de_en/themes/passenger_cars/chassis_safety/adas/rdp_en.html
- (URL29 2017) http://fleddartech.com/app/uploads/dlm_uploads/2016/02/Solution-Overview-Leddar-for-automotive-ADAS.pdf
- (URL30 2017) <http://www.sabertek.com/automotive-radar.html>
- (URL31 2017) <http://www.ti.com/lscs/ti/applications/automotive/adas/overview.page>
- (URL32 2017) https://www.utwente.nl/en/et/aida/education/Rapport_MP.pdf
- (URL33 2017) <https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/facing-digital-disruption-in-mobility-as-a-traditional-auto-player>
- (URL1 2018) <https://t3n.de/news/byton-chef-breitfeld-zukunft-autoindustrie-1062843/> (in German)
- (URL2 2018) <https://www.plattform-i40.de/I40/Navigation/DE/Industrie40/Handlungsfelder/Sicherheit/sicherheit.html>



The Connected Car

5

This chapter introduces the key technologies essential for the evolution of connected cars. Section 5.1 introduces cyber-physical systems (CPS) and describes engineered systems that integrate computing and networking technologies. The cyber part of CPS is deeply embedded in and interacts with physical processes, the physical components. Section 5.2 presents the concept of the Internet of Things (IoT) with regard to its communication capabilities anytime, from anywhere, with everything and key object radio-frequency identification (RFID) technology, which enables objects, things, or entities to be connected wirelessly. Section 5.3 focuses on telematics, infotainment, and the evolution of the connected car, taking into account technology maturity levels, driving factors, and business models of connected cars. Section 5.4 refers to platforms and architectures with regard to connected cars as well as the connected car reference platform and the connected car in the cloud. Section 5.5 introduces autonomous vehicles with regard to the respective guidelines for the testing and deployment of autonomous vehicles published by the National Highway Traffic Safety Administration (NHTSA). In Sect. 5.6, the GENIVI Alliance®, essential for the telematic and infotainment components, is introduced. Section 5.7 introduces several case studies of specific themes essential for the evolution of the connected car, such as the BMW ConnectedDrive Store, the Mercedes COMAND® Online, and HERE, which provides digital mapping for fully autonomous driving. Section 5.8 contains comprehensive questions for verifying the knowledge gained and finally followed by references and suggestions for further reading.

5.1 Cyber-Physical Systems

The integration of embedded systems, physical systems where the computer is completely encapsulated by the device it controls, and the interaction with physical processes via networked computing, led to the emergence of a new generation of

engineered systems, the cyber-physical systems. Cyber-physical systems (CPS) are complex, multidisciplinary, physically aware, next generation engineered systems (Möller 2016). CPS integrate computing and networking technologies, the cyber part, which are deeply embedded in and interacting with physical processes which add new capabilities to physical systems, the physical components. The term physical components in this book refers to automotive E/E components, such as electronic control units (ECUs) and others.

5.1.1 Introduction to Cyber-Physical Systems

In 2006, the National Science Foundation (NSF) in the USA identified CPS as one of the promising research themes of the future. In the following year, based on the recommendation of the US President's Council of Advisors on Science and Technology (PCAST) (PCAST 2007), a research program was established by NSF entitled Cyber-Physical Systems, through which about 65 projects have been funded. In a subsequent PCAST report in 2010, further research needs for CPS were identified; and the related NSF program was initially extended until 2013. A special interest organization has been set up in the USA, the Cyber-Physical Systems Virtual Organization (CPS-VO), to foster collaboration among CPS professionals in academia, government, and industry.

In 2012, a study funded by the German Federal Ministry of Education and Research (BMBF) was published by the German Academy for Science and Engineering (acatech) on the topic “Information and Communication Technology (ICT),” which was tied to the megatrend, Internet of Things (IoT), addressing the future opportunities and challenges of the technology trends of CPS (Geisberger and Broy 2012).

Furthermore, the European Union’s joint technology initiative, Advanced Research and Technology for Embedded Intelligence Systems (ARTEMIS), has invested in research and development (R&D) efforts on the next generation of engineered systems with a public-private partnership (PPP) between European nations and European industry. ARTEMIS supports the vision of a digital transformation where all systems, machines, and objects become smart and physically aware and have a presence in the cyber-physical space, exploiting the digital information and services around them and communicating with each other as well as with the environment. In addition, the European Commission covers CPS as an advanced computing research and innovation theme in their Horizon 2020 Programme (URL1 2017).

According to these studies and programs, CPS have an essential role in industry and society leading to breakthroughs in all relevant areas and bridging the wide range of fields of action in which CPS can be applied. CPS are composed of:

- *Cyber components:* Represent the next generation of embedded devices, processing information, and communicating wirelessly with their distributed environment.

- *Interfaces:* Deal with communication networks and other components, such as interconnected sensors, actuators, analog-to-digital converters (ADCs), and digital-to-analog converters (DACs). ADCs and DACs are responsible for converting continuous-time analog signals to discrete in-time digital signals and vice versa, respectively. Furthermore, interconnected devices bridging cyber components with physical components whereby sensors, sensor nodes, sensor networks, and actuators convert other forms of energy or information to electrical signals and vice versa, respectively.
- *Physical components:* In this book they represent vehicle electrical/electronic (E/E) components and architectures.

5.1.1.1 Wireless Sensor Networks

A wireless sensor network (WSN) consists of spatially distributed autonomous sensors to monitor physical and/or environmental conditions to cooperatively send measured data through the network to a main location. Today's modern WSNs are bidirectional, which means they also enable control of the sensor activity. Therefore, WSNs can also be introduced as ubiquitous communication networks to access relevant remote information and tasks, anywhere and anytime, following the IoT paradigm (see Sect. 5.2).

WSNs are used in many industrial and consumer applications, such as industrial process monitoring and control and machine health monitoring. Against the background of the manifold applications, different combinations of network functions and services are required, which results in a wide variety of WSNs based on the infrastructure, network range, frequency range used, bandwidth, and power consumption. Despite the disparity in the objectives of sensor applications, the main task of WSNs is to sense and collect data from a target domain, process the data, and transmit the information back to specific sites where the underlying application resides. Conducting this task efficiently requires the development of an energy-efficient routing protocol to set up paths between sensor nodes and the data sink.

WSNs are built of nodes, from a few to several hundreds or even thousands, where each node is connected to one or sometimes several sensors. Each such sensor network node typically has several parts, as shown in Fig. 4.6. The topology of the WSN can vary from a simple star-based network topology for monitoring and security applications to an advanced multihop, wireless mesh network, where the propagation technique between the hops of the network can be routing or flooding (Dargie and Poellabauer 2010; Sohraby et al. 2007).

With regard to WSN, described in Sohraby et al. (2007), which is taken as the basis for the design for routing protocols for WSNs, the following must be considered:

- Possibility of packet loss and delay
- Power and resource limitations of the network nodes
- Time-varying quality of the wireless channel

To address these design requirements, several routing strategies for WSNs are available:

- *Data-centric approach*: Disseminates interest within the network. This approach uses attribute-based naming, whereby a source node queries an attribute for the phenomenon rather than an individual sensor node. The interest dissemination is achieved by assigning tasks to sensor nodes and expressing queries relative to specific attributes. Different strategies can be used to communicate interests to the sensor nodes, including:
 - Anycasting
 - Attribute-based multicasting
 - Broadcasting
 - Geocasting
- *Flat network architecture*: Includes minimal overhead to maintain the infrastructure and the potential for discovery of multiple routes between communicating nodes for fault tolerance.
- *Location-to-address sensor node*: Location-based routing useful in applications where the node position within the geographical coverage of the network is relevant to the query issued by the source node. Such a query may specify a specific area where a phenomenon of interest may occur or the vicinity to a specific point in the network environment.
- *Network structure*: Imposes a structure on the network to achieve energy efficiency, stability, and scalability. Network nodes are organized into clusters in which a node with higher residual energy takes the role of a cluster head. The cluster head is responsible for coordinating activities within the cluster and forwarding information between clusters. Clustering has the potential to reduce energy consumption and extend network lifetime.

In Sohraby et al. (2007), several routing algorithms that have been proposed for data dissemination in WSNs are described. In addition, the design trade-offs and performance of these algorithms are also discussed.

In general, routing algorithms are based on various network analytical and graph-theoretical concepts as well on operations research (OR), including:

- Maximum flow
- Minimum span problems
- Shortest route

Routing is closely associated with dynamic programming and the optimal control problem in feedback control theory. The shortest path routing schemes find the shortest path from a given node to the destination node. If the cost, instead of the link length, is associated with each link, these algorithms can also compute minimum cost routes. Algorithms that are centralized find the shortest path from a given node

to all other nodes; if decentralized, they find the shortest path from all nodes to a given node. There are certain well-defined algorithms for shortest path routing, including:

- *Bellman-Ford algorithm*: Routing algorithm used for routing in mesh networks to determine the shortest route with the least number of hops through the network. The distance metric is based on the number of hops measured.
- *Dijkstra algorithm*: Finds the shortest paths between nodes in a graph, for example, road networks; has polynomial complexity.

Routing schemes based on competitive game theory notions have also been developed as described in (Lewis 2004).

Furthermore, large-scale communication networks contain cycles (circular paths) of nodes. Nodes as shared resources can handle multiple messages flowing along different paths. Therefore, communication nets are susceptible to deadlock, wherein all nodes in a specific cycle have full buffers and are waiting for each other. Then, no node can transmit because no node can get free buffer space, so all transmission in that cycle comes to a halt.

Livelock is the condition wherein a message is continually transmitted around the network and never reaches its destination. Livelock is a deficiency of some routing schemes that route messages to alternate links when desired links are congested without taking into account that the message should be routed closer to its final destination. Many routing schemes for routing with deadlock and livelock avoidance are available, as described in (Lewis 2004).

Flooding is a common technique frequently used for path discovery and information dissemination in wired and wireless ad hoc networks, as described in (Sohraby et al. 2007). The routing strategy is simple and does not rely on costly network topology maintenance and complex route discovery algorithms. Flooding uses a reactive approach whereby each node receiving a data or control packet sends the packet to all of its neighbors. After transmission, a packet follows all possible paths. Unless the network is disconnected, the packet will eventually reach its destination. Furthermore, as the network topology changes, the packet transmitted follows the new routes.

To prevent a packet from circulating indefinitely in the network, a hop count field is usually included in the packet. Initially, the hop count is set to approximately the diameter of the network. As the packet travels across the network, the hop count is decremented by one for each hop that it traverses. When the hop count reaches zero, the packet is simply discarded. A similar effect can be achieved using a time-to-live field, which records the number of time units that a packet is allowed to live within the network. At the expiration of this time, the packet is no longer forwarded. Flooding can be further enhanced by identifying data packets uniquely, forcing each network node to drop all of the packets that it has already forwarded. However, such a strategy requires maintaining at least a recent history of the traffic to keep track of which data packets have already been forwarded.

5.1.1.2 Shared Sensor and Actuator Networks and Control Systems

In shared sensor and actuator networks (SANs), resource scheduling is an important feature for CPS operation. In this regard, actuation coordination is essential to decide which actuators must be scheduled to perform a particular action or how to manage control actions properly. Various parameters must be considered during control task allocation to a particular actuator (Mo et al. 2014), such as:

- Actuator capabilities
- Energy consumption of each actuator
- Physical system requirements
- Real-time capability
- Task completion time

Regarding actuator scheduling, an important difference between CPS and most cyber systems is the reversibility or preemption of actuator operations. While rollback operations and preemption are available in most cyber systems, e.g., databases or bus access protocols, physical operations executed by the actuators typically cannot be reversed (Gunes et al. 2014). If an actuation is performed based on erroneous data, it is often challenging if not impossible to roll back the activity, as discussed in (Yan et al. 2012) in more detail. Additionally, challenge of non-reversibility affects real-time scheduling in cases where several jobs are managed on a shared platform. Even hard real-time tasks may be blocked by low-priority processes if a shared actuation resource access cannot be preempted or rolled back (Gunes et al. 2014; Springer et al. 2014).

With regard to control laws and the theory behind control systems, which are the basis for state-of-the-art continuous-time, dynamic control systems, these systems have a crucial role in CPS design (Gunes et al. 2014). Conventionally, control policies are completely separate from the system infrastructure and implemented after developing the system prototype (Erdem et al. 2010). Such an approach is not feasible for meeting the demands expected from CPS because of their complex and dynamic nature. To meet those demands and perform complex control laws, the physical system itself and its dependency relationship with those control laws must be well defined and modeled (Zhou and Baras 2013).

In cases where the feedback loop is closed over wireless sensor and actuator networks (WSANs), control design can be applied making the control system insensitive to network uncertainties, such as time-varying delays (Koutsoukos et al. 2008). Fidelity-aware utilization control, which integrates sensor data fusion (SDF) within the feedback loop, can be adopted in wireless cyber-physical surveillance systems (WCPSS) to optimize system fidelity and adaptively adjust the control objective of microcontroller utilization in the presence of environmental variations, such as noise characteristics and others characteristics (Chen et al. 2011). The importance of control theory in CPS design has been addressed by a number of studies (Gunes et al. 2014; Lee and Seshia 2011; Radhakisan and Gill 2013; Rajkumar et al. 2010).

In case where feedback control of a system is closed through a shared network, the control system is called a networked control system (NCS), in which the control input/output is passed through interconnected system components, such as (Gupta and Chow 2010):

- Actuators
- Controllers
- Sensors

Another type of control system is the so-called supervisory control and data acquisition (SCADA) system (Barr 2004), which represents control systems utilized to monitor and control processes. A SCADA system gathers data in real time from sensors in local and remote locations and transfers the data to the central computers in order to control equipment/conditions and take necessary actions (URL2 2017). CPS entail requirements far beyond the expectations of legacy control systems, such as SCADA (Gunes et al. 2014). With regard to their complexity, CPS go beyond traditional engineered systems employed in industry which requires a much closer networking of the appropriate systems and software engineering disciplines. Therefore, the design of CPS requires a significant amount of reasoning with regard to unique challenges and complex functionality, reliability, and performance requirements, such as real-time capability (see also Sect. 5.1.3.4).

The decreasing cost of computation, networking, and sensing provides the basic economic motivation for embedding networking and information and communication technology (ICT) into every industry and application domain. Moreover, the exponential growth in computing power has brought extremely sophisticated computers at reasonable prices to the market. The same trends have vastly improved sensing and actuation technologies. Thus, computers and communication have become the universal system integrator that keeps large systems together, thereby enabling the composition of the respective CPS components and infrastructures. Thus, CPS have an advanced technology and complex system architecture that connects computing and networking with the physical and cyber, or virtual, environment within one paradigm. Hence, it provides services such as:

- Control
- Information feedback
- Real-time monitoring

Other CPS characteristics are essential to merging the interaction of the physical and the cyber components by integration and collaboration of computation, communication, and control (the so-called 3Cs) (Ning 2013).

5.1.1.3 Technological Innovations

Against the background of technological innovations, it can be stated that CPS and the IoT (see Sect. 5.2) have many similarities. Both are actuating, computing, sensing, transmitting information, and using interaction technologies to merge

their cyber and physical components; but some differences can be recognized. The IoT emphasizes the connection of things with networks, while CPS emphasizes the integration of information on computational and physical elements (Li et al. 2011).

CPS incorporate the following with situational adequacy and ergonomic issues:

- Actuating through actuators, actuator nodes, and actuator networks
- Algorithms to adopt the behavior of networked systems
- Human-machine interfaces (HMI)
- Interoperability standards (see also Sect. 5.1.3.3)
- Ontologies to interlink the CPS applications (see also Sect. 5.1.3.1)
- Sensing through sensors, sensor nodes, and sensor networks

Comparing embedded systems with CPS shows the integration of computing and networking with physical and virtual processes with the objective to convey how to interact with the physical components to monitor and control the physical processes. Thus, CPS have many benefits, like:

- Allow individual machines to work together to form complex systems that provide new capabilities
- Make systems safer and more efficient
- Reduce the cost of building and operating these systems

The design of such systems requires an understanding of the dynamics of hardware, software, networks, and physical and cyber processes (Pellizzoni 2015), allowing new and advanced systems with complex dynamics and high reliability to be created. Thus, the consolidating technological advances are that embedded computing systems (ECS) created networked embedded computing systems (NECS) which progressed to CPS and which finally converge to the Internet of Everything (IoE), Data and Services, as shown in Fig. 5.1, based on the report of Geisberger and Broy (2012). Figure 5.1 illustrates the technical evolution from embedded computing systems, through networked embedded systems to CPS and finally the Internet of Everything, Data and Services (Geisberger and Broy 2015).

The last two decades have launched a change in technological innovations that is driving a digital transformation of the industry. This change is not a matter of choice; it is driven by fundamental, long-term technological and economic trends which will continue. Thus, the challenge for CPS is to translate these technological and economic drivers into activities which will transform the industry and the manufacturing of products (Möller 2016).

At the systems level, CPS are mapping physical objects and corresponding virtual objects that communicate via ubiquitous information networks, whereby algorithms and services, as well as dynamic integration of services and service providers, and particularly information exchange, will cross borders. At the cyber level, data collected in arbitrary and alterable information networks, 3D models and simulation models, documents, relations, work conditions, and more will become available anywhere and anytime through ubiquitous or cloud computing (Möller 2016).

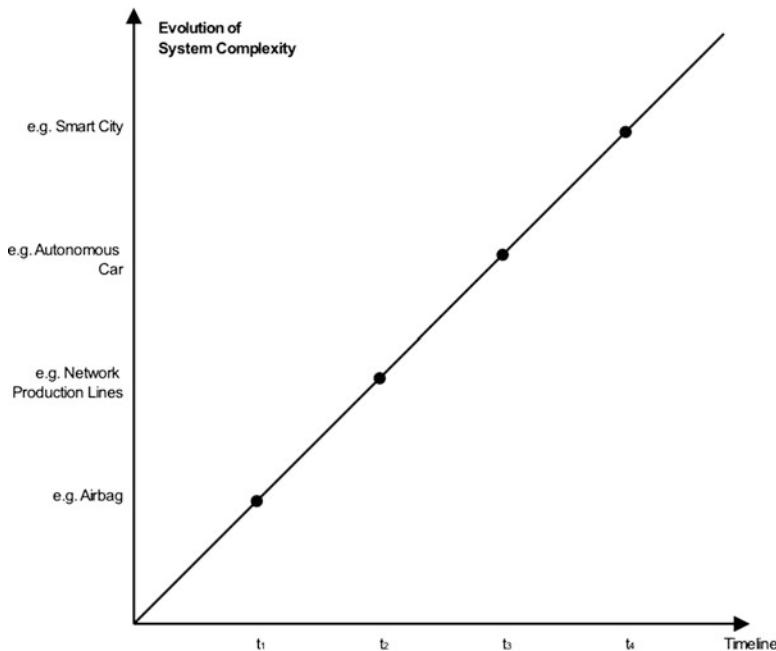


Fig. 5.1 Evolution of ECS into the IoE (Möller 2016)

Ubiquitous computing is everywhere at the same time, meaning it is omnipresent and makes computing an embedded, invisible part of everyday life. Tiny computing devices, which vanish into the environment are required to introduce ubiquitous computing, thereby creating a completely new paradigm of a computing environment for heterogeneous sets of devices, including invisible computers embedded in everyday things/objects such as automation devices, home devices, mobile devices, personal devices, security devices, vehicles, and wearable devices situated in various environments.

Cloud computing is a metaphor for the utility and consumption of computing resources. It involves deploying groups of remote servers and software networks that allow centralized data storage and online access to computer services or resources.

The main consolidation of a typical CPS can be seen in the integration of the dynamics of physical processes with those of the software and networking, providing abstractions and modeling, design, and analysis techniques for the integrated whole (URL3 2017), as shown in the CPS concept map in Fig. 5.2.

From Fig. 5.2, it can be seen that CPSs are primarily considered as an engineering discipline, focused on technology with a strong foundation in mathematical abstractions. The technical challenge is to conjoin abstractions that have evolved for modeling physical processes, such as differential equations, stochastic processes, and more, with abstractions that have evolved in computer science with regard to algorithms and programs, which provide a procedural epistemology (Abelson and Sussman 1996).

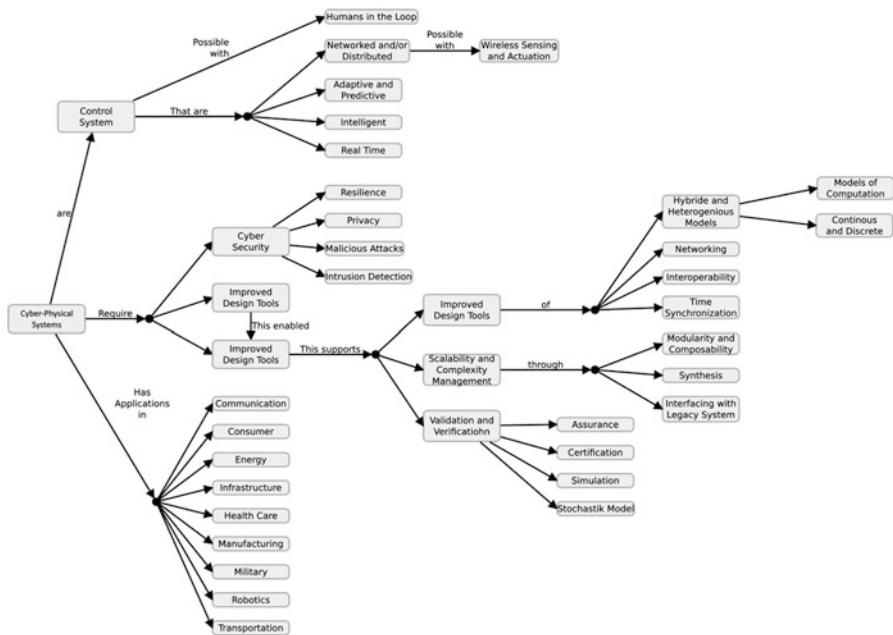


Fig. 5.2 CPS concept map (Möller 2016)

The former abstractions focus on the dynamic evolution of a system state over time, whereas the latter focus on the processes for transforming data. Computer science abstracts away core physical properties, particularly the passage of time that are required to include the dynamics of the physical world in the domain of discourse (URL3 2017). Thus, CPS have foundational characteristics such as:

- Capacity, with regard to information theory
- Formal methods, with regard to computer science
- Information processing, with regard to sensor networks
- Middleware, with regard to software engineering
- Real-time communication, with regard to networking
- Performance, with regard to control engineering

5.1.2 Cyber-Physical Systems Design Recommendations

Current industrial experience provides only limited knowledge of how to combine computers and physical systems. Therefore, continuing to design systems based on this limited knowledge (methods and tools) is not efficient; the risk of unsafe and unpredictable systems can be estimated. These shortcomings become extremely

important in CPS design because these systems are heterogeneous, comprised of multiple types of physical systems and multiple models of computation and communication. Therefore, heterogeneity in CPS design results in system-specific design flows which are inappropriate for design automation. Increasing design complexity and the lack of effective, specialized design automation tools can limit design productivity and increase time to market. This means that there is a need to realign the abstraction layers in the design flows and build a new infrastructure for agile design of CPS, as described in the following text and based on Möller (2016).

The challenges in the design of CPS result in the abstraction of levels which can be introduced as part of a stack-based process to abstract away the low-level architecture details and make the underlying system components more effective and transparent to the designer. Therefore, components at any level of abstraction should be made predictable and reliable. So far, this is technologically feasible if the number and capabilities of the available system components can be queried and software developed for code portability between the cyber and physical parts. If it is not technologically feasible, then the next level of abstraction must compensate with a robust principal component analysis. But abstractions do not directly encapsulate the essential characteristics which means it is hard to predict whether or not the cyber part will meet requirements of the physical (Pellizzoni 2015).

A successful system design follows these principles, assuming that it is technically feasible to build predictable and reliable components. It is much harder to make wireless links predictable and reliable due to the increasing needs of interactive network traffic. This may result in delays which raises the fundamental question of how to support delay guarantees over an unreliable medium, such as the wireless one. This is an important issue for automakers because a premium car has >150 sensors and more than 100 switches that are connected by wiring on the order of >1500 m, for making the wiring harness. This is very costly, very heavy, and very complicated and can result in multiple and complex electromechanical failures in the harness. Therefore, automotive engineering tries to solve the fundamental question of how to overcome delays over the wireless medium. One possible option is to compensate one level up, using robust coding and adaptive protocols.

Another obvious fundamental question is whether it is technically feasible to make software-engineered systems predictable and reliable. At the foundations of computer architecture and programming languages, software is essentially perfectly predictable and reliable, as it is limited to what is expressed in simple programming languages. With regard to imperative programming languages with no concurrency, such as C, designers can count on a computer to perform exactly as specified with essentially 100% reliability.

A problem arises when scaling up from simple programs to software-engineered systems, particularly to a CPS. The fact is that even the simplest C program is not predictable and reliable in the context of a CPS because the program does not express any aspect of the behavior that is essential to the CPS. It may execute perfectly, exactly matching its semantics, and still fail to deliver the behavior needed by the CPS. For example, it could miss timing deadlines. Since timing is not in the semantics of C, whether a program misses deadlines is, in fact, irrelevant in

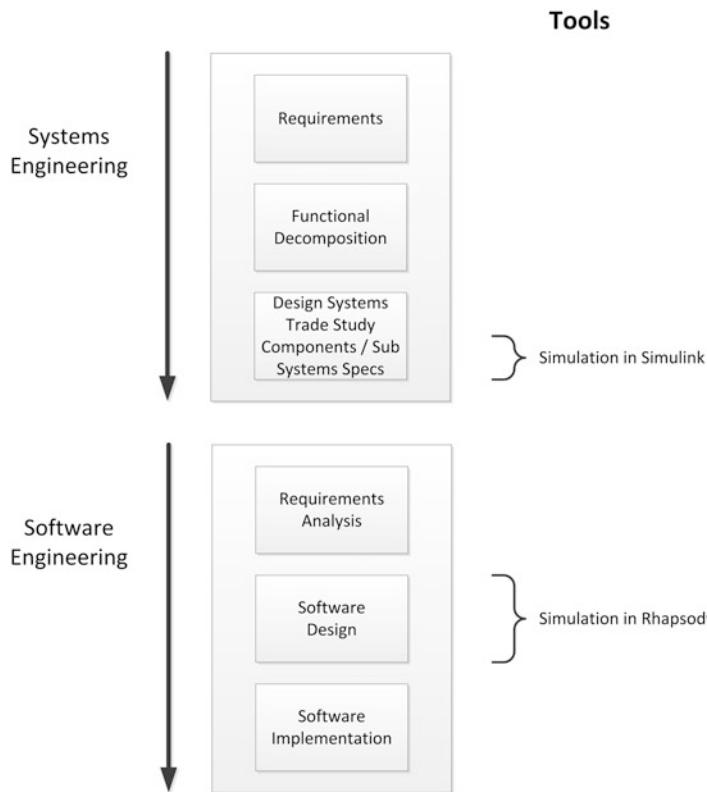


Fig. 5.3 Systems *and* software engineering approach in cyber-physical systems design (Möller 2016)

determining whether it has executed correctly. It is, however, very important to determine whether the system has performed correctly (Lee 2008). Thus, CPS design requires an adopted systems *and* software engineering approach with regard to their intrinsic complexity, as shown in Fig. 5.2.

As shown in Fig. 5.3, two essential engineering sectors in CPS design are integrated: systems *and* software engineering. The systems engineering approach can be characterized as follows:

- A systems engineering approach is a basic necessity for the planning, coordination and implementation of large complex projects as these projects are more difficult to manage and mistakes in early phases can lead to massive problems in the execution phase.
- Current systems engineering frameworks do not enable conceptualization and design which takes into account the deep interdependencies among engineered

systems and the natural world. Thus, there is a clear need for a new cyber-physical systems engineering framework (CPSEF) for the development, implementation, and operation of highly efficient CPS capable of handling the complexity of today's and tomorrow's many application domains. An interdisciplinary approach for developing and implementing complex technical systems in major projects in systems engineering is required.

- Mastering the engineering of complex and trustworthy CPS is crucial to planning, implementing, and sustaining business models.
- Ongoing integration of software-intensive, embedded computing systems and global communication networks (GCNs) in CPS is considered to be the next big step in the technological progress in ICT with a great deal of change in business potential and introduction of novel business models for integrated products and services.

The software engineering approach can be defined as:

- An approach concerned with all aspects of software production
- An approach to systematic, quantifiable design, development, operation, and maintenance of software
- Use of sound engineering principles in order to economically obtain software that is reliable and works efficiently on real machines

The Institute of Electrical and Electronic Engineers (IEEE) Computer Society and the Association for Computing Machinery (ACM) are the main US-based professional organizations of software engineering which have published guidelines to the profession of software engineering. The IEEE *Guide to the Software Engineering Body of Knowledge—2004 Version* defines the field and describes the knowledge IEEE expects of a practicing software engineer.

Today, CPS present a range of challenges which call for better and more effective architectural design environments with regard to the:

- Complexity of CPS, which results from parallel and distributed development, usually using different tools and methods
- Diverse sets of different algorithms with unique challenges
- Desire to reuse existing and future intellectual properties
- Electronics and software content that can be intricate
- Increased need to manage requirement changes during the development cycle
- Needs for integration and testing that can become costly and time consuming

With regard to specific constraints, Systems Modeling Language (SysML) and Unified Modeling Language™ (UML) (URL4 2017) have been designed as architectural frameworks and have been validated across numerous industries with their

separate views for functional, physical, and software architectures as well as their requirements capture and elicitation, and have been expanded by Rhapsody which integrates a rich set of external components, such as:

- *Code*: C, C++, Java, or Ada
- *Tools*: Simulink®, Statemate®, and SDL Suite

A very important intrinsic characteristic in CPS design is the interface. The CPS interface inherits all of the elements from the cyber and physical parts and adds new elements that bridge the gap between computational and physical systems. To model the interactions between the cyber and the physical worlds, two directed connector types are essential: the physical-to-cyber (P2C) and the cyber-to-physical (C2P) connectors. Thus, simple sensors can be modeled as P2C connectors; and simple actuators can be modeled as C2P connectors. One of the major difficulties in providing tool support for architectural design and analysis is the need to tailor those capabilities to the application domain.

5.1.3 Cyber-Physical Systems Requirements

Gathering and analyzing CPS requirements require a perspective that is sensitive to scope and interplay between the cyber, the physical, and the behavioral aspects of the system to emphasize disciplined approaches to design. This includes functional decomposition, abstraction, and formal analysis from a systems engineering perspective, as shown in Fig. 5.3. To keep the complexity of a design in check, it is necessary to employ mixtures of semiformal and formal approaches to CPS development. At the semiformal level of CPS design, the goals and possible scenarios are allocated at the system analysis level by SysML and UML. The task allocated at the formal analysis level is design space exploration and, at the system analysis level, the detailed simulation analysis. Simulation analysis is an essential task, with regard to the lack of an integration science with the needed mathematical foundation.

5.1.3.1 Requirements Characteristics

The semantics of the heterogeneous data sources in CPS are captured by their ontologies representing terms and relationships, an important method that builds sharable and reusable knowledge repositories and supports their interaction (Zhai et al. 2007). Hence, ontology can be defined as an abstract representation of real-world objects of the CPS under investigation, which means that the ontology constitutes a domain-specific model defining the essential domain concepts, their properties, and the relationships between them, represented as a knowledge base. Therefore, an ontology (O) organizes domain knowledge in terms of concepts (C), properties (P), and relations (R).

$$O = (C, P, R)$$

In other words, an ontology (O) is a triplet where C is a set of concepts essential for the domain, P is a set of concept properties essential for the domain, and R is a set of binary semantic relations defined between concepts in C . A set of basic relations is defined as $R_b = \{\approx, \uparrow, \nabla\}$ with the following interpretations (Zhai et al. 2007):

- For any two ontological concepts, $c_i, c_j \in C$, \approx denotes the equivalent relation, meaning $c_i \approx c_j$. If two concepts, c_i and c_j , are declared equivalent in ontology, then instances of concept c_i can also be inferred as instances of c_j and vice versa.
- \uparrow is the generalization notation. In cases where the ontology specifies $c_i \uparrow c_j$, then c_j inhibits all property descriptors associated with c_i ; and these need not be repeated for c_j while specifying the ontology.
- $c_i \nabla c_j$ means c_i has part c_j . If a concept in ontology is specified as an aggregation of other concepts, it can be expressed by using ∇ .

5.1.3.2 Requirements Engineering

Requirements engineering can be introduced as the process of defining, documenting, and maintaining requirements, which are documented physical and functional needs that a particular design of a system must be able to perform. The fields concerned with requirements engineering are systems and software engineering as described in Möller (2016). The activities involved in requirements engineering vary widely, depending on the type of system being developed and the specific style guide practices of the organization involved. These may include (URL5 2017):

- *Requirements elicitation*: Practice of collecting requirements of a system from users, customers, and other stakeholders; sometimes also called requirements gathering.
- *Requirements identification*: Concerned with analyzing, eliciting, elaborating, structuring, specifying, negotiating, documenting, and modifying requirements to propose future direction for successful and efficient work.
- *Requirements analysis and negotiation*: Tasks determining the needs or conditions to meet for a new product, with regard to possibly conflicting requirements of the various stakeholders.
- *Requirements specification*: Documenting the requirements in a requirements document.
- *System modeling*: Developing models of the system, often using a notation such as the:
 - UML®.
 - UML Profiles Requirements Validation: Check that the documented requirements and models are consistent and meet stakeholder needs.
- *Requirements management*: Managing changes to the requirements as the system is developed and put into use.

UML is a general purpose modeling language designed to provide a standard way of visualizing the system design. UML describes the order in which actions are carried out. All actions taken together describe a process. UML activity diagrams consist of nodes and edges. Certain events occur on the node. Edges connect nodes. Tokens are spread out over the entire activity diagram.

When project information is spread across multiple documents, it is difficult to assess the completeness, consistency, relationships between requirements, design, engineering analysis, verification, and validation information. It is also difficult to establish the end-to-end traceability needed to support change impact assessments.

In order to address the document-centric limitations, more advanced systems engineering processes are transitioning to a data-centric approach, which allows all systems engineering team members to access any project and/or related data. Thus, the data-centric approach is a major part of the Cradle® software tool, a requirements management and systems engineering tool which can easily be used for CPS design purposes. Cradle integrates the entire project life cycle into one, massively scalable, integrated, multiuser software product. It identifies the data to be captured, as shown by Baker (2015) in Fig. 5.4 and explained by Möller (2016). The term *mission* used in Fig. 5.4 is synonymous with the term *use case*.

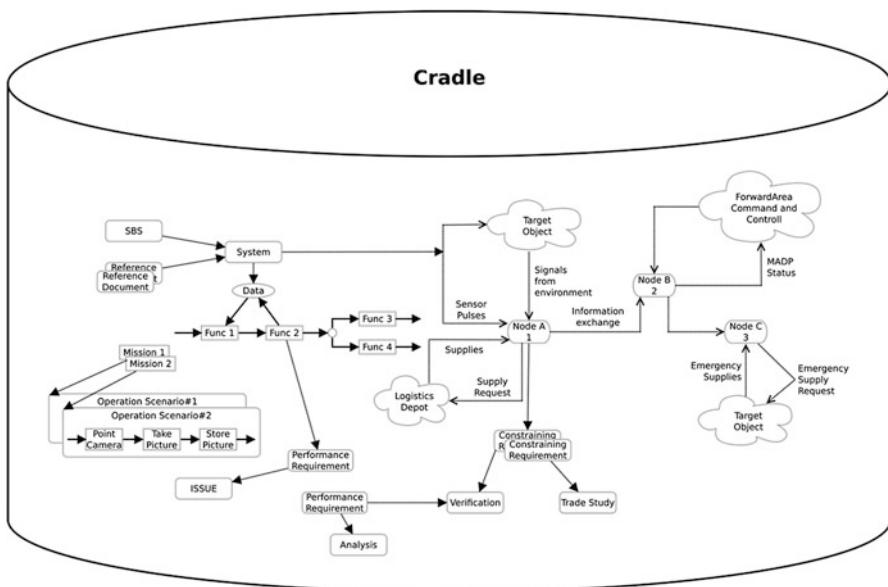


Fig. 5.4 Project data repository for the data-centric approach in the Cradle® software tool (Möller 2016)

Cradle software tool can be used to manage requirements definition and management activities for system development and modification. Cradle groups the requirements definition and management activities into eight stages, as shown in Fig. 5.5.

5.1.3.3 Interoperability Requirement

Interoperability describes the ability of systems working together, which means to interoperate. The term was initially defined as allowing a seamless information exchange for services in ICT and systems engineering. A more general definition refers to social, political, and organizational factors in regard to their impact on system performance. From a more technical perspective, interoperability is the task of building coherent services for systems when individual system components are technically different and managed by different software systems. Interoperability can be introduced as (URL6 2017):

- *Syntactic interoperability*: Necessary condition in the case of two or more systems capable of communicating and data exchange with regard to specified data formats and communication protocols. Extensible markup language (XML) or structural query language (SQL) standards are tools for syntactic interoperability.
- *Semantic interoperability*: Ability to automatically interpret the information exchanged meaningfully and accurately in order to produce useful results as defined by the end users of the systems which is important for vertical integration. To achieve semantic interoperability, two or more systems must refer to a common information exchange reference model.

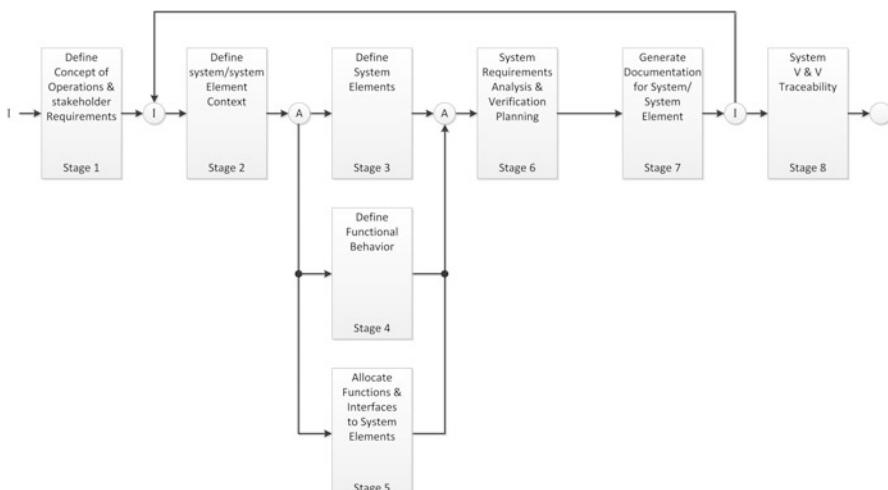


Fig. 5.5 Eight stages of requirements definition and management activities of the Cradle software tool (Möller 2016)

Interoperability must be distinguished from open standards. Open standards are being defined by a group of individuals that includes representatives from vendors, academicians, and others holding a stake in the development. The technical and economic merits, demerits, and feasibility of a proposed common protocol are discussed and debated. After the doubts and reservations of all members are addressed, the resulting common document is endorsed as a common standard. This document is subsequently released to the public; henceforth, it becomes an open standard. It is usually published and is available free of charge or at a nominal cost to any and all comers, with no further encumbrances. Various vendors and individuals can use the standards document to make products that implement the common protocol defined in the standard and are thus interoperable by design with no specific liability or advantage for any customer for choosing one product over another on the basis of standardized features (URL6 2017).

5.1.3.4 Real-Time Requirement

A real-time system is required to complete its tasks and deliver its services on a timely basis, which means that the time taken for the system to respond with an output from the associated input is within a sufficiently small acceptable timeline. Real-time systems include digital control, command and control, signal processing, and more. The Oxford Dictionary of Computing gives the following definition of a real-time system:

Any system in which the time at which output is produced is significant. This is usually because the input corresponds to some movement in the physical world, and the output has to relate to that same movement. The lag from input time to output time must be sufficiently small for acceptable timelines.

The Predictably Dependable Computing Systems project (Randell et al. 1995) gives the following definition:

A real-time system is a system that is required to react to small stimuli from the environment (including the passage of physical time) within time intervals dictated by the environment.

Fortunately, it is usually not a disaster if the system response is not in time. These types of systems can be discerned from those where failure to respond can be considered just as bad as a wrong response. It is this aspect that distinguishes a real-time system from others, where response time is important but not crucial. Consequently, “the correctness of a real-time system depends not only on the logical result of the computation but also on the time at which results are generated”. Practitioners in the field of real-time computer system design often differentiate between hard and soft real-time systems (Burns and Wellings 2001; Liu 2000).

- *Hard real-time systems (HRTS):* Systems where it is absolutely imperative that responses occur within the specified deadline.

- *Soft real-time systems (SRTS)*: Systems where response times are important, but the system will still work correctly if deadlines are occasionally missed. Soft real-time systems can themselves be distinguished from interactive ones in which there are no explicit deadlines (Burns and Wellings 2001).

The use of the term soft does not imply a single type of requirement but incorporates a number of different properties, such as:

- Deadline can be missed occasionally: typically with an upper limit of misses within a defined interval
- Service can occasionally be delivered late: typically with an upper limit of tardiness

As mentioned, real-time embedded control systems are used for process control, complex applications with regard to communication, command and control in the military domain, as well as in the control of aircraft, automobiles, autonomous robots, chemical plants, medical equipment, power distribution systems, and more.

The reliability requirement of real-time systems is to translate the need to meet critical task deadlines with a very high probability. Hence, the following question needs to be answered, “How can tasks be scheduled such that deadlines continue to be met despite processor permanent, transient, or software failures (Chen et al. 2011)?

5.1.4 Cyber-Physical Control Systems

Control refers to the directed influence of an engineering system, such as a CPS, whose properties correspond to the observed characteristic transfer elements. Moreover, in CPS-based control systems, the system’s output not only depends on the unilateral impact of the arrangement of the reference input (set value) but also depends on the disturbances occurring. The reference input acts as a control input for the output transfer characteristic according to physical laws and links and/or timing so that the desired systems behavior is determined. Although, the system’s output has no influence on the reference input (missing feedback), the system’s output may differ due to external disturbances from the desired target value. Hence, a CPS-based control system can be an open-loop type system consisting of a number of transfer-block-based characteristics connected in series. This control principle in its conceptual annotation is shown in Fig. 5.6.



Fig. 5.6 Block diagram structure of a control system (Möller 2016)

In reality, disturbances frequently occur at different times and with different levels of strength. They have the potential to significantly displace the control system's output from the reference input. Against this background, it is useful to capture the system's output by a separate transfer block.

In case of deviations of the system's output from the reference input, the influence of the disturbance on the system can be compensated for through the principle of feedback control. With a simple open-loop control system, one cannot act against foreseeable disturbances. Hence, a system characteristic is required which in the simplest case has transfer characteristics for observing the system's output and comparing it with the reference input to calculate the identified error between them, forcing the system's output to follow the reference input. This principle is the closed-loop control system in which the system's output, whose dynamic depends on the chosen system's specific model, is forced to follow a reference input while remaining relatively insensitive to the effects of disturbances.

In the case of a difference between both signals, the summing point of the feedback loop generates an error signal which is transferred to the controller input. The controller acts on the error with regard to a control strategy and manipulates the system to make it track the reference input. Moreover, this closed-loop feedback forces the system's output to follow the reference input with regard to present disturbance inputs. Thus, closed-loop control contains more transfer elements than open-loop control. The transfer elements of closed-loop control are (Möller 2016):

- *Plant or process:* System to be controlled.
- *System output:* Particular system aspect to be controlled.
- *Reference input:* Quantity desired for the system's output.
- *Actuator:* Device used to control input to the plant or process.
- *Controller:* Device used to generate input to the actuator or plant to force the system's output to follow the reference input. Therefore, the controller contains the control strategy to make the desired output track the desired reference input.
- *Disturbance:* Additional undesirable input to the plant imposed by the environment that may cause the system's output to differ from the expected output with regard to the reference input.

To this point, these transfer elements are the same in an open-loop control system. The closed-loop control system has the following additional transfer elements:

- *Sensor:* Device to measure system output
- *Error detector:* Determines the difference between the measured system output and reference input

Therefore, a closed-loop controller continuously detects and compares the potential difference between the reference input and the system's output by making use of the sensor and error detector. The resulting error value of the error detector is read by the controller's input which then computes a setting for the actuator to manipulate the plant of the closed-loop, cyber-physical control system. The controller uses the

feedback from the error detector to force the system's output back to the reference value based on the chosen control law. The actuator modifies the input to the plant with regard to the requirements based on the error detector output and the controller transfer function.

A block diagram of a closed-loop control system is shown in Fig. 5.7, with its conceptual annotation referring to the transfer elements described above. The parameters and symbols are summarized in Table 5.1 (Möller 2016).

The block diagram of Fig. 5.7 depicts the structure of the control systems as an interconnection of blocks and symbols representing certain basic mathematical operations in such a way that the diagram corresponds to the system's mathematical model. The interconnecting lines between blocks represent the variables describing the system's behavior, such as input and state variable. For a fixed linear system with no initial energy, the output $y(t)$ is given by

$$y(t) = G(t) \cdot u(t)$$

where $G(t)$ is the block transfer function and $u(t)$ is the input. Hence, a block diagram is merely a pictogram representation of a set of algebraic equations, allowing blocks

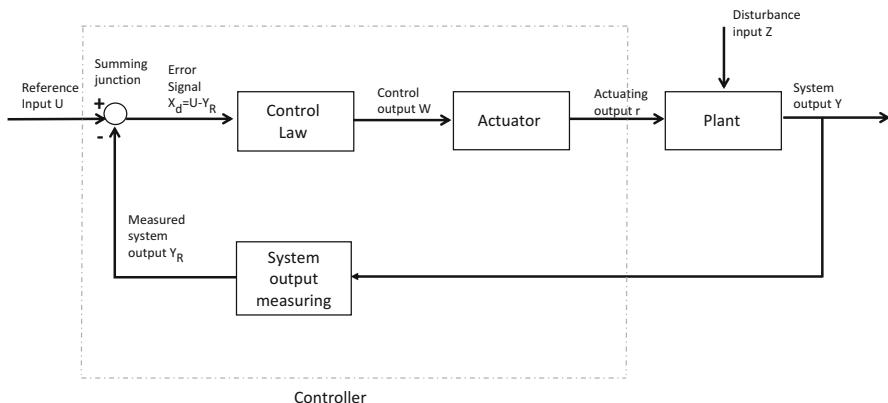
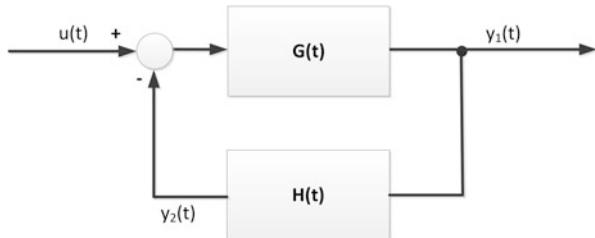


Fig. 5.7 Closed action of the control loop in block diagram form (Möller 2016)

Table 5.1 Denomination of closed-loop control system symbols

Symbol	Denomination
$u(t)$	Reference input or set value
$x_d(t)$	Error detection or control deviation
$y(t)$	Control output or correcting input
$r(t)$	Actuating output
$z(t)$	Disturbance input
$x(t)$	System output or control variable
$x_R(t)$	Measured system output or measured control variable

Fig. 5.8 Feedback loop
(Möller 2016)



to be combined by calculating the equivalent transfer function and, thereby, simplifying the diagram.

The block diagram of a feedback system that has a forward path from the summing point to the output and a feedback path from the system's output back to the summing junction (closed-loop) is shown in Fig. 5.8.

The block diagram shows the simplest form of a feedback control system. The transforms of the control system's input and output are $u(t)$ and $y_1(t)$, respectively. The transfer function is introduced as the forward loop gain or forward transfer function and as the feedback loop gain or feedback transfer function.

Let the model of a feedback system be given in terms of its forward and feedback transfer functions, $G(t)$ and $H(t)$. It is often necessary to determine the closed-loop gain or closed-loop transfer function

$$F(t) = \frac{y_1(t)}{u(t)}$$

$$H(t) = \frac{y_2(t)}{y_1(t)}.$$

This function can be derived from the block algebra equations for the closed-loop system by solving them for the ratio

$$\frac{y_1(t)}{u(t)}.$$

The block diagram structure corresponds to the following set of equations:

$$\begin{aligned} v(t) &= u(t) - y_2(t) \\ y_1(t) &= G(t) \cdot v(t) \\ y_2(t) &= H(t) \cdot y_1(t). \end{aligned}$$

Combining these equations to eliminate $v(t)$ and $y_2(t)$ yields

$$y_1(t) = G(t) \cdot [u(t) - H(t) \cdot y_1(t)]$$

which can be rearranged to give

$$[1 + G(t) \cdot H(t)]y_1(t) = G(t) \cdot u(t).$$

Hence the closed-loop gain or closed-loop transfer function

$$F(t) = \frac{y_1(t)}{u(t)}$$

is

$$F(t) = \frac{G(t)}{1 + G(t) \cdot H(t)}.$$

It is clear that the sign of the feedback signal at the summing point is negative. If the sign at the summing point is positive, then the closed-loop gain or closed-loop transfer function will become negative. A particularly simple case occurs when one assumes the feedback transfer function is unity, i.e. $H(t) = 1$. This control system is called a unity feedback system, yielding

$$F(t) = \frac{G(t)}{1 - G(t)}.$$

In practice, specific feedback transfer functions are used when designing cyber-physical control systems. Their closed-loop transfer function characteristics can be described by Möller (2016):

- Transient behavior or static characteristic curves
- Mathematical methods

The mathematical notation of the respective feedback law for the dynamic behavior of cyber-physical closed-loop control system transfer functions depends on the chosen characteristic of the specific controller block. In practice, the following elements are of importance:

- Proportional control
- Integral control
- Derivative control

5.1.4.1 Proportional Control

The proportional control (P feedback) is the most straightforward feedback, where the output of the controller varies directly as the input (or system error) $x_d = u - x_R$ which results in (Möller 2016)

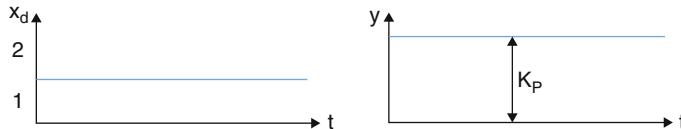


Fig. 5.9 Unit step of an ideal proportional controller; left side step input, right side step input multiplied by gain factor K_p

$$y(t) = K_p \cdot x_d(t)$$

where K_p is the gain factor of the proportional control. Increasing K_p will increase the closed-loop gain of the control system and can, therefore, be used to increase the speed of the control system response and to reduce the magnitude of any error. The cyber-physical control system with proportional feedback is referred to as a system zero order or a system without a memory element. The graph in Fig. 5.9 shows the response of the proportional control using the step response as input with a fixed gain of K_p .

The proportional control alone, however, is often not good enough because increasing K_p not only makes the system more sensitive but also tends to destabilize it. Consequently, the amount by which K_p can be increased is limited; and this limit may not be high enough to achieve the desired response. In practice, when trying to adjust K_p , conflicting requirements may occur. On one hand, it is intended to reduce any steady-state error as much as possible; but to attempt this by increasing K_p is likely to cause the response to oscillate, resulting in a prolongation of the settling time. On the other hand, the response to any change of the input signal should be as fast as possible but with little overshoot or oscillation. Fast control system response can be achieved by increasing K_p , but the increase is likely to destabilize the control system.

To solve the conflicting requirements with regard to the control system gain, a P controller is required that has a:

- K_p value that is high in order to reduce the control system error
- K_p value that is high to ensure a rapid response
- K_p value that is low enough to ensure that the dynamic response does not overshoot excessively and that any tendency to oscillate is damped fast enough

To fulfill these requirements, the P controller has to be expanded by adding, to the proportional part, one or two other control terms, such as integral control, differential or integral and differential control.

5.1.4.2 Integral Control

The prime purpose of adding an integral control part to a controller is to remove any steady-state error. The integral controller is usually used together with proportional and derivative control and in cases where the speed of response and instability are not a problem.

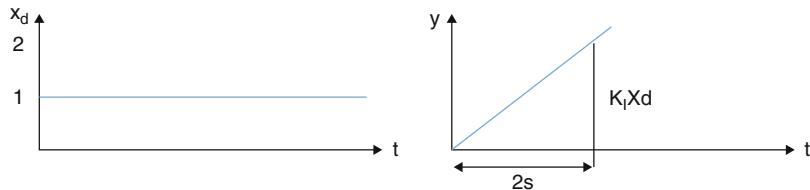


Fig. 5.10 Unit step of an ideal integral controller

An integral control dependence exists for which the output signal x and time integral of input x_d are proportional. Time integration of the control deviation e with the system's output or the actuated variable y acts with a reset time T_N . The reset time is called the integration factor or integration time constant. This means that for a reset time of $T_N = 2\text{s}$ at time $t = 0$, the output value y after 2s has reached the value of the constant input x_d . In the case of an integral controller, the actuator variable r , apart from the initial value, is proportional to the time integral of the control deviation (Möller 2016)

$$y(t) = 1/T_N \cdot \int_{t_0}^{t_1} x_d(t) dt.$$

If the input to the integral control element is zero, the output value does not change. By choosing a constant input value unequal to zero, the integral controller output changes with a constant increase. The integral controller has no steady-state error like the proportional controller. The integral controller is relatively slow in comparison to the proportional controller. By choosing a reset time T_N (proportional factor $K_I = 1/T_N$) that is too large, there is, however, an overshoot of the control variable; the controller becomes extremely unstable. Technically, the digital version of an integral controller is implemented by summation over a time interval. The graph in Fig. 5.10 shows the response of the integral controller for a unit step response at time $T_N = 1$ with $K_I = 1/T_N$ and $x_d = 1$.

5.1.4.3 Derivative Control

Derivative control is used in the controller to speed up the transient response of cyber-physical control systems. Derivative action is always accompanied by proportional control. Integral control is used only if necessary. Embedding derivative action in the controller has a stabilizing effect on the cyber-physical control system by virtue of the addition of phase lead to the closed-loop control system by reducing the phase lag of the gain factor of the derivative control.

For a derivative controller, the output u is proportional to the time derivative of the input signal x_d . Therefore, the actuating variable y is proportional to the rate of change of the control deviation x_d which yields (Möller 2016)

$$y(t) = T_V \frac{dx_d}{dt}.$$

In the case of sudden changes in the system's output (control variable), the actuated variable y increases immediately and, thereafter, goes back to its original value. Ideally, a derivative controller follows the Dirac pulse as a step response whose graph is an infinitely high, infinitely thin spike at the origin, with the total area under the spike, and physically represents the density of an idealized point mass or point charge.

A pure derivative controller cannot be realized in practice because the differentiation eliminates the set point. Therefore, the derivative controller is used in combination with the proportional controller, or integral controller, to achieve a quick response to sudden changes in a system's output (control variable) x .

Technically, the digital version of a derivative controller is implemented by differentiation over a time interval. The constant T_V is called derivative action time. The graph in Fig. 5.11 shows the unit step response of a derivative controller, for a gain factor of $K_D = T_V = 1$.

5.1.4.4 Proportional, Integral, and Derivative Control

These controls, as mentioned earlier, are widely used for controlling the response of cyber-physical control systems (CPCS). The derivative action is used to increase the speed of response, while the integral part prevents steady-state errors from occurring in the flow rate or actuator position.

The integral behavior of the proportional-integral-derivative (PID) controller is usually used when the controller is trying to maintain the system's output at its nominal working range and where changes in the system's output only occur as a result of changes in the load.

In a case where the input to a PID controller is changed significantly, the integral part of the controller is usually turned off or suppressed until the system's output is close to its nominal working range. If the integral part is not suppressed, then the large change in the input to the PID controller causes large oscillations to be superimposed onto the response of the cyber-physical control system. Hence, the oscillating response interacts with the two other control elements, the proportional and the derivative; the result is a very cyclic response of the cyber-physical control system with a very long settling time.

A general constraint for using integral control is that it should only be used if steady-state errors exist that cannot be tolerated in the cyber-physical control system

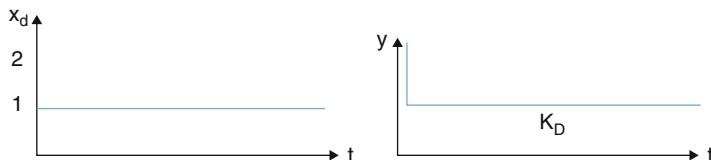


Fig. 5.11 Unit step of an ideal derivative controller; left side step input, right side step input response at time T_N with $K_I = 1$

strategy. Even the contribution of the integral behavior used should be just enough to remove the steady-state error without causing the steady response to oscillate. Where steady-state errors either do not exist or can be tolerated, then a proportional-integral-derivative controller will be sufficient.

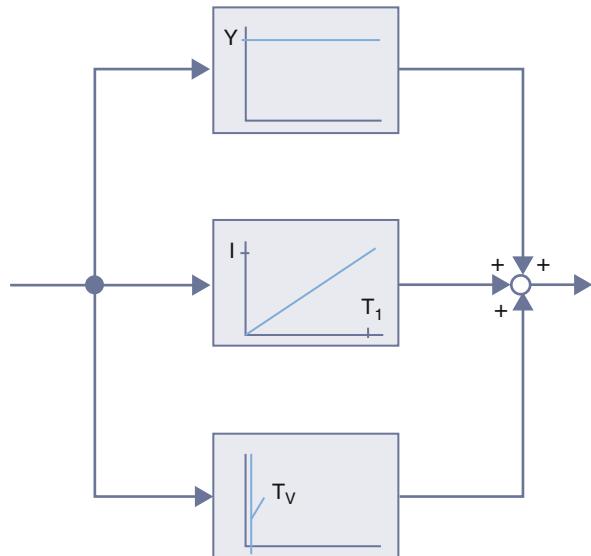
The PID controller combines all three control laws, the proportional, the integral, and the derivative. The input to the PID controller is the error signal x_d which is connected with the three parallel input ports of the controller, as shown in Fig. 5.12. The output signals of the proportional, the integral, and the derivative controller elements are merged into a summing point. The output of the summing point is the weighted sum of the proportional, the integral, and the derivative controller outputs. The three outputs have the same positive sign, and the weighting factors of the summing inputs of the summing point are assumed to have a value of 1. In Fig. 5.12, the constant T_I represents the reset time of the integral element; and T_V represents the derivative action time of the differential element (Möller 2016).

From Fig. 5.12, the following equation can be derived:

$$y(t) = K_P \cdot x_d + K_I \int_{t_0}^{t_1} x_d(\tau) d\tau + K_D \cdot \frac{dx_d}{dt} + x_d(0)$$

where $x_d(0)$ is the initial value, K_P is the gain factor of the proportional term, and $T_I = 1/T_N$ is the integral controller gain factor with T_N as the reset time; and T_V is the derivative controller gain factor. After excluding K_P and with regard to the boundary condition $x_d(0) = 0$, it follows

Fig. 5.12 Block diagram of the PID controller. For details see text



$$y(t) = K_P \left(x_d + \frac{T_1}{K_P} \cdot \int_{t_0}^{t_1} x_d(\tau) d\tau + \frac{K_D}{K_P} \cdot \frac{dx_d}{dt} \right)$$

with

$$\frac{K_P}{T_1} = T_N$$

and

$$\frac{K_D}{K_P} = T_V$$

we receive

$$y(t) = K_P \left(x_d + \frac{1}{T_N} \cdot \int_{t_0}^{t_1} x_d(\tau) d\tau + T_V \cdot \frac{dx_d}{dt} \right).$$

Using the Laplace transform, the above equation can be written as follows with the complex numbers denoting the frequency domain:

$$G(s) = K_P \left(1 + \frac{1}{s \cdot T_N} + s \cdot T_V \right).$$

For a number of calculations, it may be more appropriate to rewrite the above additive form into the following multiplicative form:

$$G(s) = K_P \cdot \frac{(1 + s \cdot T_1) \cdot (1 + s \cdot T_2)}{s \cdot T_N}.$$

Comparison of coefficients yields

$$T_1 = \frac{T_N}{2} \left(1 + \sqrt{1 - \frac{4T_V}{T_N}} \right)$$

$$T_2 = \frac{T_N}{2} \left(1 - \sqrt{1 - \frac{4T_V}{T_N}} \right)$$

where $T_N > 4 \cdot T_V$. From $T_N > 5 \cdot T_V$, the following relations can be found:

$$T_1 = T_N$$

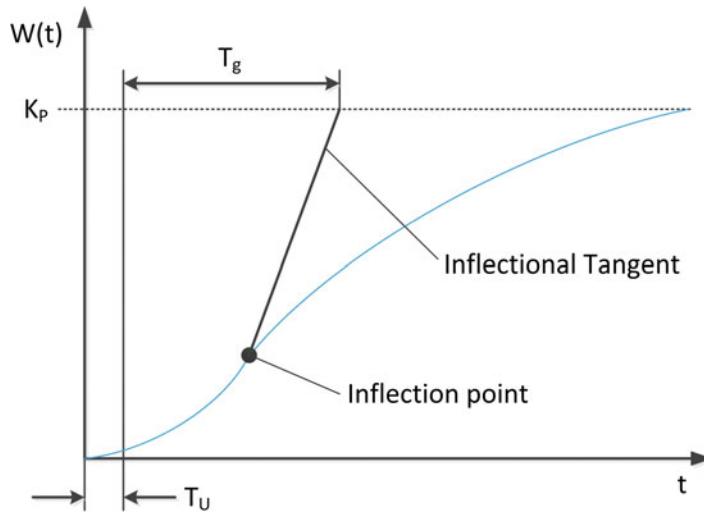


Fig. 5.13 Transient behavior of a step response (for details see text)

$$T_2 = T_V.$$

It can be seen from the above equations that the PID controller has two zero elements and a pole at the origin of the s -plane. The gain factors, K_P , T_N , and T_V , of the PID controller can be calculated using the tangent at the inflection point of the step response with the abscissa as the lower auxiliary variable T_U and the intersection of the tangent with the 5τ value of the step response as the top auxiliary variable T_g , as shown in Fig. 5.13 (Möller 2016).

From Fig. 5.13, the corresponding values for T_U and T_g , pictured on the abscissa time t , can be read. Let the PID controller overshoot the quotient of the auxiliary variable O_{max} , for the maximum overshoot height at T_{95} describes the default values for the PID controller

$$K_P = \frac{T_{95}}{O_{max}} \cdot \frac{T_g}{T_U}.$$

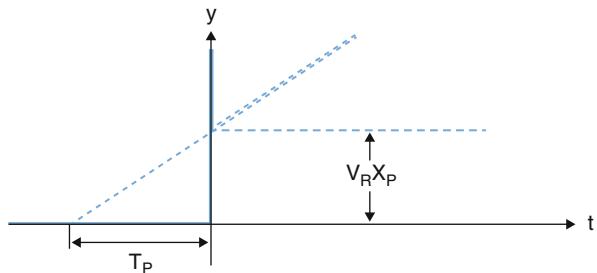
Assuming that the PID controller is not allowed to overshoot results in the following equation with regard to the above introduced auxiliary variables:

$$K_P = \frac{T_{60}}{O_{60}} \cdot \frac{T_g}{T_U}$$

$$T_N = T_g$$

$$2 \cdot T_V = T_U.$$

Fig. 5.14 Transfer function of the ideal PID controller (Möller 2016)



The ideal PID controller was introduced as a parallel connection of an ideal PID controller, which is represented by the addition of individual transfer functions as follows:

$$g(t) = K_P + \frac{K_P}{T_N} \cdot t + K_P \cdot T_v \cdot \delta(t).$$

The transfer function $g(t)$ of the ideal PID controller given above can be illustrated as shown in Fig. 5.14.

When designing a controller, simulation programs are very often used to optimize the controller design. This can be done based on the industry standard software package, MATLAB® Simulink (Chaturvedi 2010).

5.1.5 Cyber-Physical Vehicle Tracking

Analyzing and designing CPS for vehicle tracking require a priori knowledge on whether the system being designed can be assumed to be controllable, observable, and/or identifiable. Controllability, observability, and identifiability are important properties of CPS. With regard to the analysis of linear systems, it can be said that a linear system is state controllable when the system input u can be used to transfer the system from any initial state to any arbitrary state in finite time. Moreover, a linear system can be said to be observable if the initial state $x(t_0)$ can be determined uniquely for a given output $y(t)$ for $t_0 \leq t \leq t_1$ for any $t_1 > t_0$. If a mathematical model of a system can be written in the state notation, the method of controllability, observability, and identifiability analyses can be used for model predictions.

Vehicles are an important part of the overall transportation system, which includes not only a large number of human-made infrastructures, such as large bridges across lakes or rivers, long and big tunnels, urban elevated bridges, etc., but also a huge variety of vehicles, people, and goods in the complex transportation environment. Especially in urban traffic control systems, a large number of digital devices and information systems are available, as well as complex management and control systems. This allows developing road infrastructure CPS, vehicle road

Table 5.2 Function and constraints of cyber-physical transportation systems

	Physical traffic process	Information technology process	Functions
Road infrastructure CPS	Mechanics of changing the process of key transportation infrastructure, such as bridges, culverts, tunnels, subgrades, slopes, roadsides, etc.	Ubiquitous sensing in a wide range of reliable interconnected depth perception, forecasting, warning, and monitoring	Real-time monitoring of road facilities and transportation meteorological environment detection
Vehicle road coordinated CPS	Synergic relationship process of car-to-car and car-to-road which are running in the road and communication process	Wireless, high-speed, high reliability, secure communications; automatic driving	High-speed information exchange to guarantee safety of vehicles via efficient access
Traffic control CPS	Road traffic system process and traffic control process	Traffic control system model description, traffic system control, and traffic behavior control instruction optimization calculation	More secure and efficient dynamic road traffic control
Vehicle tracking CPS	Relationship between car-to-truck and truck-to-road running in the road, communication, and traffic control process	Wireless, high-speed, high reliability, secure communications of depth perception	High-speed information exchange to guarantee real-time monitoring of vehicles via efficient access

coordinated CPS, traffic control CPS, and cyber-physical vehicle tracking systems, respectively. The functions of these applications are shown in Table 5.2 (Jianjun et al. 2013).

5.1.5.1 Vehicle Tracking

To achieve efficient and safe road transportation is one of the motivations for conducting research on cyber-physical transportation systems, as it possesses information, and other features with regard to the essential needs of a cyber-physical vehicle tracking system, which is an important issue due to the growth in vehicle volume in recent years in both the public and private sectors. Public and private transportation is faced with the problem of transporting increasing volumes of passengers and freight. Within this process, freight must be identified several times. Currently, in most applications, bar code systems are used to identify the freight and the respective vehicle to which the freight belongs. These bar code systems, however, have some weaknesses, for example, they can repeatedly fail, resulting in the freight ending up in the wrong truck thereby wasting resources. Thus, the convergence of ubiquitous computing with embedded CPS, such as onboard units in trucks, is an important milestone enabling large-scale distributed cyber-physical

computing systems which are strongly coupled with their physical environment. Hence, RFID chips, as a component of wireless communication, are of great interest in transportation and logistics in the global economy, i.e., in process optimization in freight transportation in the transportation and logistics domain.

With the emergence of the recently released Internet Protocol v6 and low-power wireless personal area network (6LoWPAN) (Mulligan 2007), the convergence between CPS and the IoT becomes a reality because it enables using the Internet as supportive infrastructure to sensor networks, similar to its integration with RFID systems. This also allows tracking and on-demand delivery, ensuring freight is transported to the right destination by constantly tracking the position of freight and trucks. Tracking in this sense means that RFID readers are used to monitor the movements of RFID-tagged vehicles. With regard to the term vehicle, any mobile item used to carry freight or passengers is referred to as a vehicle. Thus, various kinds of pallets, forklifts, and other put away and load units fall under this category, as well as various passenger cars and cargo trucks.

RFID tracking applications in transportation and logistics are, in general, implemented in order to gather up-to-date information on tagged freight and its movements, thereby facilitating effective in-time management. For this reason, stolen or lost freight should be detected, as well as freight delivered incorrectly or with significant delays. This problem should not be underestimated because it has a huge impact on time and money, i.e., development of a stand-alone system for tracking freight. Therefore, RFID might be a stepping stone to achieving success in this field.

Without demonstrating other technologies, it can be stated that RFID does not require the establishment of a line of sight. Furthermore, RFID tags are resistant to environmental impact, such as physical interaction with other items. Moreover, RFID supports multiple object recognition, so that several tags can be read simultaneously. However, while it is possible to extend the list of RFID advantages, there are still some potential drawbacks that have to be kept in mind when using this technology for vehicle tracking. In order to mitigate the risk of unsuccessful RFID implementation, comprehensive requirements analysis should be performed beforehand.

5.1.5.2 RFID-Based Vehicle Tracking

Vehicle tracking systems are commonly used by fleet operators for fleet management functions such as:

- Fleet dispatching
- Fleet routing
- Fleet tracking

These activities are needed to monitor, control, and plan transportation processes based on information from onboard units, which require an extended data security approach. Along with commercial fleet operators, urban transportation agencies use this technology for a number of purposes, including monitoring the schedule

adherence of buses in service, triggering changes in bus destination sign displays at the end of the line (or other set location along a bus route), and triggering prerecorded announcements for passengers.

With regard to the aforementioned, vehicle tracking systems can also be understood as an integrated part of a layered approach to vehicle protection, recommended by the National Insurance Crime Bureau (NICB) to prevent motor vehicle theft. This approach requires at least four security layers based on the risk factors pertaining to a specific vehicle. Vehicle tracking systems are one such layer and are described by the NICB as very effective in helping police recover stolen vehicles.

In order to investigate the requirements for RFID-based vehicle tracking, several RFID vehicle tracking application use cases have been identified for further consideration. They are:

- Implementing RFID vehicle tracking in road systems
- Tracking tagged load units as a part of logistics and supply chain management

Figure 5.15 presents a simplified illustration of both use cases, showing a common RFID system structure and its interaction with other system components (Deriyenko 2012).

The first tracking application belongs to freight in logistics and supply chains. Here, freight items are usually tracked at several stages as they pass through the business workflow process. For better transparency, it is possible to perform tracking every time the freight approaches and leaves each stage.

There are several ways that RFID systems can be integrated into road systems. The first example is the implementation of RFID tracking of wagons by a Finnish railroad operator (Wessel 2011). Setting up readers along railroads allowed more precise information generation about a train's location at any particular moment in time. Another example of using RFID tracking with road systems is automatic payment on toll roads, such as those in Germany, which makes it possible to overcome severe problems, such as traffic jams, at toll points and reduction of labor costs (Xiao et al. 2008). The system consists of onboard units based on RFID chips usually fixed on windshields or bumpers of moving vehicles and RFID readers located at the toll stations. To ensure the system works effectively, each chip (tag) should be associated with a corresponding payment account. In a

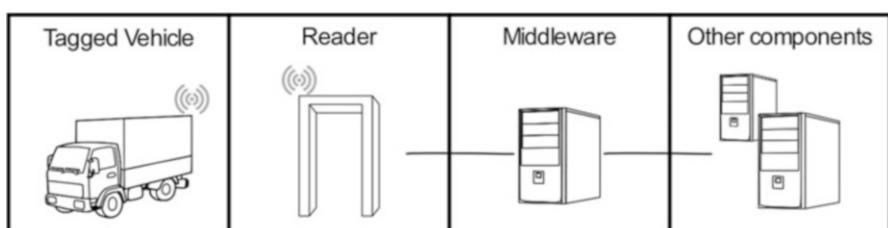


Fig. 5.15 RFID-based vehicle tracking system (Deriyenko 2012)

sunny day scenario, as a tagged vehicle enters the toll area and passes by the reader, its tag is requested to provide information for identification. Once the information is read by the reader, money is charged from the linked account.

Other examples illustrating the use of RFID in road systems are managing parking lots and tank gas stations (Pala and Inanc 2007; Mathis 2012). However, these approaches are very similar to the one used for toll collection.

5.1.5.3 Requirements Analysis

The requirements analysis for an RFID-based vehicle tracking system is essentially to identify, on one hand, the most relevant system constraints and, on the other hand, the essential knowledge required for conducting a system design that includes RFID readers, RFID tags, and RFID middleware. But the RFID-based vehicle tracking system should not only gather data, it also needs to preprocess data according to specific business operational rules. Against this background, the requirements analysis is based on analyzing available research projects and research papers published in the area of interest. Out of them, the following requirements have been identified (Deriyenko 2012):

- *Data cleanup:* Data gathered in its purest form is most likely not user-friendly and is not of great value to the user. For this reason, in most cases, preprocessing according to particular business needs is required. The middleware (see Fig. 5.15) should not only be part of an interconnection bus but should also perform such pivotal functions as cleaning up data, deleting duplicates, ordering, arranging data against selected granularity level, and carrying out other preprocessing operations, all aimed at preparing the data for further business usage, etc.
- *High throughput:* Implementation of RFID components is initially aimed at increasing system throughput ability; therefore, successful operation tracking a large volume of vehicles within a certain period of time is one of the most central requirements for vehicle tracking. Obviously, the importance of this requirement and other concrete indicators directly depends on particular business operational needs and constraints that have to be taken into account, particularly available financial resources.
- *Integration:* The main goal of the RFID-based vehicle tracking system is to provide the stakeholders in transportation and logistics with valuable, complete, and reliable information on time and in a convenient form. Creating a vehicle tracking system composed of RFID tags and readers results in generation of a certain amount of data, in some cases quite a significant amount. Gathering and stand-alone storage of this data do not make any sense; as it is only useful when it is available to the user. This results in an obvious requirement, that the RFID-based vehicle tracking system should be integrated with other enterprise information system components (with the help of middleware) in order to provide useable data.
- *Real-time operation:* In addition to handling a high volume of data quickly, the RFID-based vehicle tracking system should have feasible ways of providing the data to the user. In most cases, retrospective data gathered by tracking vehicles

has a certain value for business operations; but its importance cannot be compared to that of real-time operational data. Updating information received from readers within a short period of time is a vital requirement for vehicle tracking. However, the update rate may fluctuate according to the specifics of the business operation. Considering the use case on toll collection, there should obviously be no significant delays with processing vehicle data in order to perform payments. The same can be said for logistics and supply chain management activities. Users should be able to access the most up-to-date information about vehicle movements; otherwise, the whole system loses its advantage. Therefore, the whole RFID-based vehicle tracking system has to ensure a short response time. With regard to Fig. 5.15 and this requirement, however, other enterprise system components, meaning more than the one considered within this paper, have to be adapted to meet this need.

- *Reliability:* Reliability of the system depends on a number of influencing factors including radio-frequency interference, technical infrastructure, configuration and placement of readers and tags, etc. In general, problems that can arise while tracking can be roughly divided into two groups: false positives and false negatives. False positives owe their name to their origin. The system assumes items are present, while in reality they are absent or should not be taken into consideration, yielding a false positive recognition. This can happen for several reasons. On one hand, it is possible to be confronted with a situation where one item can be scanned two times, either by the same or by different readers. One of the solutions to that problem, mentioned in the literature, is to force tags to respond only when their first digits match the digits requested by the reader. However, this solution causes the whole system to become more overwhelmed. On the other hand, it is important to avoid reader signal collision if their reading ranges overlap. Another problem arises, if the system scans a tag that is supposed to be located beyond a reader's reading range. Therefore, tags and reader positions should be controlled properly with technical indicators of both devices, as well environmental specifics.

Nevertheless, some of the problems mentioned can be solved by a certain level of data preprocessing. But in general, the RFID-based vehicle tracking system should embed a feature that reduces the amount of false positives for real-time broadcasting by using effective anti collision solutions and requesting algorithms and other adequate approaches.

One of the most frequently mentioned problems that falls into the category of false negatives is the presence of metal or water, which affects tag readability and can be a problem for both false positives and false negatives. The reason is very simple; a tagged freight load may contain pallets carrying, for example, bottles of water. Also, mistracking of vehicles can cause numerous inconveniences and result in additional business operational costs.

There may be several reasonable solutions to overcoming these problems, e.g., using metal as an antenna or changing an antenna's impedance. In any case, regardless of the approach used, the RFID-based vehicle tracking system should

be able to overcome obstacles, such as metal and water, which are preventing tags from being read.

Moreover, the RFID-based vehicle tracking system must be able to detect if some of its components are down. This means that it has to have appropriate user notification algorithms. As mentioned earlier, the reader can perform a request using the first digits of the tag identification number. Theoretically, such an approach can help to reveal the absence or failure of necessary tags. However, this solution is not applicable for all cases of RFID-based vehicle tracking, since all tag identifications should be stored in the system.

For example, an RFID-based system can help to detect if a load pallet is missing or its tag is not readable; but, for obvious reasons, this is unfeasible, for example, in the case of toll collection. However, in the latter situation, missing tag functionality is not necessary since the absence or breakdown of the tag will be identified anyway due to the inability of a car to pass through the barrier gate without it. Anyway, the requirement mentioned above can be optional and refers not only to tag monitoring but to readers and middleware as well.

5.2 Internet of Things

The Internet is a global system of interconnected computer networks that use the standard Transmission Control Protocol/Internet Protocol (TCP/IP) to serve billions of worldwide users daily. It is a network of private, public, academic, business, and government networks, from local to global in scope. Originating from the Advanced Research Projects Agency Network (ARPANET) around 1970, the Internet became available in the 1980s; by 1990 it had grown from the initial communication framework into the most popular network in use.

The Internet has gained further importance and is currently experiencing a massive growth driven by the Internet of Things, in which the real and the virtual worlds are converging. The IoT is described as a self-configuring wireless network of sensors whose purpose is to interconnect objects or things. The IoT appears to be one step further on the path to ubiquitous computing by embedding computing everywhere and programming it to act automatically, making it omnipresent. CPS are based on converging real (physical) and virtual (cyber) components connected to the Internet forming a dynamic global network infrastructure with self-configuring capabilities and based on a standard and interoperable communication protocol, IPv6. IPv6 is the latest version, routing traffic across the Internet. It was developed by the Internet Engineering Task Force to replace Internet Protocol, version 4 (IPv4), to overcome the long-anticipated problem of IPv4: address exhaustion. Internet Protocol, version 6 (IPv6) uses a 128-bit address which theoretically allows access to 2^{128} addresses for identification and location. Also, IPv6 provides other technical benefits in addition to a larger addressing space. In particular, it permits hierarchical

address allocation methods that facilitate route aggregation across the Internet, thus limiting the expansion of routing tables. The use of multicast addressing is expanded and simplified and provides additional optimization for the delivery of services. Device mobility, security, and configuration have been considered in the design of the protocol. Internet Protocol, version 6, addresses are represented as eight groups of four hexadecimal digits with the groups being separated by colons.

A thing, object, or entity in the IoT is any possible item in the real world that joins the communication chain. Therefore, the initial main objective of the IoT was to combine communication capabilities characterized by data transmission.

The key object in the IoT is radio frequency identification (RFID) technology which enables the wireless connection of objects, things, or entities. An object, thing, or entity is any possible item in the real world that joins the communication chain. Therefore, the initial key objective of the IoT was to combine communication capabilities characterized by data transmission. Thus, the IoT can be thought of as the building of a global infrastructure for RFID or sensor radio technologies as a wireless layer on top of the Internet. Such a network of interconnected computers communicates with a wireless network of interconnected objects constantly tracking and accounting for millions of things, from parcels to razor blades to tires. These objects sometimes have their own IP addresses, are embedded in complex automotive systems, use sensors to obtain information from their environment, and use actuators to interact with it, e.g., air conditioning valves that react to the presence of people in a vehicle.

The growth in the forms of information and communication networks is evident by the widespread use of mobile devices. The number of connected mobile devices worldwide surpassed 2×10^9 in mid-2005 and was approximately 25×10^9 in 2015, as shown in Table 5.3.

From Table 5.3, it can be seen that the IoT represents the point in time when more devices are connected to each other than people are connected with/and/or/to devices. This has an impact on today's world and will change everything, including our daily lives, because staying connected has become an integral and intimate part of the 24/7 paradigm of everyday life for many millions of people.

Furthermore, the IoT has become an important concept in the global economy because wireless technology is making it possible to interact with the IoT from anywhere to everywhere at any time. This has opened the opportunity for new ubiquity-based products and services in the automotive domain with a high degree of innovation and a major impact on society and business.

Table 5.3 Connected devices in relation to world population in the third wave of computing (Möller 2016)

	Year				Increase
	2003	2010	2015	2020	
World population	6.3×10^9	6.8×10^9	7.2×10^9	7.6×10^9	+20.635%
Connected devices	500×10^6	12.5×10^9	25×10^9	50×10^9	+10 ²
Connected devices per person	0.0793%	1.8382%	3.4722%	6.5789%	+82.962%

5.2.1 Internet of Things Enabling Technologies

The availability of the Internet and advances in software and telecommunication services with the ability to connect every object and/or thing, with any object and/or thing, at any time, and in any media, have accelerated the worldwide penetration of the IoT paradigm. In particular, the basic idea that every object and/or thing can also be part of a tiny computer and/or microchip that is connected to the Internet has outperformed any forecast. The enabling technologies of the IoT are:

- Miniaturization
- Nanotechnology
- RFIDs
- Sensors and actuators
- Smart entities

In addition, the increasing processing power available in the smallest of packages and/or devices in networked computing is the fundamental enabler for the IoT paradigm. RFID and sensors, among other technologies, have been increasingly deployed and allow the real-world environment to be connected into the IoT networked services. Entity-to-entity-oriented IoT applications are monitored in real time, depending on their actual status, while the IoT automatically reacts. This has finally resulted in smart objects and/or things which can act smarter than objects and/or things which have not been tagged with a unique visual or invisible identification code and/or equipped with sensors and/or actuators. These new smart objects will obviously raise many issues, such as (Chaouchi 2010):

- Addressing, identifying, and naming
- Choice of transport models
- Communication models of connected objects and/or things
- Connecting technology of objects and/or things
- Economic impact and telecommunication value chain evolution
- Interoperability between objects and/or things
- Possible interaction with existing paradigms, such as the Internet
- Security and privacy

Most of the Internet services were designed to satisfy person-to-person interaction. In contrast, IoT services rely on easy location and tracking of connected entities which means a new dimension has been added to the world of anytime, anywhere connectivity for anyone, resulting in the connectivity for anything. In summary, the relevant characteristics of the IoT are:

- *Connections:* Multiplying and creating entirely new dynamic networks of networks and the IoT. The IoT is based on solid technological advances and visions of network ubiquity that are zealously being realized.

- **Connectivity:** Generating and processing data traffic on the IoT. Connecting entities can be wireless or wired. The IoT also allows the connection of heterogeneous entities.
- **Embedding:** Short-range mobile transceivers in a wide array of additional gadgets and everyday items, such as smartphones, are enabling new communication forms between people and things and/or entities, such as people-to-vehicle, and between things and entities themselves, such as vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), vehicle-to-X (V2X), and others.

As stated in a 2005 UN report:

Today, in the 2000s, we are heading into a new era of ubiquity, where the users of the internet will be counted in billions and where humans may become the minority as generators and receivers of traffic. (Biddlecombe 2015)

The resulting roadmap of the IoT is shown in Fig. 5.16 (URL7 2017).

Typical views of the IoT are the following application domains in conjunction with cyber-physical systems:

- **Smart city:** Collective concept for the holistic development designs to make cities more efficient and technologically advanced, greener, and more socially inclusive. These concepts include technical, economic, and social innovations, the progressive digitalization that is taking place, and the revolutionized energy sector, such as smart grids, as the basis of urban life. However, mobility and transportation are essential for a smart city to function properly. Therefore,

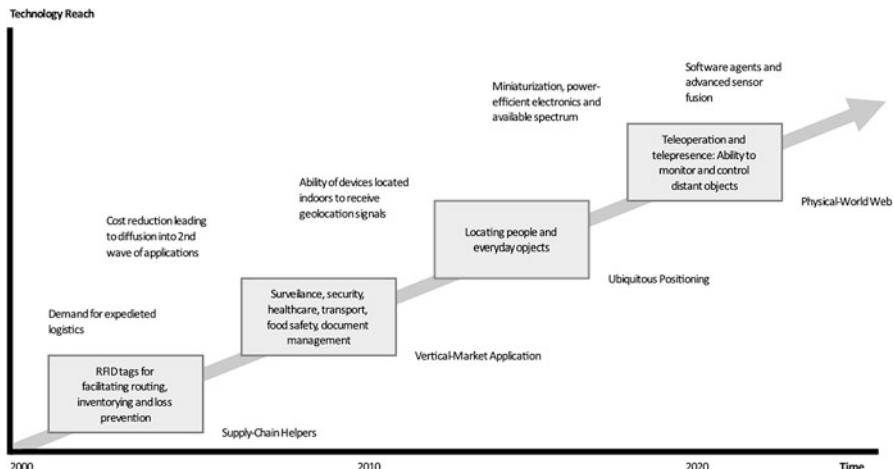


Fig. 5.16 Roadmap of the Internet of Things (Möller 2016) (Source: SRI Consulting Business Intelligence)

sharing concepts, traffic management systems, smart parking apps, and e-mobility belong to the essential features of a smart city to stay mobile, while keeping the city accessible and sustainable. Smart cities can be characterized with regard to their major attributes as follows:

- Digital City
 - Green City
 - Knowledge City
- *Smart street lights:* Light-emitting diode (LED) technology designed for energy efficiency with intelligence and sensors whose data can be used for a range of purposes, primarily to control when and where light is on, tracking people within the space incorporating wireless networking, and the ability to support cameras and environmental sensors, such as gas leak detectors and seismic monitors, to make life in an urban area of a smart city safer. Examples include:
 - Adaptive lighting to conserve energy, among all streetlights on a network
 - Communications capabilities (audio and visual display)
 - Digital street signs
 - Emergency response centers
 - *Smart mobility:* Ensures energy-efficient, comfortable, and cost-effective mobility that can be used intelligently by road users and can be interpreted as a slice of a smart city, crossing all of the mentioned before features and resulting in:
 - Reduced pollution
 - Reduced traffic congestion
 - Reduced travel costs
 - Improved safety
 - Improved travel speed
 - *Smart traffic lights:* Vehicle traffic control systems that combines traditional traffic lights with an array of sensors and artificial intelligence to intelligently route vehicular and pedestrian traffic, taking into account the natural flow of traffic which results in a certain traffic rhythm. With the use of sensors, actuators, and communication technologies, the arrival time of vehicles at the road intersection traffic light can be calculated by monitoring their actual speeds. Based on this calculation, it is assumed that the vehicle will arrive at the traffic light when it has changed from its red phase to the green phase. For this purpose CPS traffic lights have to take into account the real-world traffic flow that results in a certain traffic rhythm (Möller et al. 2015).

5.2.2 RFID and WSN Technology

In today's IoT paradigm, many things and/or objects will be part of the network in one form or another. This is where RFID and WSN technologies will meet this new approach, as the information and communication systems used are invisibly embedded in the environment.

RFID technology provides operational efficiencies and improves handling transparency in the logistics of on-demand distribution. RFID systems incorporate

microelectronic devices, called transponders, and reading units. Transponders are more commonly known as tags, and they are attached to the things and/or objects to be identified. Tags are available in a large variety of forms and functional characteristics and are classified as active and passive tags:

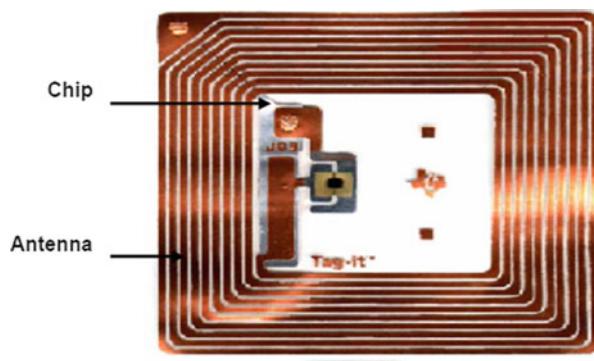
- *Passive tags*: Read/write range is shorter than most active tags; they do not possess an onboard source of power for signal broadcasting.
- *Active tags*: Read/write range is longer than most passive tags; they have their own power source for signal broadcasting.

Passive tags, such as the one shown in Fig. 5.17, are relatively inexpensive. They cost anywhere from a few cents to several dollars because they do not contain a power source. They draw power from a reader's radio signals which induce a current in the tag's antenna using either inductive coupling or electromagnetic capture. This power is used both for chip operation and broadcasting. These tags essentially reflect back the radio waves from the reader in order to broadcast, a phenomenon sometimes known as backscatter. However, their signal range is very low, usually less than 10 ft. Semi passive tags fall somewhere between the two; they use a battery for a chip's standby operation but draw energy from the reader during active broadcasting (Cisco 2008).

Low-cost tags, applicable to the grocery industry, cost from 20 to 35 cents; and the latest tag developments promise tags that will cost just around 5 cents (Kärkkäinen 2003). However, tags can also cost several dollars depending on many factors such as:

- Data capacity
- Form
- Operating frequency
- Range
- Performance requirements
- Presence or absence of a microchip
- Read/write memory

Fig. 5.17 Passive tag (Cisco 2008)



Passive RFID tags vary in how they broadcast to RFID readers and how they receive power from the RFID reader's inductive or electromagnetic field. This is commonly performed by two basic methods:

- *Load modulation and inductive coupling in the near field:* The RFID reader provides a short-range, alternating current, magnetic field that the passive RFID tag uses for both power and broadcasting. Through inductive (near-field) coupling, the magnetic field induces a voltage in the antenna coil of the RFID tag, which powers the tag. The tag broadcasts its information to the RFID reader. Each time the tag draws energy from the RFID reader's magnetic field, the RFID reader itself detects a corresponding voltage drop across its antenna leads. Tag can communicate binary information to the reader by switching a load resistor on and off to perform the load modulation. When the tag performs load modulation, the RFID reader detects this action as amplitude modulation of the signal voltage at the reader's antenna.
- *Backscatter modulation and electromagnetic coupling in the far field:* The RFID reader provides a medium-range electromagnetic field that the passive RFID tag uses for both power and broadcasting. Through electromagnetic (far field) coupling, the passive RFID tag draws energy from the electromagnetic field of the RFID reader. However, energy contained in the incoming electromagnetic field is partially reflected back to the RFID reader by the passive tag antenna. The precise characteristics of this reflection depend on the load connected to the antenna. The tag varies the size of the load that is placed in parallel with the antenna in order to apply amplitude modulation to the reflected electromagnetic waves, thereby enabling it to broadcast information payloads back to the RFID reader via backscatter modulation. Tags using backscatter modulation and electromagnetic coupling typically broadcast over a longer range than inductively coupled tags (Cisco 2008).

Active tags are typically used in real-time tracking of high-value assets in closed-loop systems, which usually justify the higher cost of the active tag. Active RFID tags are physically larger than passive RFID tags. They contain random access memory (RAM), which enables the active tag to store information from attached assets. This memory also makes active RFID preferable to passive RFID. Active RFID tag technology typically displays very high read rates and read reliability because of the higher transmitter output, optimized antenna, and reliable source of onboard power. The cost of active RFID tags varies significantly depending on the amount of memory, the battery life required, and whether the tag includes added-value features, such as onboard temperature sensors, motion detection, telemetry interfaces, and more. The durability of the tag housing also affects price, with the more durable or specialized housings required for specific tag applications available at higher costs. As with most electronic components of this nature, prices for active tags can be expected to decline as technological advances, production efficiencies, and product commoditization all exert a downward influence on market pricing (Cisco 2008).

Radio-frequency identification is now widely used for tracking things and/or objects and others. Hence, the RFID system architecture is marked by a sharp dichotomy of simple RFID tags and an infrastructure of wireless networked RFID readers. This architecture optimally supports the tracking of physical things and/or objects within well-defined confines but limits the sensing capabilities and deployment flexibility that more challenging application scenarios require.

Compared to passive tags, active tags are more expensive. Typically more than \$20 each, they provide a longer read/write range of up to 100 feet or more. They offer greater functionality, and their battery life is up to 1 year (Zaheruddin and Mandaviwalla 2005).

Comparing RFID and the bar codes used today highlights the considerable strengths of RFID: RFID does not require line of sight between tags and a reader in order to be read, tags can be read through non-metallic materials, and approximately 60 tags can be read simultaneously (Kärkkäinen 2003).

An RFID system includes:

- Transponders (tags) that allow items to be identified
- Antennas and readers/writers that allow tags to be interrogated and to respond
- Software that controls the RFID equipment, manages the data, and interfaces with enterprise applications

Table 5.4 provides approximate values for the characteristics of high- and low-frequency tags. The exact values depend upon a combination of factors, such as tag type (active or passive), presence of radio noise or radio wave absorbing materials in the environment, the size and gain of the antenna, and the type of reader (Zaheruddin and Mandaviwalla 2005).

With advancements in microelectronic components and the related miniaturization of intelligent functions, wireless sensors can be implemented in decentralized locations where they are needed. Particularly important in this context is communication outside of the immediate network. It should be noted that communication often takes place through the user interface of the device itself, which calls for more advanced technologies. This has been achieved in recent years by the development of networked sensors, the WSNs. Intelligent sensor nodes are wirelessly linked to computer networks. Current and planned applications of WSNs range from early warning systems in production control to so-called smart dust.

Table 5.4 Characteristics of active and passive tags

Tag frequency	General tag type	Approximate		
		Range	Transmission rates	Power consumption
Low	Passive	<1.0 m	1–2 kb/s	20 µW
High		1.5 m	10–20 kb/s	200 µW
Ultrahigh		10–30 m	40–120 kb/s	0.25–1.0 W
	Active	20–100 m*		

*With battery-powered tags

Smart dust belongs to one of the three forms of devices for a ubiquitous computing paradigm proposed by Marc Weiser (Weiser 1991) and can be considered as useful ubiquitous devices as introduced in Poslad (2009). Thus, smart dust is composed of systems of many tiny microelectromechanical systems (MEMS) (see Sect. 4.2.7), ranging from millimeters to micrometers to nanometers. Examples are sensors that detect physical or chemical quantities or are integrated into smart clothes (the integration of sensors, actuators, computers, power sources, etc. into the cloth, the whole being part of an interactive communication network). Smart dust is usually wirelessly operated on a computer network and distributed over a specific area to perform tasks, such as using RFID to sense a smart dust component introduced through the IoT paradigm. The size of an antenna for a tiny smart dust communication device ranges from a few millimeters to centimeters, and it may be vulnerable to electromagnetic disablement and destruction by microwave exposure.

5.3 Telematics, Infotainment, and the Evolution of the Connected Car

Telematics refers to the use of wireless components and technologies to transmit data in real time within a network. Telematic components are typically used in vehicles to collect and transmit data on vehicle use, maintenance requirements, or vehicle servicing. Telematics can also serve as a platform for the usage-based insurance (UBI) premiums business, also known as pay as you drive (PAYD) and pay how you drive (PHYD) programs, as well as fleet insurance and other telematics features. Infotainment is a made-up word combining the terms “information” and “entertainment,” and refers to a type of media which provides a combination of information and entertainment. In this regard, the term also refers to hardware/software products and systems which are built into, or added to, vehicles in order to enhance driver and/or passenger experience.

The automotive industry is transforming. Automakers are focussing on the innovative area of connected cars and autonomous vehicles, which requires new technological solutions for designing vehicles around user needs and not technology only, as well as the development of new business models. Therefore, the challenge in designing connected cars is to take all of the new technology being developed for fully autonomous vehicles, advanced driving assistance systems (ADAS), and predictive intelligence and tap into an entirely new paradigm with regard to IoT, autonomous driving, connected vehicles (V2X), and predictive intelligence. Ultimately, all this has to be embedded in new business models for automakers, service providers and Tier 1 suppliers.

5.3.1 Telematics

Telecommunications and informatics (telematics) applied in wireless technologies and computational systems are the basis for today's vehicular advanced telematic concepts such as:

- *Advanced driving assistance systems (ADAS)*: Automotive electronic components developed to support the driver in the driving process and to enhance vehicle systems with regard to safety and better driving (see Sect. 4.9 and Chap. 11).
- *Hands-free cell phone interfaces (HFCPI)*: Allow drivers to initiate hands-free cell phone calls, text/browsing during hands-free calls, and end hands-free calls with a hands-free cell phone. The hands-free interface also requires that drivers enable a Bluetooth® connection, pair their cell phone, and manually dial if their voice commands are not recognized (Fitch et al. 2013).
- *Emergency warning system for vehicles (EWSV)*: Telematic concepts developed particularly for international harmonization and standardization of vehicle-to-vehicle (V2V), roadside-to-vehicle (R2V), and vehicle-to-roadside (V2R) real-time dedicated short-range communication systems. Instantaneous direction travel cognizance of a vehicle may be transmitted in real-time to surrounding vehicles equipped with EWSV and traveling in the local area to receive warning signals of danger.
- *Satellite navigation*: Telematic concept using a Global Positioning System (GPS) and electronic mapping tool to enable the driver to locate a position, plan a route, and navigate a trip.
- *Wireless vehicle safety communications (WVSC)*: Telematics concept in vehicle safety and road safety. It is an electronic subsystem in a vehicle or in another vehicle for the purpose of exchanging safety information, such as road hazards and the locations and speeds of vehicles, over near-field communication short-range radio links which are wireless communication channels specifically designed for automotive use. This may involve temporary ad hoc WLANs. The WVSC wireless local area networks (WLANs) are based on the IEEE 802.11p standard and are marketed under the Wi-Fi® registered trademark. Wi-Fi is the name of a non-profit international association which certifies interoperability of wireless LAN products based on the IEEE 802.11 standard.

V2V, R2V/V2R, vehicle-to-home (V2H), and vehicle-to-enterprise (V2E) are new communication features of the vehicle information technology (IT) between vehicles and other objects. In general, this communication is also called V2X communication, where X can stand for vehicle, infrastructure, or other things.

- *V2V*: Direct information exchange between moving vehicles. V2V communication is the early warning system of vehicle drivers and passengers to

avoid accidents and to prevent critical driving maneuvers. Another goal is the optimization of traffic flow through fast and early information exchange about traffic situations and unfavorable weather conditions, such as severe rain with a hydroplaning hazard or icy roads.

- **R2V/V2R:** Vehicles communicate with infrastructure devices and vice versa. Traffic control systems, smart traffic light systems, finding free parking spots, and automatic parking belong to R2V/V2R. In addition to the aforementioned cases, information on congestion and accidents can be communicated, along with early warnings when emergency vehicles from police, fire department, and medical rescue are approaching to immediately open a corridor for emergency vehicle access.
- **V2H:** Communication between a vehicle and home appliances. For example, using V2H, the vehicle driver or passenger can ask whether the coffee machine is switched off at home, the bathroom window is closed, the entrance door is locked, or other possible applications.
- **V2E:** Represents communication between vehicles and infrastructure that are operated privately and commercially. This includes, for example, the communication of a vehicle with a parking garage, where a parking garage with a free car park is located. The vehicle is then navigated to it (Johanning and Mildner 2015).

In this regard, 802.11p, also referred to as wireless access for the vehicle environment (WAVE), is the primary standard that addresses and enhances telematic concepts for applications in intelligent transportation systems (ITS). Intelligent transportation systems apply information and communication technologies in the field of road transport, including infrastructure, vehicles, and users, and in traffic management and mobility management, as well as for interfaces with other modes of transport. In this regard, the term telematics refers to an interdisciplinary field of applications that encompasses:

- Computer science applications, such as:
 - Internet usability
 - Multimedia usability
- Engineering applications based on:
 - Instrumentation technology
 - Sensor technology (see Sect. 4.2.7)
 - Wireless communication networks
- Road safety
- Road transportation
- Telecommunications
- Vehicular technologies

The application of telematics concepts can involve any of the following (URL8 2017):

- *Global Navigation Satellite System (GNSS) technology:* Integrated with computers and mobile communications technology in automotive navigation systems, which mostly use a satellite navigation device for position data that is then correlated to a position on a road
- *Integrated use of telecommunication and informatics:* Applied in vehicles and for control of vehicles on the move
- *Technology features:* Receiving, sending, and storing information through telecommunications devices in conjunction with exerting control of remote objects
- *Use of such systems:* With regard to road vehicles, also called vehicle telematics

Vehicle telematics can help to improve efficiency in the transportation and logistics business. The domain-specific applications include:

- *Container tracking:* Freight containers can be tracked by GPS using, e.g., a battery-powered GPS device communicating its position via mobile phone or satellite communications. Benefits of this approach include increased security and the possibility of rescheduling the container movements based on accurate information of its location.
- *Fleet management:* Includes the activities required to control, monitor, and plan transportation processes based on vehicles such as buses, cars, trucks, aircraft, trains, and vessels. Fleet management includes a range of management functions based on data regarding the current state of the transportation systems, such as:
 - Dynamic vehicle scheduling with regard to traffic conditions to divert them to alternative routes and away from congested routes
 - Vehicle driver's working hours
 - Vehicle fuel management
 - Vehicle health and safety management
 - Vehicle maintenance based on odometer information
 - Vehicle telematics (tracking and diagnostics)

Regarding the transportation of hazardous materials, information concerning the type and state of the loaded shipments can be of significant help in taking appropriate measures in the event of an accident.

Fleet management allows motor carrier companies that rely on transportation and logistics in their business to remove or minimize the risks associated with vehicle investment by improving efficiency and productivity, reducing their overall transportation costs, and providing 100% compliance with government legislation and duty of care obligations. Duty of care requires an individual to adhere to a standard of reasonable care by taking action when harm to others is foreseeable.

The most challenging fleet management task is vehicle scheduling, which includes determining which vehicle should approach which cradle, which station of delivery, or which maintenance service station at what time. The generation of

schedules has a considerable impact on the motor carrier's profit. Hence, schedules have to be generated such that the difference between the revenue gained and the cost of fleet vehicle movement is minimized (Goel 2008).

- *GPS tracking:* Usually accurate to around 10–20 m, the European Space Agency (ESA) has developed the European Geostationary Navigation Overlay Service (EGNOS) that supplements the GPS, the Global Navigation Satellite System (GLONASS), and the Galileo system by reporting on the reliability and accuracy of positioning data. EGNOS technology is accurate to 1.5 m (4 feet).
- *Vehicle tracking:* Monitoring the location, movements, status, and behavior of a vehicle or fleet of vehicles. This is achieved through a combination of a GPS/GNSS receiver and an electronic device usually comprised of a GPS/GSM/GPRS modem or short message service (SMS) sender installed in each vehicle, communicating with the user dispatching emergency or coordinating unit and PC- or web-based software. The General Packet Radio Service (GPRS) is a packet-oriented mobile data service available to users of the 2G cellular communication systems, Global System for Mobile Communications (GSM), as well as in the 3G systems. In the 2G systems, GPRS provides data rates from 56 up to 114 kbit/s. The data is turned into information by management reporting tools in conjunction with a visual display on computerized mapping software. Vehicle tracking may also apply odometry which uses data from motion sensors to estimate changes in position over time or dead reckoning which is the process of calculating a current position by using a previously determined position, or fix, and advancing that position based upon known or estimated speeds over elapsed time and course, as an alternative or complementary means of navigation.
- *Trailer tracking:* Tracking movements and the position of a vehicle's trailer unit through the use of a location unit fitted to the trailer and a method of returning the position data via a mobile communication network or geostationary satellite communications, for use through either PC- or web-based software. Trailer tracking systems require four essential components to run:
 - Backend server and database
 - Communication network
 - Tracking device
 - User interface software

Cold storage freight trailers that deliver fresh or frozen foods are increasingly incorporating telematics to gather time series data, a series of data points listed in time order, on the temperature inside the cargo container, both to trigger alarms and record an audit trail for business purposes. An increasingly sophisticated array of sensors, many incorporating RFID technology (see Sect. 5.2.2), are being used to ensure the cold chain.

The Association of Equipment Management Professionals (AEMP) developed the industry's first telematics standard. In 2008, AEMP brought together the major construction equipment manufacturers and telematics providers in the heavy equipment industry to discuss the development of the industry's first telematics

standard. Following agreements with industry to support such a standard, the AEMP formed a standards development subcommittee to develop it. The group developed the industry's first standard for the delivery of telematics data (URL8 2017). The AEMP's telematics data standard was developed to allow end users to integrate key telematics data into their fleet management reporting systems. As such, the standard was primarily intended to facilitate importation of these data elements into enterprise software systems, such as those used by many medium to large construction contractors. Prior to the standard, end users had few options for integrating this data into their reporting systems in a mixed-fleet environment consisting of multiple brands of machines and a mix of telematics-equipped machines and legacy machines. One option available to machine owners was to visit multiple websites to manually retrieve data from each manufacturer's telematics interface and then manually enter it into their fleet management program's database. This option was cumbersome and labor intensive.

A second option was for the end user to develop an application programming interface (API), or program, to integrate the data from each telematics provider into his or her database. This option was quite costly, as each telematics provider had a different procedure for accessing and retrieving the data; the data format varied from provider to provider (URL8 2017).

A third option for mixed-fleet integration was to replace the various factory-installed telematics devices with devices from a third-party telematics provider. Although this solved the problem of having multiple data providers requiring unique integration methods, this was by far the most expensive option. In addition to the expense, many of the third-party devices available for construction equipment are unable to access data directly from the machine's electronic control modules (ECMs), or computers, and as such are more limited than the device installed by the OEMs in terms of the data they are able to provide. In some cases, these devices are limited to location and engine run time, although they are increasingly able to accommodate a number of add-on sensors to provide additional data (URL8 2017).

The AEMP Telematics Standard provides a fourth option. By concentrating on the key data elements that drive the majority of fleet management reports, making those data elements available in a standardized XML format and standardizing the means by which the document is retrieved, the standard allows the end user to use one API—the ability to integrate third-party software applications—to retrieve data from any participating telematics provider. Because one API can retrieve data from any participating telematics provider, as opposed to the unique API for each provider that was required previously, integration development costs are greatly reduced.

In addition to the new data fields, the AEM/AEMP Telematics API Standard also changes how the data is accessed in an effort to make it easier to use and integrate with other systems and processes. It includes standardized communication protocols for the ability to transfer telematics information in mixed equipment fleets to end user business enterprise systems, enabling the end user to employ their own business software to collect and then analyze asset data from mixed equipment fleets without the need to work across multiple telematics provider applications (URL8 2017).

The AEM/AEMP Telematics Standard has been approved by the International Organization for Standardization (ISO) and issued as ISO/TS 15143-3:2016, Earth-Moving Machinery and Mobile Road Construction Machinery—Worksite Data Exchange—Part 3: Telematics Data.

5.3.1.1 Carsharing

Telematics technology has enabled new services like carsharing which is a model of vehicle rental for:

- Booking decisions of customers based on spontaneous demands
- Customer mobility support
- Multiperiod usage
- Predetermined customer arrivals at a carsharing station
- Short period of time usage
- Stochastic customer arrivals
- Uncertain operating usage

Carsharing is attractive to customers who occasionally use a vehicle, as well as others who would like occasional access to a vehicle of a different type than they use day to day. Carsharing follows the trend of today's sharing economy (Meyer and Shaheen 2017). It allows customers to share resources, such as equipment, services, and skills, with one another, often at significantly lower cost than traditional rentals. The renting organizations are using a commercial business model. They could also be organized as a company, a public agency, a cooperative agency, or an ad hoc grouping, such as in ridesharing. With regard to carsharing, telematics-enabled computers allow new business models to track customer usage and bill customers on a PAYD basis. Furthermore, some of the carsharing systems show customers where to find an idle vehicle in a station-based vehicle fleet or in free-floating carsharing. Others use telematics features to monitor and report on vehicle use within predefined geofence areas in order to demonstrate the reach of a transit media's vehicle club fleet.

The carsharing model for combustion vehicles has some different aspects with regard to an e-car carsharing model. A major concern in the e-car carsharing model is driving distance due to the limited battery capacity of today's e-cars and the relatively long battery charging time at the respective loading stations. With the implementation of the necessary infrastructure in the near future, e-carsharing will also become a successful business model. In the meantime, a lot of research work needs to be done to determine the ideal operating strategies.

5.3.1.2 Vehicle Insurance

The basic idea of telematics vehicle insurance is that a driver's behavior is directly monitored while the person drives, and this information is transmitted to the vehicle insurance company that is providing financial protection against physical damage and/or bodily injury resulting from traffic collisions and against the liability that could arise. The insurance company then assesses the risk of that driver having an

accident and charges UBI premiums accordingly, also known as PAYD and PHYD. The costs depend on the type of vehicle used, measured against time, distance, behavior, and place. Hence, a driver, who drives less responsibly, will be charged a higher premium than a driver who drives smoothly and with less calculated risk of claim propensity. Other benefits can be delivered to end users with telematics, as customer engagement can be enhanced with direct customer interaction. Using the smartphone as the in-vehicle device for tracking and monitoring is of great interest for insurance telematics and automotive electronics applications.

5.3.1.3 Smart Ticketing

Smart ticketing is characterized by the usage-dependent billing modality which supports the smart mobility approach. This ensures the release and use of the means of transportation and guarantees its availability. The risk involved with the cashless transactions is mitigated by use of an identification device. Therefore, a process-oriented procedure covering the entire travel cycle can be achieved. This is independent of the traveler's route selection and chosen means of transport, but it depends on which provider is offering the transport route, unless the traveler has made special requests.

In order to implement a smart ticketing system, intelligent usage-based billing is required. It can be composed of two elements:

- User identification
- Intermodal-oriented order acceptance and payment system

In addition, the smart ticketing system requires a minimum data record for the unambiguous identification of a previously unknown customer. This enables the new customer to book and settle user-related transport orders. In the case of a well-known customer, preferences regarding the means of transport and frequent routes and services are stored in the customer's profile. These preferences can be, for example, the driver class, window or gangway, rest zone, large compartment, preferred space, and other wishes.

Smart ticketing can also be used to implement process- and personal-related pricing in a simple manner, for example, through the following, along with many more offers (Belay 2016):

- Bonus programs
- Discounts
- Special season ticket prices
- Subscription prices
- User-related offers

5.3.1.4 Machine-to-Machine Telematics

Machine-to-machine (M2M) data modules are sophisticated and come with an array of features and capabilities, such as:

- Embedded Java
- Embedded M2M optimized smart cards, known as machine identification modules (MIMs) or M2M identification modules
- A flexible land grid array surface mounting
- GNSS technology

to step up the IoT. Global navigation satellite system, short-range, 2G, 3G, and even 4G communication modules are the technologies that facilitate M2M communication. There are several telematics devices that are part of the transportation segment:

- Electronic toll systems
- Infotainment systems
- Navigation
- Stolen vehicle recovery
- Vehicle diagnostics

These components support receiving the parking, toll, and vehicle information with real-time updates. M2M is also gaining acceptance in various subsidiary industries, such as automobile leasing, fleet management, and related sectors. The rising demand for embedded telematics in vehicles is expected to drive down the prices of devices and make it affordable for companies to incorporate them. However, the lack of awareness and cost sensitivity could pose a challenge in this sector. Globally, automakers are working on embedding M2M technologies in their vehicles. Deployment of M2M telematics applications in the automotive industry is assumed to help decrease the number of road accidents and damage. Deployment of similar products will lead to the automotive sector supporting the digital transformation in vehicles.

Furthermore, today's vehicles have integrated computer systems and other automotive electronic gadgets which support the driver. Machine-to-machine telematics provide vehicles with fully loaded sensor technology, with all of the essential information about engine performance, temperature, fuel, and so forth. It is expected that all new cars will be Internet-enabled by 2025 through IoT/M2M solutions. Increasing demand for connected devices, rising awareness, and penetration of smart devices are some of the factors driving the growth for IoT and M2M solutions in the market (Gulati 2015a, b).

The telematics market segments with regard to the respective services are summarized in Fig. 5.18.

5.3.2 Infotainment

Integrated infotainment systems in vehicles, so-called in-vehicle infotainment (IVI), deliver entertainment and information content to vehicle users. The content delivered via infotainment is designed to be informative yet entertaining enough to attract and maintain the consumer's interest. In this regard, infotainment refers to a variety of content served through traditional media, such as the Internet, radio, television, and others.

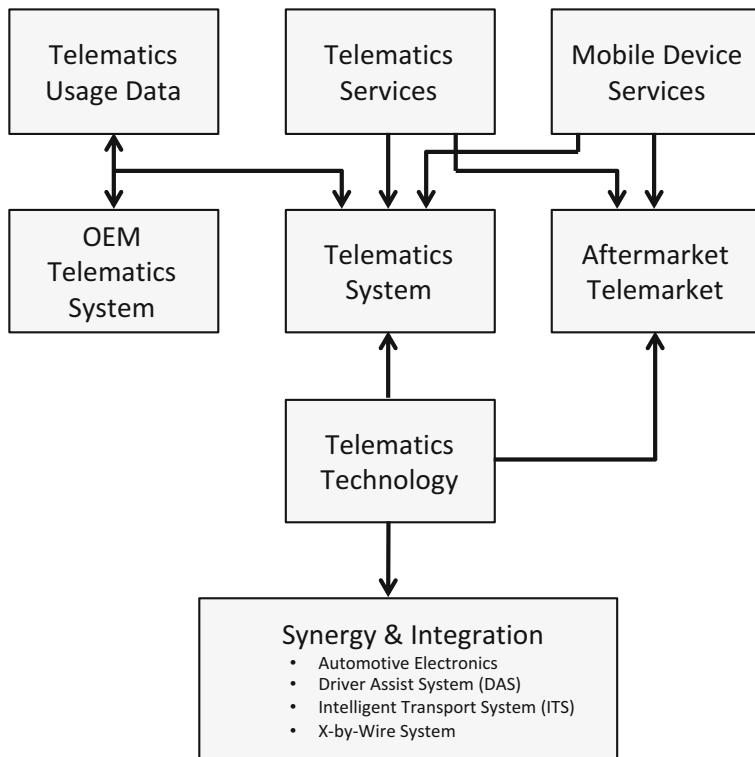
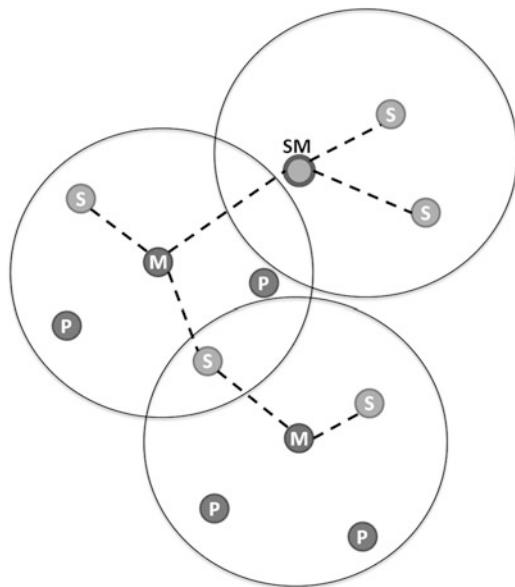


Fig. 5.18 Telematics market segments, modified (Juliusen 2003)

IVI systems frequently utilize Bluetooth technology and/or smartphones. Bluetooth is a wireless short-range radio technology simplifying communications among Internet-enabled devices and between devices and the Internet. It also aims to simplify data synchronization between Internet devices and other computers. A master Bluetooth device can communicate with a maximum of seven devices in a piconet, an ad hoc computer network using Bluetooth technology. The devices can switch roles, by agreement, and the slave can become the master. The Bluetooth core specification provides for the connection of two or more piconets to form a scatternet, a type of ad hoc computer network consisting of two or more piconets, as shown in Fig. 5.19, in which certain devices simultaneously play the master role in one piconet and the slave role in another. The master chooses which slave device to address; typically, it switches rapidly from one device to another in a round robin fashion. Since it is the master that chooses which slave to address, whereas a slave is (in theory) supposed to listen in each receiving slot, being a master is a lighter burden than being a slave.

While each IVI system is different, typical tasks that can be performed with an IVI system include managing and playing audio content; utilizing navigation for driving; delivering rear-seat entertainment, such as movies, games, social networking, etc.; listening to incoming and sending outgoing SMS text messages; making

Fig. 5.19 Scatternet with master (M) indicated as dark gray dots, slave (S) indicated as light gray dots, and piconet (P) indicated as gray dots (URL9 2017)



phone calls; and accessing Internet-enabled or smartphone-enabled content, such as traffic conditions, sports scores, and weather forecasts.

With regard to social networking, several social media websites and embedded Web 2.0 Internet-enabled applications allow the creating, exchanging, and sharing of user-generated content, such as text posts or comments, digital pictures, or video posts, in virtual communities and networks.

In addition, many vehicles now have a constant connection with the cloud of the respective automaker where vehicle data can be stored or social media services can be downloaded to the vehicle's onboard unit (see Sect. 5.4.2). In addition to Google and Facebook®, this includes news and weather services. In this regard, Google Maps™, which provides real-time traffic data, can be directly integrated into the onboard unit of the vehicle. This corresponds to the further development of the traffic message channel (TMC) feature with more up-to-date information on the radio, such as congestion information.

5.3.3 Evolution of the Connected Car

Connected car technology first emerged in the mid-1990s, with a focus on technology-driven telematics concepts. Today, the connected car technology has the potential to radically change the way people travel because vehicles are already moving computer systems that run on data as much as they depend on fuel. It promises to greatly:

- Reduce road accidents
- Reduce greenhouse gas emissions
- Speed up commuting and freight traffic

The impact on economy and lifestyle is looking increasingly impressive, and there is strong support and enthusiasm emanating from government, automakers, and their OEMs and Tier 1 suppliers. Thus, automakers all over the world are currently developing, presenting, producing, and marketing new vehicle features that enable the exchange of information with the Internet through specific interfaces, bringing the Internet into the automotive world. The joint connection of vehicles and the Internet opens up new possibilities for the entire automotive industry, providers of Internet services, and their customers who use these features, along with the challenges, opportunities, and risks this development entails. In Siebenpfeiffer (2014) a detailed overview on connected cars is given which covers the aspects safety, Car IT, and concepts.

Some automakers have broadened their overall mission and have gone beyond the production of vehicles to become real mobility providers through connected cars. This change in mission clearly has implications for the strategic motivation for in-vehicle services and the automaker's overall business model. The digital transformation paradigm has elevated the challenges, opportunities, and risks of this development because vehicles computing capabilities make endpoint security a moving target. Endpoint security for mobile computers and other vehicles devices is identified as a point of risk. Modern vehicles have as many as 20–200 moving end points that are connected to each other and to the networks of any businesses providing services to those vehicles. This enables the connected car to exchange information between itself and its environment via the Internet, whereby the vehicle's connection to the Internet is provided either by a transmitter/receiver unit built into the vehicle itself or through third-party systems, such as smartphones and/or tablets. This transforms the vehicle into a hub of communications enabling in-vehicle use of data and services through appropriate operating and display concepts. Features users expect in vehicle-connected services are, as a minimum:

- Driver safety
- Navigation
- Security

In addition, applications that enhance connectivity may be desired but not required. Furthermore, cloud applications (see Sect. 5.4.2) and a Hypertext Markup Language 5 (HTML5) engine to render content in the vehicle are seen as appropriate services. HTML5 is the fifth edition of the Hypertext Markup Language, a computer language for marking and linking text and other contents of electronic documents, mainly on the Internet.

The demand for in-vehicle services is primarily characterized by three factors:

- Individualization
- Relation to mobility
- Vehicle brand identity

According to Bechmann et al. (2015), 95% of the respondents consider the growing demand for mobility-related information as a major factor driving connected car development. Services such as enhanced navigation or traffic information are seen as highly relevant using these functions, with the quality and timeliness of the content being a major success factor. There is above-average willingness to pay for these services. This also reflects the high potential for new telematics functions in the business-to-business (B2B) sector. This trend represents an entirely new market for mobility services, services that bring people from one place to another and support them during and after a trip. In addition to navigation and traffic information, systems offer a variety of other features that can be considered, such as:

- *Driver Wellness Monitoring wizard*: Fatigue sensors and other features
- *Entertainment features*: Video streaming, Wi-Fi hotspots, and other features
- *Mobility management*: Current high-volume traffic information, gas consumption optimization, and other features
- *Safety functions*: Danger warnings, emergency functions, and other features
- *Vehicle management*: Remote control, maintenance information, and other features

With each step of automation up to fully autonomous driving, these vehicle business models will become even more important. New mobility concepts will include the joint use of vehicles to provide flexible mobility.

However, vehicle individualization is also an important factor for younger drivers in particular. Young customers who are enthusiastic about adopting new technology will want vehicles that function as an extension of their virtual environment, sporting features like individual user profiles; personalized services; and social media support.

Therefore, as reported in Bechmann et al. (2015), automakers are pushing services and offer into the market that fit their brand. Automotive OEMs are pursuing vehicle connectivity for the state-of-the-art image it conveys, with the ultimate goal of achieving strategic differentiation between themselves and the competition and earning customer loyalty. The services offered range, as previously mentioned, from safety enhancements through information on the vehicles' surroundings, to convenience services, and even to improved connection to individualized, customer-specific service points.

One possible convenience service is infotainment, a feature that their respondents view as being very important to customers. Freedom to use the Internet seems to be less important, with most respondents considering it to be a basic function. In both cases, willingness to pay is no higher than with smartphones or the levels usually seen for media balancing value and cost of services. However, consumer information becomes interesting when services relate to a specific location or to mobility.

The demand for vehicle connectivity is high, and technology will initially be used by OEMs and Tier 1 suppliers for strategic positioning in the market. Profitable business models are expected, spurred by new applications offered by various providers (see Sect. 2.3). The expected value creation potential is enormous. The overall market for connected cars in 2015 was about € 32 billion. By 2020, considerable growth to about € 115 billion is expected. Here alone, the market volume for the functional areas of safety and autonomous driving is growing to an estimated € 83 billion, which occupies two thirds of the market (see Sect. 2.4.). According to a market potential of € 32 billion, as reported in “Connected car opportunities and risks for future provider” in the automotive market Bundesverband Digitale Wirtschaft (BVDW) position paper ([URL10 2017](#)), desirable functions or experience areas include:

- Entertainment
- Home integration (see Sect. 5.3.1)
- Mobility
- Well-being

However, connectivity decisions are an important factor in determining the value proposition and the supporting business model for connected car services. Connectivity service trends for in-vehicle services include:

- *Embedded solutions*: Connectivity and applications are built directly into the vehicle, where today they are platform dependent.
- *Remote solutions*: Applications reside in a smartphone.
 - Remote skin resides in the vehicle which controls smartphone applications. Connectivity is achieved by universal serial bus (USB) or Bluetooth.
 - Remote terminal client (such as Virtual Network Computing (VNC)) in the vehicle replicates the smartphone HMI. Connectivity is achieved by USB or Bluetooth.
- *Tethered solutions*: Applications reside only in the vehicles. Smartphones provide connections to the cloud. Connectivity is achieved through USB or Bluetooth.

The first development stage to the connected car will encompass technological integration and infrastructure development. Based on considerable self-driven

activity, various market players will emerge that will introduce new in-vehicle applications for various groups of users, such as:

- Platform providers
- Software developers
- Telecommunications carriers

Automotive OEMs will have to decide whether to open their systems to these applications or to continue to favor their own proprietary applications. In line with this trend, the connected car will evolve in the market over several stages.

Luxury and high-end, trendy city vehicles, especially those associated with alternative drive concepts, such as e-vehicles, will be at the forefront of technology and innovation. Vehicle connectivity will become the standard in high-end vehicles with about 80% of vehicles in this class having these features soon. These vehicles will be equipped with their own transmitter/receiver units. At the same time, city vehicles, about 33% of which will feature connectivity, will appeal particularly to young, urban buyers. Connectivity in this segment will be provided by users' own smartphones. The challenge for OEMs in terms of this target group is to provide attractive services at the speed and quality these users are accustomed to receiving from the Internet community.

The segment of customers who are young, technology minded, but less willing to pay will demand low-cost, flexible solutions. The market potential for aftermarket in-vehicle connectivity solutions is certainly interesting, at 20% of existing vehicles. Because of the limited number of interfaces for retrieving data from the available vehicles, the scope of functions provided in these aftermarket solutions will probably be limited to providing services that largely do without data from the vehicle itself.

Since development of additional applications takes place outside of the automotive life cycle and is driven by individual business models, the aftermarket, with its focus on applications, will gain tremendous momentum in the coming years. Thus, automakers will be under pressure to ensure customer loyalty in the aftermarket segment so that they do not lose customers to third-party providers because aftermarket features are generally installed directly at OEM dealerships or at locations operated by automotive accessory specialists (Bechmann et al. 2015).

From the above it can be seen that the technologies involved in realizing the connected car currently exist in the automotive, software, and telecommunications industries and their OEM and Tier 1 suppliers. However, a major concern with regard to the development of value creation networks is to close competency gaps among individual partners with regard to the development of a shared understanding of the entire value chain. This has a huge impact on the risk of accidents and damage claims related to connected cars, especially if claims are the result of cyberattacks (see Chap. 6), which makes ensuring vehicle security an important issue.

Another major concern, especially for the automotive OEMs, are the long vehicle product life cycles, which are diametrically opposed to the short life cycle in the consumer goods and telecommunications industries. This calls for flexible architectures in connection with the far-reaching standardization of interfaces for

easy, smooth data integration as well as a special focus on the use of middleware. The challenge is, therefore, to ensure that the appropriate technologies for different vehicle segments are used throughout the vehicles' life cycles. Recommended actions given in (Bechmann et al. 2015) are:

- *All service providers:*
 - Decouple the process from the innovation, product creation, and update cycles of all value creation partners
 - Define safety and security standards
 - Define standards and interfaces
 - Define strategic portfolios of services, with high-quality mobility services as the starting point
 - Develop and establish cooperation strategies and collaborative models for potential partners
 - Outline flexible architectures
- *Automakers:*
 - Achieve branding through unique selling propositions (USPs) in the range of options offered for connected cars
 - Ensure customer loyalty in the aftermarket segment
 - Ensure elevated innovation
 - Map out product value and platform strategy for different target groups
 - Scale system between specific services related to exclusive brands and generally open services
- *Service and platform providers:*
 - Develop core competencies for intelligent data networking
 - Develop strategic portfolios of services with the potential to generate revenue
 - Review possible sources of revenue with downstream service providers, such as aftermarket service providers, infrastructure operators, OEMs, and others
- *Telecom infrastructure providers:*
 - Build infrastructure to capture real-time data from vehicles, applications, and infrastructure elements
 - Develop broad-coverage, high-performance infrastructures
- *Suppliers:*
 - Develop innovations for the automaker
 - Supply aftermarket solutions

5.3.3.1 Technology Maturity Levels and Driving Factors

Automotive manufacturers are in control of all services based on the connected vehicles marketed under their brands. Content and applications are hosted on servers operated by the automotive OEMs with access provided only via manufacturer-specific portals. All services not directly developed by the manufacturers and their partners undergo a certification and review process within the OEMs' organizations. The automotive manufacturers' main concern, system security (see Chap. 6), thus

remains under their control. Automotive OEMs, however, can hardly keep up with the rapid pace of development on the Internet, short update cycles, and the many different user profiles and applications in use.

Since 2015, a large number of applications have been available to drivers on platforms operated by service providers. They represent an intelligent way to combine different pieces of information from the Internet, thereby generating high value-add for users' cars. These applications are based on:

- *Automotive cloud service system (ACSS)*: ACSS is a service-oriented architecture (SOA) for the next generation of automotive software platform. It is a computing model for providing and sourcing IT services that are highly configurable, adaptable, and scalable over the Internet with an ongoing operating expenditure compared with traditional IT models. ACSS can facilitate new and expanded channels, as well as improve access to client data, allowing for better tailored products and services. However, automakers pioneering the design of new cloud features for cars will likely continue to address any concerns associated with data security or troublesome downtime events in an effort to make the cloud fit the needs of tomorrow's drivers.
- *High bandwidth*: At the VDI Wissensforum on automotive electronics in Baden-Baden (Germany) in 2011, several players announced their commitment to Ethernet as the future standard for in-car data, high-bandwidth communications. The Ethernet physical layer for automotive environments is the key element for a breakthrough of Ethernet in this domain. It enables OEMs and suppliers to implement Ethernet-based data bus systems without EMC problems and at very low cost, even in safety-critical automotive environments.
- *HTML5*: Is the fifth version of HTML, a computer language for marking and linking texts and other content of electronic documents, mainly on the Internet. HTML makes it possible for connected cars to be individualized with all of the component features required by the user.

The cloud, which provides data and makes it available on various devices, brings the data into vehicles easily and smoothly. Responsibility and ensuring system security within the vehicle are, however, open issues, as described in detail in Chap. 6.

The technologies involved in realizing the connected car currently exist primarily in the automotive, software, and telecommunications sectors. The biggest area where action is needed lies in the development of value creation networks and closing of competency gaps among individual players, along with development of a shared understanding of the entire value chain. The risk of accidents and related damage claims makes ensuring system security especially important.

As mentioned earlier, a major challenge, especially for automotive OEMs, are the long product development, which are diametrically opposed to the short life cycles found in the consumer goods and telecommunications industries. This means there is

great need to realize flexible architectures in connection with far-reaching standardization of interfaces for easy, smooth data integration and maintenance. These factors enable efficient use of technologies in different types of vehicles over the vehicle's entire lifecycle.

For the new in-vehicle services which are meant to be used by the driver while driving, the focus is on maximum ease and convenience in using the input and output devices. Through widespread use of smartphones, tablets, and aftermarket navigation system solutions, touch screens have become established as the preferred interface. Like voice command functions, touch screens are highly user-friendly and promise low driver distraction rates.

Besides these telematics maturity levels and driving forces in telematics concepts, the European Union (EU) has mandated the implementation of the European in-vehicle interoperable and harmonized emergency call (eCall) service initiative based on 112, the common European emergency number. Every new vehicle launched in the market must be equipped with technology enabling automatic emergency calls in the event of an accident beginning April 2018. This mandate will necessitate technology solutions, such as in the vehicle cockpit-embedded subscriber identity modules (SIMs) versus smartphones, to ensure a consistent and reliable connection between the vehicles and very fast wireless communication networks higher than 375 Mbit/s. Similar discussions are underway in the US for mandates on backup cameras and other safety-oriented features. Thus, automatic emergency calling functions are a major factor driving the connectivity of vehicles. Whether the performance of eCall systems will be sufficient for other services, e.g., to get clarity about the cause of an accident, whether the data recorded on the SIM card are reported to the police or the insurance company remains an open question, however. Furthermore, hazard warning functions and vehicle diagnosis by the OEMs are also seen as potential driving factors. Direct access to the vehicle by the OEM for service and warranty purposes is particularly important, especially in the alternative drive technologies segment. On the content side, the steadily rising volume of traffic means that mobility-related information, such as high-definition traffic real-time information (HDTRI) and floating car data telematics technology, is important too.

Development of traffic safety standards will tighten within the scope of the connected car. With regard to how binding new standards will be, expert's opinions are split, with some believing the issue will be regulated by law and a larger group considering it more likely that these features will be self-regulated (Bechmann et al. 2015).

5.3.3.2 Business Models in Connected Cars

Developing and establishing sustainable, attractive business models are based on the assumption that the vehicle user might be willing to pay for services and applications. Therefore, digital solutions play a key role with consumers. As these new offerings are introduced to the marketplace, the development of new service models and the associated internal operational transformation will be just as

important as technology investment to the success and profitability of these solutions. In this regard, connected cars are a leading area of investment for automakers. To varying degrees, connected cars offer services such as:

- Emergency
- Multimedia
- Navigation
- Security
- Diagnostics

All of these become more comprehensive and expand to include components or systems used at the user's home and/or in the user's office. Hence, automotive OEMs must consider how to transform their business to offer customers a connected transportation experience. Therefore, the major goal of connected cars is to offer exciting new digital capabilities for customers, which will change the way they will use and interact with their vehicles. Moreover, connected services, as part of connected cars, also open a myriad of opportunities to revisit revenue innovation and to enhance, extend, and redefine interaction with customers by:

- Enhancing the customer experience and increasing differentiation by offering right pricing and package combinations of connected services
- Extending the type of connected services and capabilities to generate new revenue streams beyond the traditional vehicle-to-driver relationship by enabling movement of people between vehicles and modes
- Redefining value to customers by integrating new digital characteristics, such as supporting how people should move around in increasingly crowded spaces (Gyimesi and Berman 2011)

Hence, the necessary technological investments in vehicle connectivity require that automakers must collaborate with new partners, suppliers, and customers, as well as throughout the enterprise itself in an ever more complex network. In addition, many of these partners are outside of the traditional automotive industry and include telecommunications, software, and content providers, as well as other electronics manufacturers with traditionally faster innovation cycles. Managing such complex alliances with companies that only do a small portion of their business within the automotive industry can be challenging. As services are added from various partners, automakers must efficiently and consistently engage and disengage with the new partners, as well as build an alliance. Original equipment manufacturers need to reconcile the dual—and very different—timelines of automobile development and ICT development. The ability to innovate and deploy connectivity solutions to the installed base in a shorter timeframe will be a critical operational capability and success factor for automakers.

Just like a variety of other businesses, automotive companies are determining how to stake their claim in the emerging mobility services business models. As congestion, population growth, and pollution push customers to consider the

limitations of vehicles, greater urban transportation data, smartphones, and ubiquitous telecommunications present opportunities for new, exciting digital offerings. Some of them are described more in detail in Meyer and Shaheen (2017), for the sharing economy and multimodel mobility as well as for innovative transportation technologies. Since most customers don't want to give up access to vehicles, automakers are challenged to bundle the right mix of cars and other transportation modes into compelling, integrated new offerings.

Digital solutions are needed both in the vehicle and outside of it (Gyimesi and Berman 2011). Therefore, some automakers are broadening their overall mission to go beyond the production of vehicles to become mobility providers. This change clearly has implications for the strategic motivation for in-vehicle services and the overall automakers' business model. For example, Peugeot's original strategic position was that telematics are an integral part of its core offering, so it provided these services for free for the lifetime of the vehicle. Volvo, on the other hand, views connected car services as a way to enhance and strengthen its core offering and believes that consumers should pay an upfront cost for connected car services for the first owner. These strategic decisions have influenced the types of services offered, the connectivity means employed, and the business models deployed.

Apple redefined itself when it moved from offering a simple portable media player in 2001 to providing a seamless music experience in 2003, bringing together the device with the online digital music store. This value proposition further evolved when Apple launched the iPhone® in 2007, followed by the App Store. Considering "what if" questions to contemplate the "impossible" can foster innovation on the strategic definition of the mission and the associated business models used. These strategic principles shape the creation of the value proposition, the appropriate business model, and the value chain to deliver the services and can vary across the same brand, by region or by model (GSMA 2012).

5.4 Platforms and Architectures

The connected vehicle is an advanced technology representing a traffic environment in which, potentially, every vehicle is networked with any other vehicle connected by modular, scalable, open, and secure connection platforms. This kind of platform can be cloud-based operating different telematics services. Thus, interoperability and scalability of platforms are essential for the connected vehicles paradigm and infrastructure, especially for independent data exchange by automakers, OEMs, and Tier 1 suppliers. Moreover, these platforms also process and enrich big data, turning it into valuable knowledge, an important issue with regard to V2X, combining data from vehicles, devices, and systems to enable innovative solutions that provide drivers with greater safety, service, and convenience.

Hence, the multiple processing systems used in connected vehicles serve as major drivers for the development of advanced mobility services as well as new business models. So, connected car platforms range from server-based, open, modular, secure, and highly scalable infrastructures to open, modular, secure, and highly

scalable cloud-based infrastructures on which telematics services can be based and operated. This permits fast service development and easy integration of various devices and business applications and seamless integration into existing IT systems as well. Moreover, advanced features and technologies of embedded IVI, which requires multicore architectures for in-car digital entertainment, are part of modular, scalable, open, and secure platforms. The target functions for today's IVI vary from terrestrial reception, digital reception, and compressed audio, up to hands-free voice, calling voice, and USB media playback, possibly in different user modes, such as single versus dual media sound. In the near future, new functions, such as near-field communication, wireless streaming, and more, will become important. The main challenge is that the platform must be open for future functions, which are unknown at design time.

Another challenge is to reduce the design effort by maximizing reuse of hardware and software, especially from related domains such as traditional consumer electronics (Moonen et al. 2005), which require the differences between consumer electronics and the automotive industry to overlap, taking into account the possible set of simultaneously activated functions. These functions have timing constraints, such as hard real-time (HRT), soft real-time (SRT), and best effort (BE). In the case of HRT, missing any deadline is not permitted. In the case of SRT, some deadlines may be missed, but the miss ratio should be low. BE functions, like access services, do not have deadlines. Hence, connected vehicle platforms and architectures need to be flexible and upgradable as well as operate in real time. They mainly consist of the following interlinked components and services:

- B2B interfaces to external devices
- Central platform services
- Device gateway
- Worldwide operation
- 24/7 service management

These can be realized by a modular SOA offering the following features:

- *Global*: Hosting and operation based at certified data centers for security reasons
- *Highly available*: Multi redundancy and high reliability of the connected car platform for stable and secure operation
- *Scalable*: Horizontal and vertical scalability with regard to the connected car platform's cluster capability
- *Secure*: Tier 1 connected car architecture

5.4.1 Connected Car Architecture and Challenges

Today's automotive architecture is comprised of diagnostics, infotainment, and telematics as a result of advanced communication technologies, such as Long-Term Evolution (LTE), the 3.9G wireless standard of the fourth generation. An

extension is called LTE-Advanced or 4G, which is backward compatible with LTE. With LTE 4.5, more than 375 Mbit/s allow significantly higher download speeds than older standards, depending on the reception situation. The next step will be the 5G communication network standard.

With regard to LTE communication technology standards, the future of driving through V2V and V2I (also known as V2R) and collectively referred to as V2X will allow a seamless exchange of information between them as the key to unlocking mobility.

Vehicle-to-vehicle is a mesh network in which each vehicle is a node with the ability to transmit, receive, and retransmit messages to other nodes. The resulting network is based on three sets of standards:

- *IEEE 1609*: Family of standards for WAVE, it defines the architecture and procedures of the network.
- *SAE J2735 and SAE J2945*: Define the information carried in the message packets. This data includes information from sensors on the car, such as the location, direction of travel, speed, and braking.
- *IEEE 802.11p*: Defines physical standards for automotive-related dedicated short range communication (DSRC).

Vehicle-to-infrastructure allows the vehicle to communicate with traffic lights, traffic management systems, and other stationary infrastructure components, which would also become nodes in the mesh network. This allows vehicles to receive information relating to the timing of traffic lights and roadside units (RSUs) or warn the driver of a potential hazard in a blind spot at an intersection. Vehicle-to-infrastructure will, in the future, also allow parking in free parking spaces and identify and announce free parking spots.

Therefore, V2V and V2I communications are considered to be key technology architectures for safe and intelligent mobility in the future, allowing the testing of V2X applications in real metropolitan field trials. These applications include vehicles, RSUs, and infrastructure facilities for traffic and test management. Also, several third parties are involved to provide access to additional services. As a result, such a system requires a commonly accepted architecture of the individual components and a seamless communication network for reliable and efficient information interchange. With regard to V2X, advanced use cases such as the following are feasible:

- Connected infotainment
- Real-time diagnostics
- Real-time tracking

In this regard, the software architecture of modern connected vehicles is comprised of three main components:

- *Connected car gateway (CCG)*: Entry point for a vehicle to communicate with the external environment. CCG encompasses advanced features, such as 4G

connectivity, hotspot, cloud connectivity, V2V communication, ability to control the car remotely, firmware updates over the air (OTA), remote diagnostics, predictive maintenance, eCall, and crash notification, which makes it a very complex architecture.

- *Cloud-based servers (CBS)*: A term applied to large, hosted, secure data centers, usually geographically distributed. They offer various computational services on a utility basis as a new way to host applications, as well as perform real-time analytics on data from the vehicle to generate real-time insights.
- *Applications on smartphones*: Provide an intuitive user interface that allows users to interact with the vehicle over wireless networks performing a variety of operations, ranging from getting vehicle status to controlling some of the aspects of the car, such as switching on the heating, ventilation, and air control (HVAC) or locating a car in the parking lot.

With regard to connectivity, the CCG is comprised of long-range connectivity modems, such as LTE; short-range connectivity modems, such as Wi-Fi; and positional tracking systems, such as GPS. Advanced connectivity features are required for transmission and reception of data in real time to/from the Internet. At the lower end, 2G/2.5G modems provide data rates of less than 100 kbps going to LTE, which provides data rates at a few Mbps. Modem integration can be done using either dedicated modules from vendors such as Telit or Sierra Wireless or built-in modems that come as part of a system on a chip (SoC) delivered by vendors such as Qualcomm.

eCall services provide emergency alerts to public safety access points (PSAP) so that help can be provided to victims in the shortest span of time. In most cases, it involves communicating information about the accident, including location, number of occupants, speed, direction, etc. Thus, vehicle emergency data sets (VEDS) can help the recipient of the information to assess the severity of the incident and provide the necessary help. The EU has passed a regulation that requires all passenger cars to be equipped with eCall systems by April of 2018. The eCall system automatically dials Europe's emergency number 112 in the event of a serious accident.

Global standards for eCall are emerging with 3GPP standardizing eCall requirements as part of 3GPP TS 26.267, TS 26.268. The transmission of VEDS is followed by an automatic call to the call center so that voice communication can be established with the occupants of the car.

In the US, the vision for Net GEN 9-1-1 is to enable a PSAP to automatically receive and process the VEDS. The US has not yet adopted a standard protocol for eCall, although some of the telematics service providers, such as GM's OnStar®, Ford's SYNC® 911 Assist®, Lexus Link®, etc., provide similar services and each one of them uses their own proprietary method for transmitting data to the call center.

The CCG application framework allows implementation of a software development kit (SDK) that permits third parties to develop applications that can be downloaded into the device. This will enable development of an ecosystem to provide a variety of services using the data that is available from a CCG box, such as usage-based insurance,

preventive diagnostics, and location-based services. The framework abstracts all hardware-specific intricacies from the app developer. The framework exposes APIs in Java/HTML5/JS for ease of programming (URL11 2017).

With more and more vehicles connecting to networks today, the possibility of an intruder obtaining access to internal vehicle networks and performing malicious activities are real threats (see Chap. 6). The infamous Jeep® hack occurred when someone was able to physically compromise a car as it was engineered using “old school” technology. Any failure at one single point would result in a breakdown of the cryptographic chain of trust. The connected vehicle units that are connected to the Internet need to implement multiple layers of security so that a break in one layer does not compromise the entire system. Software security issues have to be addressed at various levels right from the time the firmware is flashed in a factory, and all the way to ensuring the integrity of downloaded applications and workshops where the firmware will be flashed.

5.4.2 Connected Car Reference Platform

Qualcomm Technologies announced the Qualcomm® Connected Car Reference Platform (CCRP), accelerating the adoption of advanced and complex connectivity into the next generation of connected cars. They are keeping pace with an ever-increasing set of automotive use cases facilitated by the latest advances in 4G LTE, Wi-Fi, Bluetooth, and V2X communications. The platform is also designed to meet challenges such as wireless coexistence, future-proofing, and support for a large number of in-car hardware architectures.

The CCRP is built upon Qualcomm Technologies’ automotive product and technology portfolio, including Qualcomm Snapdragon™ X12 and X5 LTE modems, quad-constellation Global Navigation Satellite System and 2D/3D dead reckoning (DR) location solutions, Qualcomm® VIVE™ Wi-Fi technology, dedicated short-range communications (DSRC) for V2X, Bluetooth, Bluetooth Low Energy, and broadcast capabilities, such as analog and digital tuner support using software-defined radio via Qualcomm® tuneX™ chips. In addition, the platform features in-vehicle networking technologies, such as OPEN Alliance BroadR-Reach (OABR) gigabit Ethernet with Automotive Audio Bus (A²B®) and controller area network (CAN) interfaces.

Advanced features included in the CCRP design are:

- *Future-proofing*: Allowing the vehicle’s connectivity hardware and software to be upgraded through its life cycle, providing automakers with a migration path from DSRC to hybrid/cellular V2X and from 4G LTE to 5G
- *OEM and third-party applications support*: Providing a secure framework for the development and execution of custom applications
- *Scalability*: Using a common framework that scales from a basic telematics control unit (TCU) up to a highly integrated wireless gateway connecting multiple

ECUs within the car supporting critical functions, such as over-the-air software upgrades and data collection and analytics

- *Wireless coexistence:* Managing concurrent operation of multiple wireless technologies using the same spectrum frequencies, such as Wi-Fi and Bluetooth

The CCRP allows automakers and their suppliers to explore, prototype, and commercialize connectivity designs using modules and solutions offered based on Qualcomm Technologies' roadmap.

5.4.3 Connected Car in the Cloud

Cloud computing is in use by multiple industries, and the automotive one is no exception with regard to connected cars. Cloud computing, in general, includes technologies and business models to dynamically provide IT resources and to bill for their use according to payment models. Instead of businesses operating IT resources in their own enterprise data centers, they are using cloud computing which is available in the form of a service-based business model. This model is available through the Internet or an Intranet whereby the business model represents how a company can generate value for its customers and ensure a return for the company. Thus, enterprises can reduce their long-term IT capital expenditures by deploying cloud computing as their IT resource. As IT resources of various types are flexibly deployed in a service-based way, they are referred to as everything as a service (EaaS) and are divided into four classes of cloud services:

- *Business Process as a Service (BPaaS):* Allows customers to outsource all of their business processes to a cloud provider and implement them through business process technologies. Therefore, the provider offers all of the IT resources, and not the IT-based services a customer needs to support his business processes. Hence, BPaaS abstracts more from IT resources and focuses on the customer's business processes.
- *Infrastructure as a Service (IaaS):* Provided when physical or virtual servers are offered. The cloud manages the servers and ensures their connectivity.
- *Platform as a Service (PaaS):* Superior-grade functionalities are available, enabling the operation of customer-specific applications which may include, for example, entire databases, process engines, web services, and other features. In general, application functionalities which are not directly used by people but are integrated into other applications belong to this service class.
- *Software as a Service (SaaS):* Includes offering complete customizable software applications. Users access these applications through a network, sharing hardware, and platform IT resources but without noticing or interacting with each other. Many business sectors, for example, customer relationship management (CRM) or enterprise resource planning (ERP), can be supported by cloud services.

To make these cloud-based services available to automotive consumers, new vehicles will increasingly rely on innovative cloud-based technologies for requisite tasks, such as vehicle connectivity to the off-board world and Internet, two-way data, information transfer between the vehicle and the cloud, and reliable access to highly scalable data storage, processing, and analytics capabilities.

For connected cars, and thus smart mobility, cloud-based systems are ideally suited as reference architectures. In this case, the lowest level of the architecture is characterized by different sources of information, from smart devices or weather forecast stations, delivered to the cloud and processed into important information available from the cloud. This information can be delivered to user devices and used, for example, by an app or the browser of a user's smartphone. In this regard, the cloud platform provides cloud application containers and services for the development and operation of applications. This layer is the PaaS. Compared to the IaaS, the PaaS defines which platform runs on a server. Once the application is developed and tested, it has to be adapted to the different services. To do this, PaaS must provide services for the deployment of the various components.

Today, connected car and integrated cloud technology are already state of the art. Ford, for example, has announced an expansion of connected vehicle services capability with the creation of the Ford Service Delivery Network powered by the Microsoft® cloud platform which was launched in 2008 as Windows® Azure and rebranded in 2014 as Microsoft Azure, which is the major cloud platform that is consistently used for both IaaS and PaaS.

The advantage of cloud services is the scalability and accessibility to new applications, resources, and services. Microsoft categorizes Azure services into 11 main product types (URL12 2017):

- *Analytics:* Provides distributed analytics and storage, as well as real-time analytics, big data analytics, Data Lake machine learning, and data warehousing.
- *Computing:* Provides virtual machines, containers, batch processing, and remote application access.
- *Data storage:* Category includes database as service, as SQL and non-SQL (NoSQL), as well as unstructured and cached cloud storage.
- *Development:* Services help application developers share code, test applications, and track potential issues. Azure supports a range of application programming languages, including JavaScript, Python, .NET, and Node.js.
- *Hybrid integration:* Services for server backup, site recovery, and connecting private and public clouds.
- *Identity and access management (IAM):* Ensures only authorized users can employ Azure services and helps protect encryption keys and other confidential information.
- *Internet of Things:* Services help users capture, monitor, and analyze IoT (see Sect. 5.2) data from sensors and other devices.
- *Management and security:* Helps cloud administrators manage their Azure deployment, schedule and run jobs, and create automation. This product group also includes capabilities for identifying and responding to cloud security threats.

- *Media and content delivery network:* Includes on-demand streaming, encoding, and media playback and indexing.
- *Networking:* Includes virtual networks, dedicated connections, and gateways, as well as services for traffic management, load balancing, and domain name system (DNS) hosting.
- *Web and mobile:* Supports the development and deployment of web and mobile applications and also offers features for API management, notification, and reporting.

The full list of Azure services is constantly subject to change.

Most cloud-based services are available courtesy of a vehicle's connection to a smartphone, such as Apple's iPhone or the Motorola Droid. That is because those devices provide the wireless data connections required for getting information from the remote servers where it originates into a vehicle. The idea is to connect the smartphone with the user interface of the vehicle making the interaction safer. With that data stream, today's leading telematics systems from automakers and their OEM and Tier 1 suppliers provide a number of different functions to vehicle passengers.

Connected car cloud platforms are not only offered from technology giants like Microsoft. Airbiquity, for example, has a connected car cloud platform, an open platform architecture, that integrates the entire spectrum of vehicle systems, connectivity devices, communication networks, content providers, and backoffice IT systems for traditional and emerging use cases. The platform is called Choreo; it enables automotive OEMs to deploy, manage, and dynamically update innovative connected car software globally for their customers. It provides service delivery for eight automotive brands and powers over six million vehicles around the world. Choreo also enables driver safety and convenience features such as remote vehicle monitoring, geofencing, and automatic crash notifications.

As more automakers launch new cloud-based applications, resources, and services, it will become increasingly clear that cloud technology is the best way to power connected car software today and in the future (URL13 2017). That is because:

- Automakers must fully embrace the cloud technology if they want to achieve faster software deployments to deliver competitively differentiated features and services to existing and prospective customers. Cloud technology is uniquely suited to efficiently configure, scale, manage, and to update connected car software features and services dynamically. For example, Nissan saw this potential and seized the opportunity by using cloud technology to deploy their NissanConnectSM with a mobile apps infotainment program across more than 50 countries and over 20 vehicle models in just 16 months. This kind of software deployment speed was unheard of prior to the introduction of cloud-based technology and service delivery capability.

Automakers are not the only ones that directly benefit from the connected car cloud technology because consumers will have many opportunities to derive value

as well. Connected cars have the ability to provide a steady stream of valuable information about the vehicle and driver that can be used to enhance the consumer's driving experience post-purchase. By learning more about their vehicles and individual consumer driving habits and preferences, automakers can create highly personalized and relevant driving-centric services and promotions with current and new third-party partners, such as automotive dealers, online service providers, and brick and mortar retailers for oil changes, collision repair, parking, food, beverages, and other convenience items ([URL13 2017](#)).

With the emergence of connected infotainment systems, such as connected navigation, social media, music streaming, and in-car Wi-Fi, and accompanying automotive application frameworks, more advanced vehicle connectivity platforms and cloud capabilities are required. This is resulting in advanced cloud-based connected car platforms with capabilities that far exceed those of legacy telematics platforms and also require broadband cellular connectivity, initially 3G but now increasingly 4G, and with 5G-based services. Additionally, OTA is quickly becoming a key vehicle life cycle management tool as well as an enabler of analytics and big data approaches. Finally, with connected vehicles increasingly communicating, interacting, and engaging with other connected industries, such as energy, transportation, and smart home, the cloud is quickly becoming the key technology for enabling cars to connect with the wider Internet of Everything (IoE), as reported in the *2015 ABI Technology Analysis Report AN-1999* ([URL14 2017](#)). The report also discusses typical connected car cloud applications, benefits, limitations, constraints, solutions, main players, and forecasts per application and service category.

Due to the emergence of more advanced vehicle connectivity platforms and cloud capabilities, the automotive industry has to pay more attention to transformative technologies like ADAS, V2X, 5G connectivity, AI, augmented reality (AR), driverless vehicles, electrification, and IoT that will enable the mobility as a service (MaaS) paradigm and allow for new business models, such as carsharing and ridesharing. This allows new business models, such as the carsharing and ridesharing. Third-party platforms, such as Apple CarPlay® and Android Auto, increasingly dominate the industry.

The next innovative step will be car-to-cloud vehicle sensor data crowdsourcing for traffic management, automated parking, weather, and high-definition (HD) map services and cloud-to-car OTA updates for life cycle and cyber security (see Chap. [6](#)) management. Thus, commercial connected car technology will evolve from after-market fleet telematics to embedded connectivity, active safety, ADAS, platooning, and autonomous vehicles.

5.5 Autonomous Vehicles

Meanwhile, innovative companies are working hard and fast to create the technology that will enable 100% self-driving vehicles, also called autonomous vehicles or driverless driving. From a more general perspective, autonomous driving can be seen as the most advanced technological development in smart mobility.

Autonomous driving means the independent and purposeful driving of a vehicle without the intervention of a human. Technical, legal, and social aspects have to be taken in account carefully. On the way to fully autonomous driving, several steps are required, which range from accompanying vehicle functions, partially automated, highly automated, and fully automated vehicle functions, which are actively carried out by the vehicle, as published in the National Highway Traffic Safety Administration (NHTSA) guidelines for the testing and deployment of autonomous vehicles (NHTSA 2016). These levels are as follows:

- *Level 0:* The human driver does everything.
- *Level 1:* An automated system on the vehicle can sometimes assist the human driver with certain aspects of the driving task.
- *Level 2:* An automated system can actually handle some parts of the driving task, while the human continues to monitor the driving environment and performs the rest of the driving tasks.
- *Level 3:* An automated system can actually, in some cases, both perform parts of the driving task and monitor the driving environment, but the human driver must be ready to take back control when the automated system requests to do so.
- *Level 4:* An automated system can drive and monitor the traffic independently while the human does not need to take back control. However, the automated system can operate only in certain environments and under certain conditions.
- *Level 5:* An automated system on the vehicle can perform all driving tasks that a human driver can perform and under all conditions.

As can be seen from the NHTSA classification, a distinction is drawn between Levels 0–2 and 3–5 based on whether the human operator or the automated system is primarily responsible for monitoring the driving environment. Therefore, an automated vehicle which is built up on a combination of hardware and software, both remote and onboard, can be characterized as an object that performs a driving function, with or without a human actively monitoring the driving environment. Thus, in the case of partial automation, Level 2, the driver must continue to monitor the vehicle. In the highly automated vehicle, Levels 3 and 4, the vehicle is controlled for a certain period of time and under certain conditions. In fully automated operation, Level 5, the vehicle has permanent control and drives autonomously.

The first steps toward autonomous vehicles can be seen in existing vehicle safety and convenience features, such as:

- *Automatic braking:* The purpose of automatic braking is to aid in stopping vehicles more quickly with the potential for preventing a high number of fatal vehicle accidents each year.
- *Lane departure warning (LDW):* See Sects. 4.2.4 and 4.9.1, as well as Chap. 11.
- *Self-parking:* See Sect. 4.9.1 and Chap. 10.

As these features continue to evolve toward true autonomy, it is assumed that there will be no need for a steering wheel and pedals because of the intelligent algorithms embedded into autonomous vehicles which take over control of the vehicle. An autonomous vehicle will have a fully integrated sensor system capable of detecting its surroundings within a 360° angle of view. This fully integrated sensor system will have up to 12 vehicle-integrated sensors including:

- Cameras (see Sect. 4.9.2.4)
- Laser sensors (see Sect. 4.9.2.3)
- LiDAR sensors (see Sect. 4.9.2.2)
- Radar sensors (see Sect. 4.9.2.1)
- Ultrasound sensors (see Sect. 4.2.6)
- Vision sensors (see Sect. 4.9.2.5)

The sensor information detected is transmitted via data buses to the central control units, which emit commands to activate the following driving systems depending on the data position commands:

- Acceleration
- Braking
- Steering

A fully automated (autonomous) vehicle will thus be realized by integrating various kinds of technologies, including the following core technologies (Johanning and Mildner 2015):

- Active and passive safety
- Active driving systems:
 - Brakes
 - Drive system
 - Steering
- Car in the cloud:
 - Big data
 - Breakdown service
 - Vehicle maintenance/repair
 - Vehicle supervision
- Communication with mobile devices
- Data security
- Intelligent algorithms
 - Action and response logic:
 - Fueling
 - Navigation
 - Vehicle maneuver

- Recognition of special situations:
 - Accident
 - Congestion
 - Construction area
 - Detour
- Detection of traffic signs
- Navigation
- Sensors
 - Inside the vehicle:
 - Camera
 - LiDAR
 - Radar
 - Sonar
 - Outside the vehicle
- V2 remote site:
 - Vehicle-to-enterprise (V2E)
 - Vehicle-to-home (V2H)
- Vehicle-to-vehicle (V2V) communication
- Vehicle-to-infrastructure (V2I) communication
 - Near:
 - Locating equipment
 - Municipal public service and rescue services
 - Traffic light system
 - Traffic management system
 - Far:
 - Commercial traffic services
 - Traffic situation

Besides the aforementioned core technologies for autonomous driving, governments will need to be increasingly involved to set the legal requirements and framework conditions as guidelines for technology adoption and integration into existing and new smart public infrastructure, such as:

- *Smart roads:* Despite the many technological advances made to vehicles and mobile devices, little change can be seen to roads which would help to improve the driving of vehicles, particularly when it comes to road safety. Therefore, the technologies required to turn a road into a smart road can be summarized as follows:
 - *Induction priority lanes:* E-vehicle drivers can charge their car batteries on the go.
 - *Interactive lighting:* Motion sensors will light only a particular section of a road where a vehicle is approaching.
 - *Reflective dots:* Advanced through incorporation of LED lighting, sensors, microprocessors, and wireless communication capabilities, powered by built-in

- photovoltaic cells or piezoelectric panels to generate electricity when a passing vehicle drives over the marker.
- *Road safety units:* Focus solely on conducting strategic traffic enforcement to reduce serious injuries and fatal collisions on roads.
 - *Smart traffic signals:* Take into account traffic flow which results from a certain traffic rhythm. This will have a dramatic impact on the quality of urban living. It uses sensors, actuators, and communication technologies to calculate the arrival time of vehicles at road intersections with traffic lights by monitoring vehicles' actual speed. Based on this calculation, it is assumed that the vehicle will arrive at the traffic light when it has changed from its red phase to the green phase (Möller et al. 2015).
 - *Smart transit systems:* Allows access to real-time departure information from bus or metro stop locations throughout a city. By using online trip planning apps, the user can download complete schedule information for the day as well as other essential features.

In the not-too-distant future, automakers will increasingly be under pressure—and in some cases required—to comply with and participate in government-sponsored policies and infrastructure initiatives. This is good and needs to happen; but it will take an exponentially larger amount of money, technology, and time to enable the evolution from the connected car as we know it today to the autonomous vehicle of the future (Airbiquity 2016).

In this regard, connected trucks can lead the transition to digitized vehicles. One of these digitization efforts is an open software platform and a real-time network of all involved stakeholders in the supply chain, from express agents, loaders, dispatchers, and drivers to the recipients. Based on the data collected, all users can benefit, e.g., express agents, from potentially lower transport costs, e.g., by reducing the number of empty trips, usage of data of the tour, truck position, driving times, and more. However, this implies fully networked trucks which also are the basis for the introduction of platooning, where networked trucks closely follow each other with a very short distance between vehicles (10–12 m, 32–39 feet) which also means less space taken up on the road.

The technology used is based on the automotive standard for wireless connections. The vehicles can communicate within a radius of 200 m of one another, which is enough for truck platooning with no need for elaborate infrastructure. In addition to the technical development, the necessary legal framework needs to be adjusted. Currently, the legal minimum distance of trucks on the highway is 50 m (164 feet). Truck platooning can improve traffic safety. Moreover, platooning is also a cost saver as the trucks drive close together at a constant speed.

The next big step in technology is the driverless, autonomous driving truck, or the autonomous driving vehicle in which an “autopilot” controls the vehicle. But real-world traffic is complex. The automotive industry is experimenting with autonomous prototypes on the road, apparently without major problems so far. However, city traffic is much more difficult. Semiautomatic functions are now available everywhere.

There are also critical voices concerned about autonomous driving in general. Engineers and automakers agree that the transition to autonomous driving is the real problem today. This has to do with the time frame in which both autonomous vehicles and those controlled by drivers are on the road together. This transition is referred to as mixed traffic in which some vehicles can communicate with one another and some vehicles cannot. This means that the transition from nonautonomous to autonomous driving will be characterized by mixed traffic over a significant length of time. Essentially, the following crucial ethical questions have to be answered:

- Can an automaker's software engineers make life and death decisions?
- Are the automaker and its software engineers allowed to develop, implement, and execute intelligent algorithms which may have to decide whether, e.g., a car will run over a playing child or run over another human being standing right next to the child?

These are difficult ethical questions—whether or not a vehicle's computer with its intelligent algorithms should be allowed to make life or death decisions based on the technology embedded by the automaker and its programmers. In the end, the individual driver has always had to make such difficult decisions and bear the responsibility for them. The issue is how the driver can trust that the embedded software is not fraudulent and thereby hazardous. The Volkswagen diesel emissions scandal is an example of how legal requirements can be circumvented by using software algorithms to outwit legal guidelines, cheating both the buyers of the vehicles and the registration office. Who will guarantee that such a fraud will not be perpetrated in autonomous vehicles?

Despite the current debates on autonomous driving, the unanimous opinion is certainly that the driver, and in the case of autonomous driving the current user, is always the final decision maker and, therefore, responsible for the vehicle. There are some legal challenges to autonomous driving, however, as reported in the 52nd Santa Clara Law Review 1145 (Beiker 2012). The traditional approach to traffic litigation assumes the cause of an accident to be a human or technical failure, environmental conditions, or some combination thereof. Considerations become more complex in the case of an autonomous vehicle. As the vehicle navigates itself through traffic, it makes mission-critical decisions, which, in a narrow range of circumstances, can and will contribute to accidents. Such an event cannot necessarily be classified as a technical failure, however, in the same way as, for instance, a damaged tire. This presents an arguably novel situation wherein artificial intelligence acts on behalf of a human with life or death consequences. It is unclear how the courts, regulators, and public will react to accidents involving robotic cars. Overreaction is a clear danger, even if it could be shown that a transition to autonomous vehicles leads to far fewer traffic-related deaths overall. Mitigating these issues will require, at a minimum, research and education. Examples of how to prepare the courts and the public for autonomous vehicles may include:

- Extensive beta testing with limited autonomy
- Mandatory data recorders for autonomous vehicles
- Mock trials and focus groups
- Pilot fleet communities with statistical comparisons
- Special insurance policies for autonomous vehicles

5.6 GENIVI Alliance

A nonprofit industry alliance, GENIVI, is committed to adopting an IVI open source development platform, setting requirements and implementation standards, and providing certification programs. GENIVI seeks to provide entertainment and information features and functionality so that infotainment applications will be universally available.

The GENIVI Alliance was announced in 2009 at the CeBIT fair in Hannover, Germany, with eight founding members from different industries:

- *Automakers:* BMW, PSA Peugeot Citroen, General Motors
- *Tier 1 suppliers:* Delphi, Magneti-Marelli, Visteon
- *Operating system vendor:* Wind River
- *Silicon vendor:* Intel

The goal of the GENIVI Alliance was to define a common software platform, based on Linux® and open source software that implemented the non-differentiating functionality required for IVI systems.

Within the automotive industry, IVI delivery and maintenance are a challenge for many automakers. IVI features and applications commonly include:

- Connectivity and external communications
- Connectivity to mobile devices and the Internet
- Entertainment, radio, and media player
- Navigation- and location-based services

Some of these functions are unique to automotive applications, but most are strongly influenced by the nonautomotive consumer sector. Hence, adopting an open source software development model and enabling the transfer of innovation between related industries were a logical consequence. Therefore, the innovative infotainment systems, along with the development of the latest feature-rich smartphones that are able to seamlessly interact with the IVI system, are playing an increasingly large part in today's vehicle purchasing decisions.

Consequently, the GENIVI Alliance concentrates on developing and delivering the precompetitive components of the IVI stack, such as:

- Linux-based core services
- Middleware
- Open application layer interfaces

Historically, with regard to the automotive industry, automakers competed across the whole IVI stack. However, much of it is non-differentiating from a customer point of view. Therefore, the logic behind the GENIVI Alliance was to identify which areas of the IVI stack are non-differentiating and to come up with a level of standardization enabling OEMs to continue to compete in their nonautomotive businesses on a higher level in the solution stack.

During IVI product development, the OEMs or Tier 1 suppliers build the remainder of the solution on top of that non-differentiating middleware driven by GENIVI. This approach enables developers, who traditionally found it difficult to work within the closed automotive industry, to gain access and work together. Meanwhile GENIVI has over 180 automotive industry companies promoting deployment of open source software in the automotive electronics business, specifically in the infotainment business.

The GENIVI Alliance members produce and maintain code in open source development projects and, in parallel, collaborate in technical workgroups to combine technical requirements and interfaces with the goal of simplifying production of commercial implementations. Thus, GENIVI delivers a reusable, open source platform, providing the industry at large with a competitive environment for faster innovation and lower software development cost. Linux is the basis for the platform, and all software components defined and implemented by GENIVI members are hosted through the Linux Foundation in GENIVI repositories.

GENIVI members also engage directly in established open source projects to introduce the automotive perspective and needs. In excess of 150 software projects make up what is called the GENIVI baseline, to which Tier 1s and OEMs add additional open and closed source code to meet the OEMs' system requirements. Where no code exists, GENIVI will sponsor and launch a new open source project to develop the needed software. The GENIVI baselines serve as reusable platforms for organizations to use in product development and commercial activities.

Today, nine years after the alliance launch, automakers and their suppliers are reaping the benefits of GENIVI's open source approach, including adopting the open source middleware platform for IVI. As an example, Bosch Car Multimedia develops smart integration solutions for entertainment, navigation, telematics, and driver-assistance functions used in the automotive original equipment business. More specifically, Bosch's requirements for infotainment in the connected car included mandatory features of:

- Application frameworks
- Cloud access
- Fast updates of single features along with the related security and privacy requirements
- Smartphone integration

However, traditional requirements for quality, safety, cost, maintainability of variants, and time to market were still valid and desirable. Based on the above requirements, a new approach for delivering software was required to enable the connected car.

The IVI system developed by Bosch includes MirrorLink® and Apple CarPlay as well as an integrated Bluetooth hands-free kit. Audio streaming and digital audio broadcast in Europe brings music into the vehicle. Additional features include a 7-inch (0.1778 meter) touchscreen, steering wheel controls, and voice control.

5.7 Case Studies

The following section provides insight into some selected applications that automakers and telecommunication companies are working on with regard to V2X technologies, allowing vehicles to be connected, providing the capability of alerting or warning the driver of surrounding conditions or hazards, with the potential to reduce traffic jams, prevent accidents, and save lives. The case studies chosen illustrate how, in the not-too-distant future, vehicles will not only talk to us but communicate with each other and the roads.

5.7.1 BMW ConnectedDrive Store

BMW has developed and tested the ConnectedDrive Store, an application which allows users to buy all available services and vehicle IT functions for a BMW vehicle. The store uses a driver's data, contacts, and places, from settings stored by using cookies, to offer drivers and passengers the best possible services for driving with their BMW vehicles. Cookies are small files that are stored by a website on the user's computer or mobile device which contain information such as personal page settings and credentials.

BMW ConnectedDrive forms the center of an intelligent network of the vehicle, the driver, the passengers, and the outside world. Digital services, smart apps, and assistance systems ensure more comfort, more entertainment, and more security. This includes a variety of vehicle IT functions, such as access to the mobile office, access to social networks, or the feeding of routes from Google Maps from home via the Internet to the vehicle, to name a few of the many possibilities. In general, the digital services offered by the BMW ConnectedDrive Store connect the driver to everything that is important, such as:

- *Send to car:* Via smartphone apps or objectives found on the web, data are transferred seamlessly to the vehicle and the integrated navigation system. Even data in the smartphone calendar, such as stored meetings and desired arrival times, are automatically transferred to the BMW ConnectedDrive services as well as departure recommendations based on the current traffic situation.

- *Time to leave:* Via the iPhone or the Apple Watch®, BMW ConnectedDrive services inform the user about the ideal departure time to reach the next destination at the desired time. The system calculates the proposed departure time according to the whereabouts of the driver and based on real-time traffic data.
- *Last mile route:* On time and relaxed arrival at the appointment is made possible by the last mile route functions from BMW ConnectedDrive. After arriving at the destination, the driver is navigated, via iPhone or combined with Apple Watch, from the parked vehicle to the final meeting location.
- *Personal learned destinations:* The BMW ConnectedDrive service app learns from the driver's usage patterns and is able to automatically add frequently visited destinations into a personal mobility plan. Manual input is, therefore, now no longer necessary as everything can be delivered to the vehicle's integrated navigation system with one click.
- *Remote services:* With the remote functions of the BMW ConnectedDrive service app, different functions in the vehicle can be controlled from a distance, such as locking or unlocking the vehicle. This is a perfect aid in case of a key emergency. Assuming a driver has left his key on a journey abroad in the vehicle and went to a restaurant for lunch. Meanwhile the vehicle locked itself. After returning to his vehicle, the driver noticed the problem. With the help of his smartphone, he calls the BMW ConnectedDrive service in Munich and asks for help. The built-in SIM card in his BMW allows the service people to unblock the vehicle via ConnectedDrive from a distance. The driver can access the vehicle and continue the journey.
- The BMW ConnectedDrive service app also provides, through the customer portal, the ability to retrieve vehicle information, such as the fuel level and potential driving range or the status of windows and doors. In addition, the programmable climate control and charging timer ensure that the BMW is, for example, perfectly preconditioned and fully charged for each ride from Monday through Friday.

The current location of the vehicle can easily be determined on the map in the BMW ConnectedDrive service app. In addition, if the driver is close, the vehicle can also be found by remotely activating the horn or flashing the headlamps. Thus, the driver is always informed, even from afar, of the vehicle location and can see all of the information by a glance at the smartphone or the smartwatch. To use the functions, a onetime activation of remote services in the BMW ConnectedDrive service customer portal at www.bmw-connecteddrive.com is necessary.

On top, the BMW ConnectedDrive Store also offers the following digital services:

- Connected home.
- Driving profiles.
- Mobile office; see above for more details.
- Music and entertainment.

- Nice travel, arrive relaxed.
- Park info.
- Search, find, discover.
- *Highly automated driving:* BMW Active Assist represents technologies that enable partial and highly automated driving for more safety, comfort, and efficiency. This means that the vehicle completely or partially, in certain situations, takes over the driving task. Therefore, BMW Active Assist has a total of four laser scanners that measure the exact distances to other objects and detect the size and speed as well as the whole environment of the vehicle. The laser scanners are located in front, back, left, and right of the vehicle. Thus, the vehicle gets a view of which areas are passable and free from obstructions. The next major goal of BMW Active Assist is to enable highly automated driving on European motorways with all challenges, such as driving over national borders or driving through construction sites. These highway pilots are expected to be available in serial vehicles by 2020.
- *Intelligent emergency call:* In the event of an airbag deployment, the Intelligent Emergency Call service will send data on an accident and will make an automatic immediate emergency call to the BMW Call Center via the car's installed telephone unit, regardless of the driver's mobile phone. This service functions both domestically and abroad. Specially trained staff will communicate with the caller, in his/her native language, if possible, and inform rescue workers.
- *Real-time traffic information:* Provides the driver with the current traffic situation in real-time. In addition, the system calculates the delays that are to be expected and displays rerouting recommendations. Hence, the driver is always accurately informed about the traffic situation on the planned route and potential alternative routes, and can react to congestion and blocked routes to avoid them.

Other services offered by the BMW ConnectedDrive Store are (Johanning and Mildner 2015):

- *Concierge services:* The BMW ConnectedDrive service offers the service Information Call with the “i” key, for information, the emergency call as Break Down Call with a tool symbol on its button, and an Emergency Call (eCall) function which automatically calls for assistance in case of an accident by providing the accident location. In addition to the call center-related functions, the search for a point of interest (POI), which is the search for an object of interest for different reasons, is part of the concierge services. With regard to the POI, the nearest restaurant, hotel, gas station, hospital, pharmacy, etc. can be found, and route guidance to a parking space close to the location can be given.
- *Online entertainment:* This service offers online downloads of music into the vehicle which can be ordered directly from BMW cooperation partners. The downloaded music can also be stored on the vehicle's hard disk.
- *Remote services:* This includes all vehicle-related IT functions which are all operated remotely by apps.

5.7.2 Mercedes COMAND Online

Mercedes Cockpit Management and Data System (COMAND or COMAND Online) is the brand name of Mercedes' communications and navigation system, which was the first fully integrated telematics system in the automotive industry. COMAND Online offers a solution for Internet connectivity, infotainment, multimedia, voice recognition, and telephony in the vehicle. The actual version is the New Telematics Generation (NTG) 4.5 representing a comprehensive basis for the connected car. In general, Mercedes COMAND Online combines audio (music streaming from the phone, radio, CD/DVD, optional TV reception), telephone, and navigation functions. It also provides an Internet browser and various Internet services, such as ([URL15 2017](#), [URL16 2017](#)):

- *Facebook client:* Allows, for example, a quick glance at the pin wall or the news. Specific status messages can also be sent out by COMAND Online by using either preformulated text blocks or user formulated texts.
- *Google local search:* With access to Google live traffic information, Google Maps, and Google Street View™.
- *Parking information.*
- *Weather maps:* Germany and Europe.

The entire Internet connection runs via the Bluetooth-connected smartphone. Mercedes-Benz chose the option of a firmly built-in SIM card. But the user can, with the optional telephone module with Bluetooth (SAP Profile), insert a separate SIM card in the car and let it run the broadcast traffic.

Similar to other automakers, an emergency call function is part of the equipment. After triggering the airbags or belt tensioners the Mercedes emergency call center is notified by command COMAND Online mobile phone or telephone module which attempts to establish a voice connection to the occupants, and transmits the GPS location of the vehicle. The driver can also manually handle the emergency.

A USB port is fitted in the center armrest as a standard feature. With the optional special equipment, SPLITVIEW, the driver and front passenger can simultaneously use different media on the COMAND Online display. While the vehicle driver has the information on the control and display system in his/her view, the passenger can, for example, watch a movie on the same screen.

Mercedes COMAND Online also offers the following features:

- *Internet and apps:* Specific features such as:
 - *HRS hotel finder and cross-trade information:* Providing information on the current land traffic rules, maximum speeds, tolls/toll road, and more.
 - *News ticker:* Offers the usual selection by issues, such as economic or domestic topics. If the driver finds a message particularly interesting, he/she can also send a message by mail and can even have it read out loud.
 - *Stock market info*

- *Internet browser:* Web browser operates only when the vehicle is stationary for reasons of road safety, but the engine must be running. During the trip, only certain Mercedes apps, such as the weather display or the Google local search function, work. The exception: the passenger can surf the Internet through SPLITVIEW.
- *Web radio:* In order to receive radio over the Internet and become independent of the radio stations that can be received via the FM band, more music at the current location, sports, podcasts, and more can be searched while driving, as well as a search for the driver's favorite stations.
- *Navigation:* The user interface of the navigation system is functional; it reliably guides the driver. The driver can choose and switch between different map representations (2D, 3D, etc.) and specify what information to display on the screen, including the actual directions. Alternative routes can be determined at any time. Simple route information, such as the arrival time and the distance to the target, can be displayed and read aloud by voice command. For highway exits, there is a split screen wizard: left continues with the route; right, a separate window is displayed with a stylized representation of the exit. Route calculation and determination of traffic incidents are still provided on the basis of Navteq's TMC Pro information. Mercedes-Benz replaced the traffic calculation by embedding TomTom LIVE.
- *Voice control and telephony (hands-free):* With the voice control system, LINGUATRONIC, audio, telephone, and navigation functions can be controlled. On the radio or the CD, the station or the next track can be changed by voice. Targets for navigation can be entered by voice, and route information can be read out, such as travel time and arrival time. The traffic information, however, cannot be called by a voice command; it must be activated via the Mercedes COMAND Online controller.
- *Radio, CD, and audio streaming:* Audio streaming and playing songs that are on the driver's smartphone are possible via Mercedes COMAND Online. For this, the smartphone must be connected to Mercedes COMAND Online to play music again separately as a Bluetooth audio device.
- *Safety assistants:* Mercedes has always been a pioneer in passive and active safety. For example, DISTRONIC PLUS (Mercedes name for ACC), which automatically maintains a level of safety to vehicles ahead so that the vehicle brakes by itself, if necessary, and automatically follows the vehicle ahead without the driver having to touch the steering wheel.
- *Lane departure warning (LDW) and lane change assistant (LCA):* Also included.

In addition, the Mercedes Intelligent Drive system demonstrates the full range of advanced driver assistance functions and shows what is already possible in high automation, both on cross-country journeys and in city traffic; but extensive preparations are necessary before autonomous driving is available and widely deployed. This includes creating HD maps, as there is no existing material with the required precision (URL17 2017). HD mapping technology consolidates aerial

imagery, aerial LiDAR data, and mobile (driven) LiDAR data to create standardized, high-precision 3D base maps focusing specifically on self-driving vehicles with an accuracy of less than 7 cm of the absolute range. Mercedes embedded a high-precision map together with a stereo camera scanning the roads along the route and gathering image data. High-precision GPS information is combined with positional data producing a highly detailed representation of the roads, which is more like a 3D model of the world instead of a traditional map. To improve the accuracy, the route will be driven several times in order to provide the necessary depth of data, however, the data recorded will not necessarily be as up-to-date as it needs to be because the vehicle might not be driven on the same roads again for some time. The actual development concept improving the accuracy of the maps was completed with the aid of self-learning algorithms. The initial aim was to have a vehicle create its own high-precision map. Test vehicles were fitted with the necessary sensors and computer equipment. Drive by drive, they gathered the data from which an ultraprecise representation of the road and its immediate surroundings was put together ([URL15 2017](#)).

5.7.3 HERE: Digital Maps for Fully Autonomous Driving

Self-driving must be able to determine their position exactly. Thus, high-precision maps that are updated and made available via the Internet have been created to help in doing so. Hence, fully autonomous driving can be introduced as an additional assistance function that, in many situations, will bring in high-level support on trips, in case of traffic jams, and other unforeseen situations. But to some extent, it is also the culmination of all of the previous safety innovations targeted at accident-free driving, a vision still in the future.

In autonomous, driving the vehicle follows the vehicle ahead, even around gentle bends in the road, detecting speed limits along the way and making sure they are not exceeded. But the as yet unsolved problem for autonomous driving is the transition period with mixed traffic, when autonomous and nonautonomous vehicles will simultaneously share the same road infrastructure.

A few years ago, some automakers supposed that autonomous vehicles might be able to find out their position themselves using today's low-definition maps available in turn-by-turn navigation devices and apps, assuming that sensors would do the rest. They hypothesized that with clear road markings, visual sensors could keep autonomous vehicles safely within their lanes and even spot the solid or dotted lines that indicate stop signs and exits. But the problem is that autonomous vehicles need to operate safely in all environments because road markings can wear away or disappear under snow.

Modern LiDARs may not be as accurate as needed in those conditions (see Sect. 4.9.2.2). LiDARs calculate distances by illuminating a target with a laser light and measuring the time it takes for the light to bounce back to the source. Radar is the same but based on radio waves (see Sect 4.9.2.1). In autonomous vehicles, LiDAR

and radar have an effective range of around 50 m (164 ft) but that can shrink significantly in rain or when objects are obscured by vehicles ahead. Even the most advanced vehicle traveling at highway speeds can sense objects only about 1.5 s ahead.

Therefore, self-driving vehicles may use as many sensors as possible, but without an ultraprecise HD map, the vehicle does not have the ability to locate itself precisely (URL18 2016). When turning, for example, the autonomous vehicle cannot approximate the point at which the steering wheel should be turned, an area in which human drivers are experts at making adjustments. Digital driving instructions need to be ultraprecise. In addition to the map data, various sensors on the self-driving vehicle will also provide it with important information about its environment, which is essential for autonomous driving. Another example is an error of a couple of meters which could place an autonomous vehicle on the wrong side of a road. Commercial GPS systems are accurate only to around 5 m (16 ft) but can be wrong by 50 m (164 ft) in urban areas and fail completely in tunnels. However, HD maps include a so-called localization layer that works with a variety of sensors to position a vehicle within centimeters.

HERE, a company which provides mapping data and related services to individuals and companies, owned by a consortium of German automotive companies, namely, Audi, BMW, and Daimler, is experimenting with several such layers for their HD map. One involves extracting features such as bridges, road signs, and guard rails from images shot by the mapping vehicle and then comparing them to what the vehicle sees through its own cameras. Google, which has long been testing autonomous vehicles, builds its localization layer in a fashion similar to HERE. HERE is also trying out a system that uses artificial intelligence (AI) to identify features from cameras and LiDAR technology to collect billions of 3D points and model road surfaces down to the number of lanes and their width. It can now position an autonomous driving vehicle on the road to an accuracy of within 10–20 cm. HERE captures important details, such as the slope and curvature of the road, lane markings, and roadside objects, such as sign posts, including what that signage denotes. AI systems in HERE can identify road signs and traffic lights from photos. Humans then modify and optimize the results and check for errors. Assuming that current maps are not completely up-to-date, the next task will be to keep the map as accurate as possible. However, vehicle sensors must be robust enough to handle existing discrepancies. A partial solution is to use the digital traces of millions of people using smartphones and connected in-car systems for navigation.

HERE receives around two billion individual pieces of such data daily, comprised of a vehicle's location, speed, and heading, for their ongoing development of HD maps. This data also serves as the foundation for real-time data about the road environment and modeling road surfaces down to the number of lanes and their widths. It captures important details such as the slope and curvature of the road, lane markings, and roadside objects such as sign posts, including what that signage denotes, a technology patented in 1999 by HERE and called Electronic Horizon. It enables a vehicle, for example, to adjust the cruise control or to be more fuel efficient based on road attributes included in a map, such as the slope and curvature of the road, traffic signs, and lane information.

Meanwhile, HERE's Electronic Horizon is an embedded software solution that processes and displays detailed road network information from the cloud to help the vehicle's ADAS make more intelligent and informed decisions without driver involvement. The software translates map information with detailed road characteristics into actionable data for the vehicle. This information is used to provide both the vehicle and the driver with relevant predictive information that can assist driving decisions and enhance vehicle functionality, thereby extending the vehicle's awareness beyond what its onboard sensors can see. HERE's Electronic Horizon enables the map to act as an additional sensor for in-vehicle ADAS and highly automated driving solutions (URL19 2017).

These real-time updates could be sent to the vehicle in advance if it was driving along a road where, e.g., new potholes had appeared. While sensors can inform the vehicle about its immediate surroundings, advance warnings from real-time map data allow the vehicle to react to changes in the road in a timely manner.

HERE also announced a significant step forward in efforts to establish a global standard for vehicle-to-cloud data with regard to the design of a universal data format called SENSORIS. It was initiated by HERE in June 2015 when the company published the first open specification for how vehicle sensor data gathered by connected cars could be sent to the cloud for processing and analysis. HERE has now submitted the design for a universal data format (SENSORIS) to ERTICO-ITS Europe, the European public-private partnership for intelligent transport systems. ERTICO-ITS Europe has agreed to continue using SENSORIS as an Innovation Platform to evolve it into a standardized interface specification for broad use across the automotive industry. Defining a standardized interface for exchanging information between the in-vehicle sensors and a dedicated cloud, as well as between clouds, will enable broad access, delivery, and processing of vehicle sensor data, enable easy exchange of vehicle sensor data between all players, and enable enriched location-based services, which are the key to mobility services as well as automated driving. To date, 11 major automakers and supplier companies have joined the SENSORIS Innovation Platform under the coordination of ERTICO. They are AISIN AW, Robert Bosch, Continental, Daimler, Elektrobit, HARMAN, HERE, LG Electronics, NavInfo, PIONEER, and TomTom. More organizations are expected to join the platform soon.

Furthermore, HERE outlined its Open Location Platform which intends to harness real-time data generated by the onboard sensors of connected vehicles to create a live depiction of the road environment. Drivers will be able to access this high-quality view of the road through four services that provide information on traffic conditions, potential road hazards, traffic signage, and on-street parking. The goal is to ensure that drivers have more accurate and timely information with which they can make better driving decisions. HERE plans to make the services commercially available to any customers both within and outside the automotive industry by the first half of 2017. The services are:

- *HERE real-time traffic:* The next generation of HERE's live traffic service, HERE real-time traffic, provides real-time traffic information enhanced with the new

streams of data. The result is a high-quality, low latency feed showing hard braking alerts; jam tail warnings, with improved coverage and positional accuracy; and traffic flow, with more precise and granular data (also for lower-class arterial roads).

- *HERE hazard warnings:* Service that provides high-quality, near real-time information about potential hazards, including accidents and extreme weather events, such as slippery roads and reduced visibility. Because this service is fueled by real-time, rich sensor data, the validity of the hazards is of high quality and more trustworthy than competing services.
- *HERE road signs:* Service that provides near real-time traffic signage information, including permanent and temporary speed limits, which is useful for both the driver as well as for cars equipped with connected ADAS, such as adaptive cruise control.
- *HERE on-street parking:* Service that provides information to drivers showing roads where parking is or is not permitted for each side of the street, availability predictions and time-to-park estimations for each street and at a particular time of day based on historical driver data, and streets with paid, free, or lower-priced parking options.

HERE plans to license these four services to any automaker, municipality, road authority, smartphone maker, or app developer. As connectivity and vehicle sensor technologies become more pervasive across the industry, HERE also plans to enable other automakers to contribute their vehicle data.

The data HERE plans to use from Audi, BMW, and Mercedes-Benz vehicles in the provision of these services will be anonymized with no personal identifiers so as to ensure privacy of drivers (URL18 2016; URL20 2017).

5.8 Exercises

What is meant by the term *cyber-physical system*?

Describe the characteristics of a cyber-physical system.

What is meant by the term *cyber component*?

Describe the characteristics of a cyber component.

What is meant by the term *physical component*?

Describe the characteristics of a physical component.

What is meant by the term *wireless sensor network*?

Describe the characteristics of a wireless sensor network.

What is meant by the term *flat network architecture*?

Describe the characteristics of a flat network architecture.

What is meant by the term *Bellman-Ford algorithm*?

Describe the characteristics of the Bellman-Ford algorithm.

What is meant by the term *Dijkstra Algorithm*?

Describe the characteristics of the Dijkstra algorithm.

What is meant by the term *shared sensor network*?

Describe the characteristics of a shared sensor network.

What is meant by the term *human-machine interface*?

Describe the characteristics of a human-machine interface.

What is meant by the term *cyber-physical systems roadmap*?

Describe the characteristics of a cyber-physical systems roadmap.

What is meant by the term *cyber-physical systems design recommendations*?

Describe the characteristics of cyber-physical systems design recommendations.

What is meant by the term *requirements characteristics*?

Describe the characteristics of requirements characteristics.

What is meant by the term *requirements engineering*?

Describe the characteristics of requirements engineering.

What is meant by the term *interoperability requirement*?

Describe the characteristics of an interoperability requirement.

What is meant by the term *real-time requirements*?

Describe the characteristics of real-time requirements.

What is meant by the term *control system*?

Describe the characteristics of a control system.

What is meant by the term *proportional control*?

Describe the characteristics of proportional control.

What is meant by the term *integral control*?

Describe the characteristics of integral control.

What is meant by the term *derivative control*?

Describe the characteristics of derivative control.

What is meant by the term *proportional, integral, and derivative control*?

Describe the characteristics of proportional, integral, and derivative control.

What is meant by the term *vehicle tracking*?

Describe the characteristics of vehicle tracking.

What is meant by the term *fleet management*?

Describe the characteristics of fleet management.

What is meant by the term *Internet of Things*?

Describe the characteristics of the Internet of Things.

What is meant by the term *Internet Protocol version 6*?

Describe the opportunities of this manifold of addresses.

What is meant by the term *RFID*?

Describe the characteristics of RFIDs.

What is meant by the term *telematics*?

Describe the characteristics of telematics.

What is meant by the term *carsharing*?

Describe the characteristics of carsharing.

What is meant by the term *vehicle insurance*?

Describe the characteristics of vehicle insurance.

What is meant by the term *smart ticketing*?

Describe the characteristics of smart ticketing.

What is meant by the term *machine-to-machine telematics*?

Describe the characteristics of machine-to-machine telematics.

What is meant by the term *infotainment*?

Describe the characteristics of infotainment.

What is meant by the term *connected car*?

Describe the characteristics of connected cars.

What is meant by the term *Automotive Cloud Service System*?

Describe the characteristics of an Automotive Cloud Service System.

What is meant by the term *HTML5*?

Describe the characteristics of HTML5.

What is meant by the term *business models in connected cars*?

Describe the characteristics of business models in connected cars.

What is meant by the term *connected car platform*?

Describe the characteristics of connected cars platforms.

What is meant by the term *connected car architecture*?

Describe the characteristics of connected cars architectures.

What is meant by the term *connected car gateway*?

Describe the characteristics of connected cars cloud-based services.

What is meant by the term *Connected Car Reference Platform*?

Describe the characteristics of the Connected Cars Reference Platform.

What is meant by the term *OEM*?

Describe the characteristics of an OEM.

What is meant by the term *Tier 1*?

Describe the characteristics of a Tier 1.

What is meant by the term *connected car in the cloud*?

Describe the characteristics of connected cars in the cloud.

What is meant by the term *BPaaS*?

Describe the characteristics of BPaaS.

What is meant by the term *IaaS*?

Describe the characteristics of IaaS.

What is meant by the term *PaaS*?

Describe the characteristics of PaaS.

What is meant by the term *SaaS*?

Describe the characteristics of SaaS.

What is meant by the term *autonomous vehicle*?

Describe the characteristics of an autonomous vehicle.

What is meant by the term *highly automated driving*?

Describe the characteristics of highly automated driving.

What is meant by the term *eCall*?

Describe the characteristics of eCall.

What is meant by the term *intelligent drive*?

Describe the characteristics of intelligent drive.

What is meant by the term *LiDAR sensor*?

Describe the characteristics of a LiDAR sensor.

What is meant by the term *laser sensor*?

Describe the characteristics of a laser sensor.

What is meant by the term *radar sensor*?

- Describe the characteristics of radar sensor.
What is meant by the term *camera sensor*?
Describe the characteristics of Camera Sensors.
What is meant by the term *GENIVI Alliance*?
Describe the characteristics of the GENIVI Alliance.
What is meant by the term *BMW ConnectedDrive Store*?
Describe the characteristics of BMWs ConnectedDrive Store.
What is meant by the term *Mercedes COMAND Online*?
Describe the characteristics of Mercedes COMAND Online.
What is meant by the term *HERE*?
Describe the characteristics of HERE.
What is meant by the term *digital map*?
Describe the characteristics of a digital map.

References and Further Reading

- (Abelson and Sussmann 1996) Abelson, H., Sussmann, G. J.: Structure and Interpretation of Computer Programs. MIT Press, 1966
- (Airbiquity 2016) Connected Car Evolution: What's Next? Airbiquity Company Info, 2016
- (Baheti and Gill) Baheti, R., Gill, H.: Cyber-physical systems: In: The Impact of Control Technology, IEEE, pp. 161–166, 2011.
- (Baker 2015) Baker, L.: Model-Based Systems Engineering Process with Functional Model Analysis. Presentation AlaSim 2015
- (Barr 2004) Barr, D.: Supervisory Control And Data Acquisition (SCADA) Systems. NCS Technical Information Bulletin 04-1, 2004
- (Bechmann et al. 2015) Bechmann, R., Scherk, M., Heimann, R., Schäfer, R.: Trend Analysis: Connected Car 2015. MBtech Consulting GmbH, 2015. Available from: <https://www.yumpu.com/en/document/view/10955202/trend-analysis-connected-car-2015-mbtech-group>
- (Beiker 2012) Beiker, S.A.: Legal Aspects of Autonomous Driving. Santa Clara Law Review, Vol. 52, No. 4, 2012
- (Belay 2016) Belay, S.: Smart Ticketing. In: Smart Mobility, Ed. B. Flügge, Chapter 10 (in German). Springer Publ., 2016
- (Biddlecombe 2005) Biddlecombe, E.: BBC News 17.11.2005
- (Burns and Wellings 2001) Burns, A., Wellings, A.: Real-Time Systems and Programming Languages. Addison Wesley Publ. 2001
- (Chaouchi 2010) Chaouchi, H.: The Internet of Things – Connecting Objects to the Web. J. Wiley Publ. 2010
- (Chaturvedi 2010) Chaturvedi, D. K.: Modeling and Simulation of Systems Using MATLAB and Simulink. CRC Press 2010
- (Chen et al. 2011) Chen, M., Gonzalez, S., Vasilakos, A., Cao, H., Leung, V.: Body Area Networks: A Survey. ACM/Springer Mobile Networks and Applications, Vol. 16, No. 2, pp. 171–193, 2011
- (Cisco 2008) WiFi Location Based Services 4.1. Design Guide, Cisco Systems Inc. 2008, Text Part OL-1161201
- (Dargie and Poellabauer 2010) Dargie, W., Poellabauer, C.: Fundamentals of Wireless Sensor Networks: Theory and Practice, John Wiley Publ., 2010
- (Deriyenko 2012) Deriyenko, T.: RFID Application in Vehicle Tracking. Project Work in ITIS Class Internet of Things, TU Clausthal, 2012

- (Erdem et al. 2010) Erdem, E. Y., Chen, Y. M., Mohebbi, M., Darling, R. B., Bohringer, K. F., Suh, J. W., Kovacs, G. T. A.: Thermally Actuated Omnidirectional Walking Microrobot. In: Journal of Microelectromechanical Systems, Vol. 19, No. 3, pp. 433–442, 2010
- (Fitch et al. 2013) Fitch, G. M., Soccolich, S. A., Guo, F., McClafferty, J., Fang, Y., Olson, R. I., Perez, M. A., Hanowsky, R. J., Hanky, J., M., Dingus, T. A.: The Impact of Hand-Held and Hands-Free Cell Phone Use on Driving Performance and Safety-Critical Event Risk. Final Report DOT HS811-757, 2013
- (Geisberger and Broy 2012) Geisberger, E., Broy, M.: Integrated Research Agenda Cyber-Physical Systems (in German), Springer Publ. 2012
- (Geisberger and Broy 2015) Geisberger, E., Broy, M.: Living in a networked world - Integrated research agenda Cyber-Physical Systems (agenda CPS). Herbert Utz Verlag, Munich, 2015
- (Goel 2008) Goel, A.: Fleet Telematics. Springer Publ. 2008
- (Golatowski et al. 2003) Golatowski, F., Blumenthal, J., Handy, M., Haase, M., Burchardt, H., Timmermann, D.: Service-Oriented Software Architecture for Sensor Networks. In: Proc. Intern. Workshop Mobile Comp. (IMC), pp. 93–98, 2003
- (GSMA 2012) Connected Cars: Business Model Innovation. GSM Connected Living Programme: Automotive, 2012. Available from: <https://www.gsma.com/iot/wp-content/uploads/2012/05/GSMA-Connected-Cars-Business-Model-Innovation1.pdf>
- (Gulati 2015a) Gulati, A.: Telematics Redefining Automotive Industry. Auto Tech Review No.1, 2015
- (Gulati 2015b) Gulati, A.: All new cars to be internet-enabled by 2025. The Financial Express, March 2015
- (Gunes et al. 2014) Gunes, V., Peter, S., Givargisi, T., Vahid, F.: A Survey on Concepts, Applications, and Challenges in Cyber-Physical Systems. In: Transact. on Internet and Information Systems, Vol. 8, No. 12, pp. 4242–4268, 2014
- (Gupta and Chow 2010) Gupta, R. A., Chow, M.-Y.: Networked Control System: Overview and Research Trends. In: IEEE Transactions on Industrial Electronics, Vol. 57, No. 7, pp. 2527–2535, 2010.
- (Gyimesi and Berman 2011) Gyimesi, K., Berman, S.: Digital Transformation in the Automotive Industry. IBM Executive Summary Automotive: GBE03409-USEN-00, 2011
- (Jianjun et al. 2013) Jianjun, S., Xu, W., Jizhen, G., Yangzhou, C.: The Analysis of Traffic Control Cyber-Physical Systems. In: Procedia – Social and Behavioral Sciences, Vol. 96, pp. 2487–2496, 2013
- (Johanning and Mildner 2015) Johanning, V., Mildner, R.: Car IT Compact - The car of the future - networked and autonomous (in German). Springer Publ. 2015
- (Juliussen 2003) Juliussen, E.: The Future of Automotive Telematics. In: Business Briefing: Global Automotive Manufacturing and Technology, pp. 1–4. 2003. <https://people.cs.clemson.edu/~johnmc/courses/cpsc875/resources/infotainment/auto.pdf>
- (Kärkkäinen 2003) Kärkkäinen, M.: Increasing Efficiency in the Supply Chain for Short Shelf Life Goods. In: Internat. Journal of Retail and Distribution Management, Vol. 31, No. 10, pp. 529–536, 2003
- (Koutsoukos et al. 2008) Koutsoukos, X., Kottenstette, N., Hall, J., Panos, A., Sztipanovits, J.: Passivity-Based Control Design for Cyber-Physical Systems. In: Proc. International Workshop on Cyber-Physical Systems – Challenges and Applications (CPS-CA), 2008.
- (Lee 2008) Cyber Physical Systems: Design Challenges. In: Proc. 11th IEEE International Symposium on Object Oriented Real-Time Distributed Computing, pp. 363–369, 2008.
- (Lee and Seshia 2011) Lee, E. A., Seshia, S. A.: Introduction to Embedded Systems – A Cyber-Physical Systems Approach. LeeSeshia.org, 2011
- (Lewis 2004) Lewis, F. E.: Wireless Sensor Networks. In: Smart Environments: Technologies, Protocols, and Applications, pp. 1–18. Eds.: Clark, D. J., Das, S. K., John Wiley Publ. 2004
- (Li et al. 2011) Li, W., Jagtap, P., Zavala, L., Joshi, A., Finin, T.: CARE-CPS: Context-Aware Trust Evaluation for Wireless Networks in Cyber-Physical System using Policies. In: Proceed. IEEE International Symposium on Policies for Distributed Systems and Networks (POLICY), pp. 171–172, 2011
- (Liu 2000) Liu, J. W. S.: Real-Time Systems. Prentice Hall Publ. 2000

- (Mathis 2012) Mathis, R.: Neste Oil Launches Automatic Vehicle Identification at Fueling Stations, 2012. Available from: <https://www.secureidnews.com/news-item/neste-oil-launches-automatic-vehicle-identification-at-fueling-stations/>
- (Meyer and Shaheen 2017) Meyer, G., Shaheen, S. (Eds.): Disrupting Mobility - Impacts of Sharing Economy and Innovative Transportation on Cities. Springer Publ. 2017
- (Mo et al. 2014) Mo, L., Cao, X., Chen, J., Sun, Y.: Collaborative Estimation and Actuation for Wireless Sensor and Actuator Networks. In: Proc. 19th World Congress International Federation of Automatic Control, pp. 5544–5549, 2014
- (Möller et al. 2015) Möller, D. P. F., Fidencio, A. X., do Cota, E., Jehle, I. A., Vakilzadian, H.: Cyber-Physical Smart Traffic Light System. In: Proceed. IEEE/EIT Conference, pp. 546–551, 2015, DOI: <https://doi.org/10.1109/EIT.2015>
- (Möller 2016) Möller, D. P. F.: Guide to Computing Fundamentals in Cyber-Physical Systems – Concepts, Design Methods, and Applications. Springer Publ., 2016
- (Moonen et al. 2005) Moonen, A., von den Berg, R., Bekooij, M., Bhullar, H., van Meerbergen, J.: A Multi-Core Architecture for In-Car Digital Entertainment. <http://www.es.ele.tue.nl/epicurus/publications/gspx05.pdf>
- (Mulligan 2007) Mulligan, G.: The 6LoWPAN Architecture. In: EmNets Proceed. 4th Workshop on Embedded Networked Systems, pp. 78–83, ACM, 2007
- (NHTSA 2016) Federal Automated Vehicle Policy – Accelerating the Next Revolution in Roadway Safety. U.S. DoT HHTSA, 2016
- (Ning 2013) Ning, H.: Unit and Ubiquitous Internet of Things. CRC Press, 2013
- (Pala and Inanc 2007) Pala, Z., Inanc, N.: Smart Parking Applications Using RFID Technology. In: RFID Eurasia 1st Annual Conference, 2007. DOI: <https://doi.org/10.1109/RFIDEURASIA.2007.4368108>
- (PCAST 2007) PCAST: Leadership under Challenge: Information Technology R&D in a Competitive World. PCAST by Executive Order 13226, Published 2007
- (Pellizzoni 2015) Pellizzoni, R.: Cyber-Physical Systems. Available from: <https://ece.uwaterloo.ca/~rpellizz/ECE720T5-2014.php>
- (Poslad 2009) Poslad, S.: Ubiquitous Computing – Smart Devices, Environments and Interactions. John Wiley and Sons Publ., 2009
- (Radhakisan and Gill 2013) Radhakisan, B., Gill, H.: Cyber-Physical Systems. The Impact of Control Theory, IEEE, pp.161–166, 2011
- (Rajkumar et al. 2010) Rajkumar, I. L., Sha, L., Stankovic, J.: Cyber-Physical Systems: The Next Computing Revolution. In: Proc. 47th IEEE/ACM Design Automation Conf., pp. 731–736, 2010.
- (Randell et al. 1995) Randell, B., Laprie, J. C., Kopetz, H., Littlewoods, E.: Predictably Dependent Computing Systems. Springer Publ. 2007
- (Rauch et al. 2014) Rauch, S., Aeberhard, M., Ardel, M., Kämpchen, N.: Autonomous driving on the motorway – A potential study for future driver assistance systems (in German). <http://mediatum.ub.tum.de/doc/1142101/1142101.pdf>
- (Siebenpfeiffer 2014) Siebenpfeiffer, W. (Ed.): Networked Automobile - Safety, Car IT, Concepts (in German). Springer Publ. 2014
- (Sohraby et al. 2007) Sohraby, K., Minoli, D., Znati, T. F.: Wireless Sensor Networks: Technology, Protocols and Application. John Wiley Publ. 2007
- (Springer et al. 2014) Springer, T., Peter, S., Givargis, T.: Resource Synchronization in Hierarchically Scheduled Real-Time Systems using Preemptive Critical Sections. In: Proc. IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems (SEUS), 2014
- (ST Microelectronics 2013) Complete car door module – AN 2334 Application Note. ST Microelectronics, 2013
- (Vembo 2016) Vembo, D.: Connected Cars – Architecture, Challenges and Way Forward. Whitepaper Sasken Communication Technologies Pvt. Ltd. 2016. Available from:

- https://www.sasken.com/sites/default/files/files/white_paper/Sasken-Whitepaper-Connected%20Cars%20Challenges.pdf
- (Weiser 1991) Weiser M.: The Computer for the 21st Century. In: Scientific American, pp. 94–100, 1991
- (Wessel 2011) Wessel, R.: Finnish Railroad Streamlines Operations. RFID Journal online. July 14th 2011. Available from: <http://www.rfidjournal.com/articles/view?8594>
- (Xiao et al. 2008) Xiao, Z., Guan, Q., Zheng, Z.: The Research and Development of the Highways Electronic Toll Collection System. In: International Workshop on Knowledge Discovery and Data Mining, pp. 359–362, 2008
- (Yan et al. 2012) Yan, L., Chakrabarty, K., Ho, T.-Y.: A cyber-physical synthesis approach for error recovery in digital microfluidic biochips. In: Proc. IEEE Design, Automation and Test in Europe, Conference and Exhibition (DATE), 2012
- (Zaheruddin and Mandaviwalla 2005) Zaheraheruddin, A., Mandaviwalla, M.: Integrating the Supply Chain with RFID: A Technical and Business Analysis, In: Communications of the Association for Information Systems, Volume 15, 2005, pp. 393–427
- (Zhai et al. 2007) Zhai, J., Zhou, Z., Shi, Z., Shen, L.: An Integrated Information Platform for Intelligent Transportation Systems Based on Ontology. In: IFIP Vol. 254, Research and Practical Issues of Enterprise Information Systems, pp. 787–796. Eds.: Xu, I., Tjoa, A., Chaudhary, S., Springer Publ. 2007
- (Zhang and Mi 2011) Zhang, X., Mi, C.: Vehicle Power Management – Modeling, Control and Optimization. Springer Publ. 2011
- (Zhao and Guibas 2004) Zhao F., Guibas, L.: Wireless Sensor Networks: An Information Processing Approach. Morgan Kaufmann Publ. 2004
- (Zhou and Baras 2013) Zhou, Y., Baras, J. S.: CPS Modeling Integration Hub and Design Space Exploration with Application to Microrobotics. In: Control of Cyber-Physical Systems, Lecture Notes in Control and Information Sciences, Vol. 449, pp. 23–42, 2013

Links

- (URL1 2017) <http://ec.europa.eu/programmes/horizon2020/>
- (URL2 2017) <https://www.mems-exchange.org/MEMS/what-is.html>
- (URL3 2017) <http://CyberPhysicalSystems.org>
- (URL4 2017) https://de.wikipedia.org/wiki/Systems_Modeling_Language
- (URL5 2017) http://en.wikipedia.org/wiki/Requirements_engineering#cite_note-5
- (URL6 2017) <http://en.wikipedia.org/wiki/Interoperability>
- (URL7 2017) http://en.wikipedia.org/wiki/Internet_of_Things
- (URL8 2017) <https://en.wikipedia.org/wiki/Telematics>
- (URL9 2017) <https://en.wikipedia.org/wiki/Scatternet>
- (URL10 2017) <https://www.bvdw.org/themen/publikationen/detail/artikel/diskussionspapier-connected-cars-chancen-und-risiken-fuer-die-kuenftigen-anbieter-im-automobilmarkt/> (in German)
- (URL11 2017) www.sasken.com
- (URL12 2017) <http://searchcloudcomputing.techtarget.com/definition/Windows-Azure>
- (URL13 2017) <https://www.airbiquity.com/product-offerings/choreo-platform>
- (URL14 2017) <https://www.abiresearch.com/market-research/product/1022093-connected-vehicle-cloud-platforms/>
- (URL15 2017) <http://www.computerbild.de/artikel/cb-Tests-Connected-Car-Mercedes-Benz-Connectivity-Test-11358656.html>
- (URL 16 2017) http://www.mercedes-benz.de/content/germany/mpc/mpc_germany/website/de/home_mpc/passengercars/home/new_cars/models/cclass/w205/facts/interactive_manual.html

- (URL17 2017) <https://www.mercedes-benz.com/en/mercedes-benz/next/connectivity/high-precision-maps-for-self-driving-cars/>
- (URL18 2016) <https://globenewswire.com/news-release/2016/09/26/874541/10165310/en/HERE-unveils-next-generation-real-time-data-services-for-automotive-industry.html>
- (URL19 2017) <https://here.com/en/products-services/products/here-electronic-horizon>
- (URL20 2017) <https://here.com/en/innovation/sensoris>



Automotive Cybersecurity

6

Cybersecurity is the body of technologies, processes, and practices designed to protect computers, data, networks, and programs against intrusion, damage, or unauthorized access by cyberattacks. Therefore, this chapter begins, in Sect. 6.1, with an overview of automotive cybersecurity issues subdivided into ten subsections. It focuses on the scale and complexity of vehicles cyber and physical components' vulnerability to a variety of security challenges, intrusions, threats, and malicious cyberattacks whose intent is to disrupt communication, steal sensitive information or records, and impair the functioning of the system, identifying the risk level as a function of likelihood and consequences. Hence, a solid theoretical foundation for cybersecurity of vehicle cyber-physical systems is introduced too, based on concepts of artificial intelligence, deep neural networks (DNN), and deep learning (DL), control theory, epidemic theory, game theory, graph theory, and the importance of cybersecurity w.r.t. different kinds of attack scenarios, for example, the spear phishing attack. Section 6.2 introduces information technology security in automotive cyber-physical systems (CPSs) and the measures taken to ensure that automotive cyber-physical systems remain secure while interacting with other digital systems connected to a controller area network (CAN) system bus. It also describes the characteristics of today's attack taxonomies. As a logical next step, Sect. 6.3 focuses on hacking, automotive attack surfaces, and vulnerabilities and summarizes the anatomy of attack surface intrusion points in vehicles and the associated risks. Therefore, vehicle security depends on a variety of different methods and tools that systematically perform security testing, such as functional security testing, fuzzing, penetration testing, and others. Section 6.4 discusses intrusion detection, described as the detection of any set of actions that attempts to compromise the integrity, confidentiality, or availability of a system, as well as intrusion prevention, actions which attempt to prevent a detected intrusion from succeeding. Different detection methods for different kinds of intrusion types are described, including numerous static, dynamic, and hybrid methods for prevention. Section 6.5 discusses security and functional safety with regard to wireless mobile and sensor networks, platform

security, cloud computing, and data security, as well as functional safety. Section 6.6 includes several examples of car hacking. Section 6.7 contains a comprehensive set of questions on automotive cybersecurity topics, and finally followed by references and suggestions for further reading.

6.1 Introduction to Cybersecurity

The rapid growth in the development of computing technology and the Internet is having a huge impact on today's lifestyles, making day-to-day tasks easier and more convenient through wireless connection technologies. However, there is also a negative impact of this growth due to the emergence of new types of cybercrime being conducted through the use of information technology and communication (ICT). As ICT is increasingly used as a tool for committing crimes, security is a critical factor for the continued acceptance of the digital transformation and as part of the cyberspace defense against cyberattacks. Cyberattacks are facilitated by or committed using computers, networks, smart hardware devices and others, where they are agents, facilitators, or targets of the crime (Gordon and Ford 2006).

The cyber-physical systems (see Sect. 5.1 in Chap. 5, and Möller 2016) which are being used to embed the manifold of driver assistance systems, and safety and control systems into today's automobiles depend on sophisticated software to carry out specific functionalities. They develop quickly and increase in complexity, integrating communication, computing, and control into an infrastructure which plays a dual role with regard to the cyber and physical components used. Due to their scale and complexity, the cyber and physical devices of mission-critical automotive components are vulnerable to a variety of security challenges, intrusions, threats, and malicious cyberattacks. The purpose of these attacks is, for example, to:

- Compromise the functioning of the embedded cyber-physical system
- Denial of service
- Disrupt communication
- Steal sensitive information or records
- And others

Furthermore, the worldwide availability of the Internet allows cyber criminals to launch attacks worldwide on both cyber and physical system components from anywhere, at anyplace, at anytime. As a result, these cyber criminal attack-related security challenges require effective techniques for detecting, preventing, and recovering from cyberattacks. However, the main objective of automotive cybersecurity with regard to cyberattacks is to:

- Detect
- Deter
- Avert

This includes both previously known and unknown potential cyberattacks. Hence, cybersecurity is a body of knowledge about technologies, processes, and practices developed to protect networks, computers, programs, and data from cyberattacks, damage, or unauthorized access.

The traditional security approach has been to focus the most resources on the most crucial system components and to protect them against the biggest known threats. This necessitates leaving some less important systems or system components undefended and vulnerable to attack with regard to less dangerous known risks. Such an approach is insufficient when it comes to the current transformations in digitization as automakers embed automotive cyber-physical systems (CPS) to enhance and create new paradigms, such as connected cars and mobility services, which require extensive internal transformation across automakers' operations. Therefore, cybersecurity is one of the cross-cutting issues in automotive ICT because it is fundamental that authorized messages be delivered at anytime and at the right time to the right place without any disturbance or malicious attack.

Automotive cyber-physical systems (ACPSs) are engineered systems that are built from, and depend upon, the seamless integration of computational algorithms and physical objects composed of sets of wireless networked components, including sensors, actuators, control processing elements, and communication devices. Thus, using these smart and highly reliable automotive CPS, one must carefully consider the possible vulnerabilities of these systems which may result in potential security problems. In fact, concerns with the security of automotive CPS include malicious attempts through cyberattacks to:

- Intercept
- Defect
- Disrupt
- Fail

These types of attacks affect a large group of mission-critical systems or system components, which could result in the denial of available services, the theft of data, and could cause various types of damage.

Cybersecurity, from a general perspective, also deals with risk analysis, i.e., once a risk for an unauthorized intrusion has been identified, an analysis is carried out to determine the likelihood (probability) of the risk occurring and the consequence (impact) of that risk should it occur, which often is called risk quantification.

Modern vehicles can be targets of cyberattacks because of their complexity. Premium segment vehicles typically contain:

- ≥ 100 embedded electronic control units (ECUs)
- ≥ 2 miles of cable
- ≥ 100 million lines of software code
- ≥ 5 in-vehicle networks

Table 6.1 Risk level as a function of likelihood and consequences

Consequences	Likelihood		
	Highly likely	Possible	Unlikely
High	High	High	Medium
Moderate	High	Medium	Low
Low	Medium	Low	Low

This causes the ICT security requirements to dramatically increase. Therefore, the level of risk needs to be calculated as a function of likelihood and consequences. Table 6.1 illustrates the identification of a risk level with regard to the likelihood and consequences.

To define proper guidelines, automotive (vehicle) cybersecurity requires a well-defined risk analysis strategy. Automotive cybersecurity is vulnerable, and risk is an unequal vulnerability. The type and amount of risk depends on, for example, the:

- Cyberattacker's motivation
- Internal, local, and remote attacks
- Magnitude of hazards when security is compromised
- Vulnerability of system security
- And others

Vulnerabilities are weaknesses which allow a cyber attacker to reduce a system's information. With regard to vehicle cybersecurity, vulnerabilities include:

- Hazards to the lives of drivers and passengers
- Hazards to real-time operation
- Limited computational performance
- Limited vehicle external connectivity
- Unpredictable attack scenarios and threats
- Large number of components/parts from many different suppliers

The automotive industry is on the edge of a digital transformation, driven by trends such as:

- Emergence of new growth markets, such as services
- Increasing need for greater fuel economy
- New opportunities with regard to connectivity and its security
- Rapidly changing consumer behavior
- And others

In order to remain competitive and proactively address these trends, automakers embrace innovations specific to vehicle cyber-physical systems. The digital

transformation is playing a key role in taking the automotive industry into the future. Digital transformation across the automotive industry's ecosystem focuses on:

- Evolution of processes:
 - Optimal capacity planning and production
 - Reduced product development time and costs
- Evolution of products:
 - Increasing complexity and role of software
 - Move toward providing connected vehicle services
- New customer and original equipment manufacturer (OEM) relationships:
 - Better customer engagement and higher retention
 - Higher productivity through analytics and business intelligence
- New mobility solutions embedded in existing business models:
 - New service formats which focus on the holistic customer experience
 - New service and business models through cloud access
- Supply chain management:
 - Better component traceability and reduced warranty or recall costs
 - Greater supply chain visibility and reduced risks

Automakers are aware of the need to develop a new portfolio of capabilities and flexibility to generate value propositions for new customers or to transform their use models. Thus, enhancing and creating new features, such as those related to connected cars and mobility services, requires extensive digital transformation across automakers' operations. However, the ongoing trend of digitization has led to exponential growth in the volume of data generated. The real value is derived from the insights that businesses are able to draw from this data rather than from the information per se. Hence, this data is also of interest to cyber criminals. This demands an answer from automakers how to defend against the growth of intrusion points that results in manifold difficulties, such as:

- High endurance and long vehicle life cycles in which cyberattacks increase compared to the computational vehicle performance
- The difficulty of monitoring the status of automotive electronics with regard to limited vehicle external connectivity compared to traditional ICT-based systems
- Unpredictable cyberattack scenarios and threats
- Unpredictable hazards to the lives of drivers and passengers
- Difficulty of updating security software with regard to limited external connectivity of vehicles compared with traditional ICT-based systems

Therefore, with the increasing use of CPS for mission-critical operations in the automotive domain, cybersecurity issues must always be at the forefront of design. A new paradigm for automotive design and manufacturing is required, which can be stated as security by design (see, for example, German Industry 4.0 Platform (URL 2018)). Cybersecurity is a challenging, comprehensive, interdisciplinary task and a major concern in today's automotive industry because it is imperative that anomaly

and vulnerability as a consequence of cyberattacks be detected, identified, and resolved for the protection of the vehicle's mission-critical systems. The determination of the intrusion method is especially important so that the regular operation of the mission-critical vehicle system will remain undisturbed. Cybersecurity requires coordinated efforts across CPS responsible for the manifold of vehicle functionalities, with respect to:

- Application security
- Computing security
- Data security
- Intrusion security
- Network security

Nevertheless, one of the most problematic aspects of cybersecurity is the fast and constantly evolving nature of security risks because cyberattacks are becoming more sophisticated and possess the ability to spread in a matter of seconds. Therefore, it is essential to provide the necessary tools to detect, classify, and defend against the various types of cyberattacks. Cybersecurity professionals argue that the traditional approaches to securing vehicle CPS information can become unmanageable because the threat environment can become too complex.

The majority of today's anomalies and vulnerabilities in automotive electronic control systems (ECUs) are a result of their network-based accessibility, which makes them vulnerable to remote cyberattacks. Accessibility provides an entrance for launching cyberattacks on ECUs, enabling new categories of vulnerability with regard to communication network channels:

- Interception
- Replacement
- Removal of information

Hence, at the most basic level, a cyberattack requires some form of access to the targeted system, and this is normally followed by some kind of exploit. The effects of the exploit phase can include data breaches such as:

- Defective system operation
- Denial of service (DoS)
- Destruction of data systems
- Disclosure of data
- Exfiltration of data
- Information removal or corruption
- Modification of data
- Unauthorized data access
- And others

which may cause the CPS to fail in fulfilling its mission-critical operations. This type of vulnerability can be traced back to the way in which the cyber and the physical components of automotive CPS electronic control units (ECUs) are integrated. In this vulnerable space, the cyber component provides computational and control supports, facilitates the fusion and analysis of data received from various sources, and controls data for the overall operation of the respective vehicle systems.

In contrast the access phase of a cyberattack can be broken down into two forms:

- Attacks that require some kind of user action or error of emission
- Attacks that are executed automatically, without any user action required to facilitate them

Every cyberattack has a life cycle w.r.t. its impact as described in Table 6.2, which may help to understand what the cyberattacker has done, as well when and where and also create questions like “What did the cyberattacker do?,” “Is the cyber attacker still active now?,” and others.

Remote network access facilitates highly productive interaction among the various physically distributed or concurrent collaborating units of vehicle cyber-physical systems, as well as the efficient overall vehicle system management as an integral part of cyber components. This accessibility, however, also allows the easy launch of cyberattacks.

Cyberattacks not only have tremendous impact on the cyber part of a system, but they also cause the physical part of a cyber-physical system to fail because physical infrastructures are weak with regard to security. One such weakness in the infrastructure of vehicle cyber-physical systems consists of sensor nodes which make up many components, each of which is subject to physical capture. A cyberattacker can remove or destroy the sensor node creating a monitoring gap and disrupting transmission of system-critical data. Nevertheless, the major security realm of cyber-physical systems in vehicles is the cyber part.

A classification and categorization of cybersecurity risks has recently been done by (Johnson 2016), shown in Table 6.3.

Table 6.2 Generic cyberattack life cycle (Johnson 2016)

Attack phase	Description
Data exfiltration	Attacker extracts data hacked
Installation	Attacker installs malicious SW on the target system or network
Lateral movement	Attacker moves from access point in other systems or networks
Maintain persistence	Attacker may maintain a presence on compromised systems or networks or install backdoors that allow repeated access in future
Obtain credential	Attacker obtains root or administrator privileges
Penetration or access	Attacker access the target system or network
Reconnaissance	Attacker scopes the target and develops his attack plan

Table 6.3 Classification and categorization of common cybersecurity risks (Johnson 2016)

Cybersecurity risk class	Common categories
Network and web-facing app Attacks	Code Injection, cross-site scripting, man-in-the-middle attack, sniffing, WiFi penetrations;
Malware attacks	Adware, attack ware, crime-ware, spyware
Social engineering attacks	Face-to-face, pharming, phishing, social media
Hacking attacks	Access control breaches, cloud side channel attack, domain name server redirects, password hacking
Denial of service (DoS)	(D)DoS flooding, hostage taking, wipers and overwriting
Advanced persistence attacks	Botnets, cloud nets, industrial worms, malnets, rootkits

6.1.1 Cybersecurity and Vulnerability

As cyber technology evolves, the number of tools available for launching cyberattacks increases. This means that cyberattackers upped their strategies for complex attack. Therefore, a major concern when studying data processing in distributed environments, such as automotive (ECUs), deals with the problem of how to model vulnerability to an intent-based cyber criminal adversary threat. Most traditional IT solutions follow the common assumption that all components are well disciplined to follow the protocol properly, with only one exception – that an adversary may keep a record of all intermediate data processing. Such an assumption substantially may underestimate the capability of adversaries, and thus makes it difficult to defend against adversaries that are behaving arbitrarily.

Like any other new technology field, most of the effort seems to be focused on mapping solutions from existing technologies, such as sensor nodes, which share the networked operation and low capability characteristics with cyber-physical systems. Hence, a solid theoretical foundation for cybersecurity of vehicle cyber-physical systems (ECUs) can be introduced based on concepts such as:

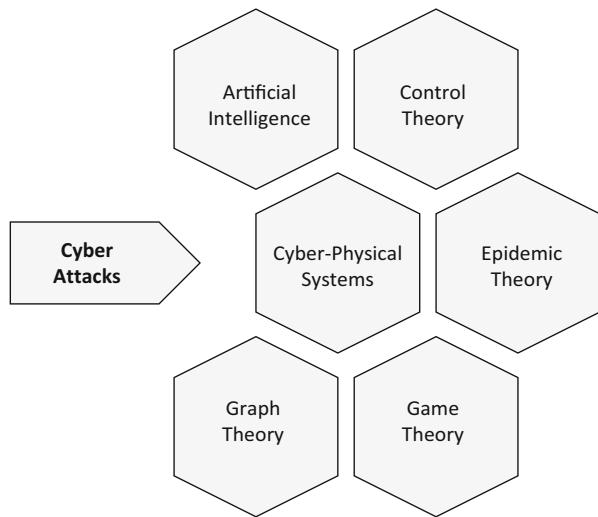
- Artificial intelligence (AI) and deep neural networks (DNN)
- Control theory
- Epidemic theory
- Game theory
- Graph theory

The aim of these concepts is to provide a holistic perspective on security, as shown in Fig. 6.1, to avoid adversary threats that consider both the cyber and the physical components.

6.1.2 Artificial Intelligence

The term artificial intelligence was coined in 1956 by John McCarthy and was defined as the science and engineering of making intelligent machines. Universal

Fig. 6.1 Holistic perspective on cybersecurity



intelligence is the study of how to make machines do things which people do better. In computer science, an ideal intelligent machine is introduced as a flexible rational agent that perceives its environment and takes actions that maximize its chance of success at an arbitrary goal. Furthermore, the term artificial intelligence is likely to be applied when a machine uses cutting-edge techniques to competently perform or mimic cognitive functions that are intuitively associated with human intelligent behavior, such as learning and problem solving. In summary, artificial intelligence can be understood as:

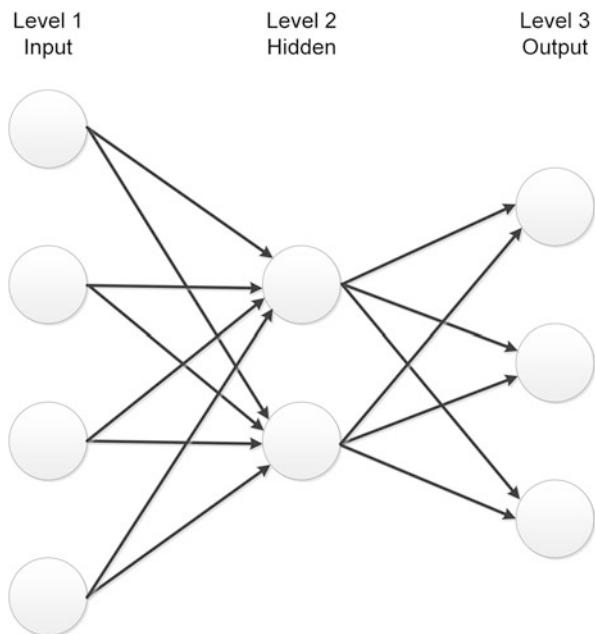
- Academic field of study on how to create machines and software that are capable of intelligent behavior
- Constituted by machines and/or software
- Study and design of intelligent agents, whereby an intelligent agent is a system that perceives its environment and takes actions that maximize its chances of success

With the pace and amount of cyberattacks, human intervention is simply not sufficient for timely cyberattack analysis and initiation of an appropriate response, especially, when the adversarial threat is carried out by intelligent agents, such as computer worms or viruses. Combatting these cyberattacks can be done with methods delivered through artificial intelligence.

6.1.2.1 Artificial Neural Networks

Artificial neural networks (ANNs) are models inspired by biological neural networks used to estimate or approximate functions depending on a large number of inputs

Fig. 6.2 Architecture of an artificial neural network with its three layers: input, hidden, output



which are interconnected with a group of nodes, as shown in Fig. 6.2, where arrows represent connections from the outputs of artificial neurons to the inputs of other ones.

Artificial neural networks are typically based on:

- *Architecture Body:* Specifies variables involved in the network and their topological relationships.
- *Activity Rules:* Represent local rules which define how the activities of the neurons change in response to each other.
- *Learning Rules:* Specify the way in which the artificial neural network's weights, $w_{i,j}$; $i, j = 1, \dots, m, n$, change with time. Usually learning rules depend on the activities of the artificial neurons. They may also depend on the target values supplied by a training phase and on the current value of the weights, $w_{i,j}$, as shown in Fig. 6.3.

From Fig. 6.2, it can be seen that ANNs are massively parallel distributed entities made up of processing units (nodes), as shown in Fig. 6.3, which have the capability for storing experimental knowledge and making it available for use in monitoring anomalous behavior in cyberspace. The nodes are also effective against hidden adversary threats. A general flowchart depicting the monitoring of anomalous behavior in ANNs is shown in Fig. 6.4.

Fig. 6.3 Node structure of an artificial neural network

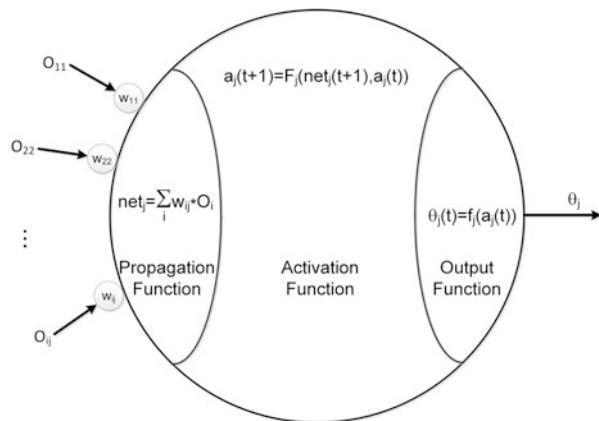
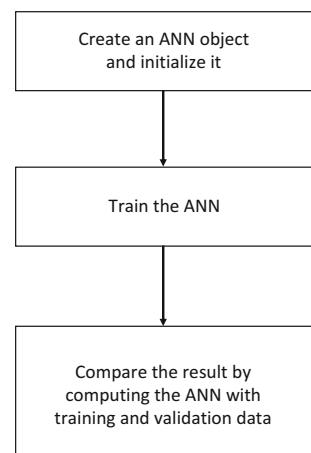


Fig. 6.4 Processing workflow of an ANN



6.1.2.2 Evolutionary Algorithms

Evolutionary algorithm represents a generic population-based metaheuristic optimization algorithm using mechanisms inspired by biological evolution, such as:

- Mutation
- Recombination
- Reproduction
- Selection

Candidate solutions to the optimization problem play the role of individuals in a population, and the fitness function determines the quality of the solution. For example, a crossover or mutation needs to be carried out with probability, p , for which a simple MATLAB program looks as follows:

Table 6.4 Common functions related to random numbers

Distribution function	C/C++	Java	MATLAB
Normal distribution $N(0,1)$	rand_max	nextGaussian	randn
Random permutation between 1 and integer n	./.	./.	randperm
Round toward infinity	ceil	ceil	ceil
Uniform distribution $U(0,1)$	(float)rand()	math.random	rand

```
%Operator M is carried out with probability p
If rand < p
    Operator M
End
```

with $rand \sim U(0,1)$ for the uniform distribution, as shown in Table 6.4. The density function of a uniform distribution random number in the range $(0,1)$ denoted as $\xi \sim U(0,1)$ is as follows:

$$p(\xi) = \begin{cases} 1 & 0 < \xi < 1 \\ 0 & \text{otherwise} \end{cases}$$

Selection solutions are randomly chosen from current solutions and determined whether one could be selected.

6.1.2.3 Fuzzy Sets

A fuzzy set is a class of objects with a continuum of grades of membership. Such a set is characterized by a membership function. Thus fuzzy sets assign to each object a grade of membership ranging between zero and one. In this regard, a set is a collection of objects that belong to some definition of a membership. Thus, a fuzzy set, A , in X is characterized by a membership function, $\mu_A(x)$, which associates, with each point in X , a real number in the interval $[0,1]$, with the values of $\mu_A(x)$ at x representing the grade of membership of x in A . Thus, the closer the value of $\mu_A(x)$ is to unity, the higher the grade of membership of x in A . The notions of complement, convexity, inclusion, intersection, relation, union, and others are extended to such sets, and various properties of these notions in the context of fuzzy sets have been established.

For example, the union of two fuzzy sets A and B with respective membership functions $\mu_A(x)$ and $\mu_B(x)$ is a fuzzy set C , written as $C = A \cup B$, whose membership function is related to those of A and B by

$$\mu_C(x) = \max(\mu_A(x), \mu_B(x)), \quad x \in X$$

It should be noted that \cup has the associative property, that is

$$A \cup (B \cup C) = (A \cup B) \cup C$$

6.1.2.4 Genetic Algorithm

A genetic algorithm (GA) is an adaptive heuristic search algorithm based on the evolutionary ideas of natural selection and genetics. Thus, it represents an intelligent exploitation of a random search used to solve optimization problems. Randomized genetic algorithms are by no means random; they exploit historical information to direct the search into the region of better performance within the search space. Genetic algorithms are based on an analogy of the genetic structure and behavior of chromosomes within a population of individuals using the following characteristics:

- Each successive generation becomes more suited to its environment.
- Individuals in a population compete for resources and mates.
- Individuals who are the most successful in each competition produce more offspring than those individuals that perform poorly.
- Genes from good individuals propagate throughout the population so that two good parents will sometimes produce offspring that are better than either parent.

After an initial population is randomly generated, the genetic algorithm evolves the three operators:

- *Selection*: Equates to survival of the fittest
- *Crossover*: Represents mating between individuals
- *Mutation*: Introduces random modifications

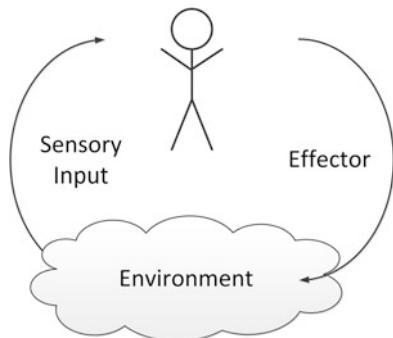
Hence, this machine learning approach imitates the process of natural selection which can be used for generating rules for classification of adversarial cyberattacks and developing specific rules for defending against specific types of cyberattacks.

6.1.2.5 Intelligent Agent

Agent theory is concerned with the question of what an agent is and the use of mathematical formalisms for representing and reasoning about the properties of agents. Agent architectures can be thought of as software engineering models of agents concerned with the problem of designing software (or hardware) that will satisfy the properties specified by agent theory. More in general, an agent can be introduced as an entity that perceives its environment through sensors and acts upon its environment through effectors, as shown in Fig. 6.5.

Thus, an intelligent agent can be viewed as an autonomous cognitive entity with standard boundaries and interfaces which understand its environment, can work by itself, and has an internal decision-making system that acts globally around other agents. Therefore, an intelligent software agent acts independently and in the interests of the user. They are used in various fields of application, for example, to control unmanned aerial vehicles, dynamic vehicle routing, route optimization in freight traffic, and others.

Fig. 6.5 Agent interacting with its environment



There are basically three different classification options for software agent types:

- *Autonomous Agent*: Is an entity that makes its own choices about how to act in its environment without any influence from a leader or global plan
- *Cooperating agent*: Involved in performing action of a plan to be executed through cooperation with the plan agent and/or other agents
- *Learning Agent*: Evaluate their actions independently in each iteration step and thus act differently in the next step

These agent attributes may occur individually or in combination. In this regard, smart agents are the highest level of intelligent agents.

In the case of a multiagent system, a group of autonomous mobile agents cooperate with each other in a coordinated and intelligent manner to plan and implement appropriate responses in case of unexpected events, such as defending against adversarial cyberattacks that an individual agent cannot solve.

6.1.2.6 Artificial Intelligence Methods

Artificial intelligence methods are helpful to detect, evaluate, and respond to cyberattacks as required for intrusion detection and prevention with regard to their specific features, as shown in Table 6.5 (Dilek et al. 2015).

An intrusion detection and prevention system is a part of software that monitors network or system activities for anomalous or malicious activities or policy violations, meaning it identifies possible adversarial intrusions and also tries to prevent them. For this reason, it contains four functionalities:

- *Analyzing*: Being able to provide efficient security against serious cyberattacks
- *Detecting Cyberattackers*: Detecting an attempt to change the system behavior which has to be realized in real time while the adversarial cyberattack is in progress (or immediately afterward)

Table 6.5 Advantages of artificial intelligence techniques suitable for intrusion detection and prevention

Technology	Feature
Evolutionary algorithm	Ant colony optimization
	Learning classifier system
Fuzzy sets	Interoperability to the environment
	Robustness of interpolative reasoning mechanics
Genetic Algorithm	Adaptability to the environment
	Flexible and robust global search
	Parallelism, allowing evaluation of multiple schemas at once
	Optimal solutions even for complex problems
	Robustness
Intelligent agent	Adaptability to the environment and user preferences
	Collaboration; awareness that human user can make mistakes, provide uncertain information, or omit important information; thus, the agent should not accept instructions without consideration and should check inconsistencies with the user
	Helpfulness; they always attempt to accomplish their tasks, having contradictory objectives
	Mobility
	Rationality in achieving their objectives
Neural net	Intuitiveness, since it mimics a biological neuron
	Intrusiveness, as they are an abstraction of a biological neural network
	Learning by example
	Nonlinearity, handling complex nonlinear functions
	Parallelism in information processing
	Resilience to incomplete data
	Versatility and flexibility with learning models

- *Monitoring in Real-Time*: Determining that a cyberattack is in progress (or immediately afterward) while minimizing false-positive alarms
- *Responding*: Reacting with regard to preventing the execution of the cyberattacker's attempt and generating reports to an a priori decided management level

The desired characteristics of a method must anticipate all possible forms of adversarial cyberattacks. The artificial intelligence intrusion detection and prevention system features are capable of detecting:

- *Buffer Overflows*: A cyberattack gaining process control or crashing another process by overflowing the other process' buffer.
- *Denial of Service (DoS)*: A cyberattack that prevents legitimate traffic or requests for network resources from being processed or responded to by the system. This cyberattack usually transmits a huge amount of data to the network. It is so busy handling the data that regular service cannot be

provided. After gaining access to the cyber-physical system, the cyberattacker can always further intrude by (Wang et al. 2010):

- Flooding a cyber-physical controller or the entire sensor network with traffic until a shutdown occurs due to the overload
- Sending invalid data to a cyber-physical controller or system network which causes abnormal termination or malicious behavior of services
- Blocking traffic, which result in a loss of access to network resources by authorized objects or entities
- *Worm Detection:* A self-replicating program propagates without using infected files. Worms usually propagate through network component services on computers or through email(s).

In the case of distributed wireless communication networks or sensor nodes, intrusion detection and prevention through intelligent agents are combined with mobile agents. This adds mobility features for monitoring suspicious cyber activities as part of an adversary's cyberattack resulting in better intrusion detection and prevention (see Sect. 6.4).

Intrusions will probably identify vulnerable weak points in cyber-physical systems which can be easily attacked. Thus, vulnerability is a vehicle cyber-physical system susceptibility or flaw. Many vulnerabilities are documented in the Common Vulnerabilities and Exposures (CVE) database supporting management of vulnerabilities discovered, thus enhancing performance with regard to the variety of functions applied to intrusion detection and prevention systems, such as:

- Classifying
- Identifying abnormal activities through statistical analysis
- Installing and operating traps to record information about intruders
- Managing audit trails and highlighting user violation of policy or normal activity
- Mitigating
- Monitoring users and system activities
- Recognizing known attack patterns in system activities
- Remediating

The CVE is maintained by the MITRE Corporation, a not-for-profit organization that operates research and development centers sponsored by the US Department of Homeland Security (DHS) Computer Emergency Readiness Team (US-CERT) Office of Cyber Security and Communications (CS&C) (URL1 2016). It lists common names for publicly known vulnerabilities. Some of these vulnerabilities are specific to a platform, operating system (OS), application, or system; but some are generic and can apply to any system. Currently, more than 50,000 vulnerabilities are identified in the CVE system. The use of CVE has been standardized by the International Telecommunication Union (ITU), the National Institute of Standards and Technology (NIST), and other standards bodies.

Some vendors provide tools that test a component or system for known vulnerabilities which vary in their approaches and coverage. For each vulnerability, the tool may implement tests that attempt to use the vulnerability as a hacker might break into a system, stop a system from functioning, or manipulate the system function in an undesirable way. Automakers may need to modify some of these tools to work in the automotive network environment to test every cyber-physical system, network infrastructure device, gateway, OS, and other integrated components. As vehicles are becoming increasingly connected, it is essential to take a holistic view of security from inside the vehicle, through the mobile vehicle networks, to the IT backend infrastructure (IXIA 2014).

A recently published review about applications of artificial intelligence techniques to combat cybercrimes (Dilek et al. 2015) gives a good overview of published research papers applying artificial intelligence techniques in intrusion detection and prevention of cyberattacks to different kinds of cyber infrastructures which are highly vulnerable to intrusion and other threats.

6.1.2.7 Deep Neural Networks and Deep Learning

A novel intrusion detection and defense system approach against cyberattacks is based on deep neural networks (DNNs) to enhance the security of vehicular networks. The DNN can be trained with probability-based feature vectors that are extracted from the improbability of each class discriminating normal and attack data, to identify malicious attacks to vehicles. This technique adapts to recent advances in deep learning, initializing the respective parameters through unsupervised pre-training of deep belief networks (DBN) improving the detection accuracy. In reality it can be very difficult to extract high-level, abstract features from raw data because many of the factors of variation may influence every observable piece of data.

When it is nearly as difficult to obtain a representation as it is to solve the original problem, representation learning does not, at first glance, seem to help. In this regard, deep learning (DL) can solve the central problem in representation learning by introducing representations that are expressed in terms of other, simpler representations. DL allows building complex concepts out of simpler concepts. The idea of learning the right representation for the data provides one perspective of DL. Another perspective on DL is that depth allows the computer to learn a multistep computer program. Each layer of the representation can be thought of as the state of the computer's memory after executing another set of instructions in parallel. Hence, DL is composed of multiple processing layers to learn representations of data with multiple layers of abstraction. It discovers intricate structures in large data sets by using the backpropagation algorithm to indicate how a system should change its internal parameters that are used to compute the representation in each layer from the representation in the previous layer. In this regard the backpropagation algorithm computes the gradient of an objective function with regard to the weights of multilayer stack architecture.

Meanwhile, DL has become more useful as the amount of available training data has increased. Thus, DL has solved increasingly complicated applications with

increasing accuracy over time, and has been successfully used in commercial applications, but was often regarded as being more an art than a technology and something that only an expert could use, until recently (Goodfellow et al. 2016). Thus, DL can scale machine learning being able to understand high-dimensional data with rich structures. Therefore, DL can take an input from a rich high-dimensional distribution and summarize it with a categorial label, for example, what CPU time is required executing an algorithm of a mission-critical system. Assuming the DL classification algorithm discards most of the input and produces a single output or a probability distribution over values of the single output, DL may be able to recognize an anomaly which shows the characteristics of a possible intrusion on the mission-critical system.

In this regard one interesting research direction is determining how distributed representations can be trained to capture the relations between entities. These relations enable to formalize facts about objects and how objects interact with each other. For example, in mathematics a binary relation is a set of ordered pairs of objects. Pairs that are in this set are said to have the relation while those who are not in the set do not (Goodfellow et al. 2016). In this regard anomaly time stamps in the execution of a mission-critical cyber-physical system identified by a DL-based intrusion detection system do not have the relation of the regular time stamps and are not in the set of regular time stamps and thus result in the identification of an intruder cyberattack situation through associative reasoning.

Associative reasoning is arguably one of the most essential intellectual capability of humans. It is the way we can reflect on ourselves. As knowledge of ourselves evolves, we find ourselves literally as spectators of our own development. The basic concept of associative reasoning is that everything is connected and networked but believing that everything is connected and networked is not so easy to understand, because everything seems to be disjointed, chaotic, and separated. The reason for this is simple, because normally humans do not know how the associations and links in their brain work. However, associating everything with anything, we can create and think on levels that we previously thought to be impossible.

John Locke describes in his essay “Concerning Human Understanding” that the task of logic is to examine the nature of the signs that the mind uses to make things intelligible to communicate them. In this regard DL and associative reasoning can be understood as the ways developing successful cybersecurity systems. For this purpose a well-defined syntax and semantic for formulating of excerpts and conclusions are required.

6.1.3 Control Theory

Cyber-physical systems (see Sect. 6.5.1) are able to connect the cyberspace and the physical space in an unprecedented manner through their increased sensing, networking, and computation capabilities. However, such connectivity options have also provided rich opportunities for adversaries to perform potential malicious cyberattacks. Therefore, control theory plays an important role in the analysis and

design of cyber-physical systems (Möller 2016) with regard to issues related to data imperfection and effects on control system performance. Data imperfection can be assumed to include:

- Delays
- Packet drops
- Quantization

These are inadequate for characterizing the possibility that transmitted data may not be true data collected by sensors or calculated by controllers because they could already have been manipulated by cyberattackers. This has raised questions relating to the secure control of cyber-physical systems. Therefore, traditional security aims to identify system anomalies and design strategies under the assumption that the system anomalies are of certain types of malicious cyberattacks; being either benign or random is not appropriate. Sophisticated cyberattackers are able to design strategies specifically to exploit vulnerabilities of the cyber-physical control system resulting in system abnormalities that are far away from random. Hence, some more formal methods can be chosen, such as:

- Control with shared processors
- Mission-critical components' privacy
- Verification and validation with timing constraints

The sensor of the control system transmits its measures at every preassigned time stamp. Then the controller calculates the control input by making use of the successfully received sensor measures. In control of cyber-physical systems, sensor data received must be consistent with the physical system behavior. If not, an adversary's cyberattack will be detected and, thereafter, removed. Therefore, the challenge for the cyberattacker is to degrade the control performance while sending data consistent with the physical part of the cyber-physical system. In contrast, the challenge for the defender is to identify if the received data is consistent with the physical part of the cyber-physical system in use.

Assume that a cyberattacker cannot be detected resulting in a trade-off between surreptitiousness and performance degradation. Therefore, the question to be answered is how to quantify surreptitiousness; or in other words, what is the performance degradation for a given level of surreptitiousness? The possible options for cyberattackers are:

- Surreptitiousness through cyber-physical system structure
- Surreptitiousness through statistical properties of the noise

Let the performance metric be the average estimation error covariance. In the absence of a cyberattack, the error covariance is $p(k)$; and in the presence of a cyberattack, the error covariance is $\bar{p}(k)$.

If the cyberattacker tries to enhance the intrusion without being detected, the following error covariance (Gupta 2016) is received

$$\bar{p}(k) = \lim_{k \rightarrow \infty} \sup \frac{1}{k+1} \sum_{n=0}^k \bar{p}(n).$$

With regard to this equation, an observer-algorithm can be embedded in the control system for evaluation of the data received in order to decide between two use cases:

UC_0 : No cyberattack detected

UC_1 : Cyberattack detected

Surreptitiousness can be measured by

$$p(\text{Decide } UC_i | UC_i) \rightarrow 0$$

and the probability of a false alarm can be described by

$$p(\text{Decide } UC_i | UC_0)$$

A cyberattack is then called surreptitious if no intrusion detection with property exists

$$p(\text{Decide } UC_i | UC_0) < p(\text{Decide } UC_i | UC_1)$$

A cyberattack is called ε -surreptitious for any $0 < \delta < 0.5$ if no intrusion detection exists such that (Gupta 2016)

$$\begin{aligned} p(\text{Decide } H_i | H_1) &> 1 - \delta \\ p(\text{Decide } H_i | H_0) &\leq O(e^{-k \times \varepsilon}). \end{aligned}$$

Thus, for a given probability, p , of a missed detection, the probability of a false alarm cannot decay faster than exponentially with the rate $k \times \varepsilon$ as the number of measurements, k , increases.

6.1.4 Epidemic Theory

Modeling epidemic diseases can be done with regard to their basic principles:

- *Basic Reproduction Rate (R_0)*: Measures the transmission potential of a disease by counting the number of secondary cases following the introduction of an

infection into a totally susceptible population. The basic reproductive rate is affected by several factors:

- Duration of infectiousness
- Probability of infection being transmitted during contact
- Rate of contacts in the host population

For an epidemic to occur in a susceptible population, R_0 must be >1 ; i.e., the number of cases is increasing.

- *Effective Reproductive Rate*: Estimates the average number of secondary cases per infectious cases in a population made up of both susceptible and non-susceptible hosts. Introduced as the number of secondary infections generated by a typical infective rate reduced by the fraction of the host population that is susceptible.
- *Herd Immunity*: Occurs when a significant portion of the population has been vaccinated, which provides protection for unprotected individuals. The herd immunity threshold is the portion of a population that needs to be immune for an infectious disease to become stable in that community. If this is reached, then each case leads to a single new case; and the infection will become stable within the population.
- *Epidemic*: An increase in the frequency of occurrence of a disease in a population above its baseline or an expected level in a given period is a mathematical approach which follows three main goals:
 - *Determine* mechanisms to control and stop epidemic and study their influence on the process.
 - *Predict* the course of an epidemic in the future, which includes, among others, the final size of the epidemic and the convergence time to the steady state.
 - *Understand* the mechanisms for spreading the epidemic and how different parameters influence its course.

Hence, an epidemic model consists of a set of assumptions about the nature of the population of interest and the spreading mechanism. Assumptions with regard to the population of interest usually belong to the following categories introduced by (Daley and Gani 1999):

- *General Structure of the Population*: Population can be homogeneous such that every individual reacts to infection and spreads infection in the same manner. There can be several different:
 - Homogeneous populations
 - Stratas interacting
 - Completely heterogeneous populations
- *Population Dynamics*: Set of individuals can be closed or open. In a closed set, the number of individuals does not change over time so there are no new:
 - Births
 - Deaths
 - Emigrations
 - Immigrations

- *Disease Status of an Individual:* Individual can be:
 - A carrier without symptoms
 - Incubating
 - Infectious
 - Immune
 - Removed
 - Susceptible to infection

In 1927, Kermack and McKendrick (1927) established a deterministic epidemic model with a fixed population of N individuals and three important states: susceptible-infected-recovered (SIR). The results constitute a benchmark for a range of epidemic models. Thus, their main result treats the epidemic threshold as an important value to separate epidemics from small infections. The deterministic SIR model, with x denoting the fraction of susceptible, y the fraction of infected, and z the fraction of recovered, results in the following equations introduced by:

$$\frac{dx}{dt} = -\beta \cdot x \cdot y; \quad \frac{dy}{dt} = \beta \cdot x \cdot y - \gamma \cdot y; \quad \frac{dz}{dt} = \gamma \cdot y; \quad \frac{1}{x} \frac{dx}{dt} = -\frac{\beta}{\gamma} \frac{dz}{dt}$$

where β denotes the pairwise rate of infection and γ is the removal rate. For this system of equations, different cases can be considered:

- *Survival and Total Size:* Assuming the infection stops spreading, the fraction of susceptible that was never infected is x_∞ , the fraction of individuals ultimately removed is $z_\infty = x_0 + y_0 - x_\infty$, and z_∞ is a unique root of the equation:

$$N - z_\infty = x_0 + y_0 - z_\infty = x_0 e^{-z_\infty \frac{\beta}{\gamma}}$$

where x_0, y_0 are initial fractions of susceptible and infected nodes.

- *Threshold Theorem:* A major outbreak occurs if, and only if:

$$\left. \frac{dy}{dt} \right|_{t=0} > 0$$

which is equivalent to $x_0 > \gamma \cdot \beta$.

- *Second Threshold Theorem:* If x_0 exceeds $\gamma \cdot \beta$ by a small value, then the final fraction of susceptible left in the population is approximately

$$x_\infty = \frac{\gamma}{\beta} - \rho$$

and $z_\infty \approx 2\rho$.

Whether a major outbreak occurs depends on the initial condition, like the fraction of susceptibles, at the start of the epidemic. Dependency of the spread on

the initial condition is a specific feature of the SIR model; in susceptible-infected (SI) and susceptible-infected-susceptible (SIS) models, the steady state does not depend on initial conditions.

In computer networks, epidemic modeling is mostly applied in the following areas:

- Epidemic algorithms and information dissemination in distributed networks (Chakrabarti et al. 2007; Eugster et al. 2004)
- Modeling computer virus and worm propagation (Kephart and White 1993)
- Propagation of faults and failures

Today, viruses and worms use different methods for spreading and different security vulnerabilities. Computer viruses are defined as small programs that can reproduce and copy themselves on other systems or on other files. The worm does not need user intervention to spread out. Most worms do not destruct the infected host computer, but some of them do. The destructive worm propagation model is derived based on a worm that writes data at a random point of a hard disc after, e.g., every 10,000 scans, until the infected computer crashes. Scanning worms are one of the most prosperous types of malware. They spread out quickly and automatically. However, they are also easy to detect and stop, leaving the Internet to stealthier types of malware. New worm types use social networks to spread. With the introduction of new web applications for the exchange of information and data, the number of cybersecurity incidents has increased.

Epidemic algorithms for information dissemination are also referred to as gossip dissemination, a computer-to-computer communication protocol. These epidemic algorithms are simple and easy to deploy, and mathematical tools allow the system behavior to be predicted. Usually, the information is either spread out forever, modeled by SI; or each node spreads the information for some time, and then it stops, following the SIR model (Eugster et al. 2004). Unreliable networks which use gossip algorithms can be modeled with an SIS model.

The epidemic dynamic model for disease propagation can be used for characterizing worm propagation, assuming that each computer is in one of the following states:

- Immune
- Infected
- Vulnerable

An immune computer cannot be infected by a worm. A vulnerable computer becomes an infected computer after being infected by a worm. The spreading mechanism – the cyber intrusion attack – determines exactly how the infection is transmitted.

6.1.5 Game Theory

Game theory is a mathematical method for studying decision-making scenarios with the interaction of at least two or more players. Such an interaction scenario includes:

- Participants
- Sets of possible utility payoffs which are called a game
- Sets of rational actions that each participant can take

In a real game, each player strives to pursue the best possible objectives by choosing courses of rational actions based on knowledge or expectations or another player's action. In game theory, game-theoretic models are studied which are abstractions facilitating the understanding of various classes of real-life situations, the so-called model of intent, which can be a utility function.

Definition 6.1

Given any pair of actions, i and j , in a set of possible actions, A , $u(i)$ and $u(j)$ refer to utility functions of i and j , which can adhere to $u(i) > u(j)$ if, and only if, the decision-maker prefers i over j .

The utility function is used to express the ordinality but not the quantity of preferences. Therefore, the player cannot know how much the decision-maker prefers i to j . Based on this characteristic, a decision maker's preferences could be represented by multiple different utility functions.

With regard to cybersecurity, one can postulate a system which incorporates the defender, D , and the attacker, A . In a case where a cyberattack is launched by multiple attackers, one has to write A_1, \dots, A_m . Cyberattackers can be classified from a more general perspective as smart insiders and naive attackers. Insider threats occur when individuals within an organization misuse their privileged access to cause a negative impact on the attacked system in terms of (Nurse et al. 2014):

- Availability
- Confidentiality
- Integrity

Therefore, insider threats are an ever-growing problem in today's world of the Internet of Things (IoT), where everything is a device that may be used to access, store, and share sensitive data. The in-depth knowledge insiders possess of the security practices and monitoring policies place organizations in dire situations if these cyberattacks are executed. Thus, identifying insiders is a significant challenge and part of international research work with regard to:

- Anomaly detection of suspicious and malicious insider activity
- Identification of behavioral factors
- Recognition of signatures in cyberattacks

But smart cyber criminal insiders are afraid of being detected and, therefore, try to make optimal attacking decisions. Thus, their strategy may vary, for example, by choosing a mixed strategy which randomly chooses between two choices according

to a probability distribution, which results in a utility function as introduced by Jin et al. (2012) as:

$$u_A = \begin{cases} 1, & \text{if attacker launches an undetected cyber attack;} \\ -\beta_A, & \text{if attacker launches a detected cyber attack;} \\ 0, & \text{if attacker abstains;} \end{cases}$$

where u_A is the cyberattacker utility function and β_A is a predetermined insider preference parameter. Since the insider is afraid of being detected, one can assume $\beta_A > 0$.

Naive cyberattackers may bring blindly significant damage to a system by launching a cyberattack without fear of being detected. In this case, the naive cyberattacker realizes that the defender is weak, he can start attacking the system, and he will always succeed. If the naive cyberattacker is technically more sophisticated, anomaly detection has to be chosen for intrusion detection. Thus, the defender, D , not only detects the incoming adversary's threats using the anomaly detection technique but makes a proper trade-off between the detection rate and the false-positive rate.

Let $\gamma \in [0, 1]$ be a trade-off parameter such that the higher the value of γ , the smaller the false-positive rate and, hence, the smaller the detection rate. Normalizing γ such that the probability for detection of an adversary's attack is $(1 - \gamma)$ means that all cyber criminal attacks will be detected when $\gamma = 0$; however, a large number of false positives will be issued. When $\gamma = 1$, no cyberattack will be detected; and no false positive will be generated. Thus, the defender, D , has two objectives: (1) to detect as many attacks as possible and (2) to reduce the number of false positives. For each cyberattacker A_i ($1 \leq i \leq m$), the loss of a defender, D , due to a cyberattack from A_i be $I_A(i) \in [0, 1]$ which results in the loss of the defender associated with A_i and can be written as (Jin et al. 2012):

$$I_A(i) = \begin{cases} 1, & \text{if } A_i \text{ launches an attack that is undetected;} \\ b, & \text{if } A_i \text{ launches an attack that is detected;} \\ 0, & \text{if } A_i \text{ abstains;} \end{cases}$$

where $I_A(i)$ is the loss of D due to a cyberattack from A_i and b refers to a detected cyberattack, whereby $b \geq 0$ captures the potential cost for the defender to repair damages caused by the detected cyberattack. In case $b \geq 1$, an undetected adversary's cyberattack leads to even greater damage; elsewhere the defender could simply abandon any detection effort (Jin et al. 2012). According to the definition of the trade-off parameter, γ , if A_i chooses to attack, then the expected loss of the defendant object is $E[I_A(i)] = \gamma + (1 - \gamma) b$ if A_i chooses to abstain, $I_A(i) = 0$.

Besides the intent-based view on smart and naive cyberattackers and defenders, the taxonomy of games shows that game theory generally can be divided into two classes:

- *Cooperative Games*: Two players can bond together depending on specific promises or relationships between them.
- *Noncooperative Games*: Players are only allowed to make decisions independently based on two kinds of models:
 - *Strategic Games*: Implying strategic interdependence of players in a decision-making environment whereby each decision of a player is affected by one or all of the other players. These models consist of a strategic set of players, the possible actions of each player, and preferences, such as payoff functions reflecting the probabilities of winning for each player.
 - *Extensive Games*: Specifies a more inclusive form, called game tree, to explicitly depict the order of play and choices that players make at each node.

Furthermore, interactions between players represented by cyberattackers and defenders can be modeled as a noncooperative, non-zero-sum dynamic game with incomplete information, which considers the uncertainty and the special properties of multistage attacks. The model for this scenario is an approach along a special game tree where the adversary is the leader and the defender is the follower. Hence, multiobjective optimization methods are used to predict the adversary's best actions at each decision node. The defender also keeps tracking the adversary's actions, updates his knowledge of the adversary's behavior after each detected cyberattack, and uses his knowledge to update the prediction of the adversary's future actions (Luo et al. 2010).

Assumptions about perfect information do not hold true in real life and have to be expanded for a stochastic game model so that it is able to capture more realistic scenarios, as the player knows the system's true state at a particular moment in time with some probability of error, i.e., at any given point in time, the true state and a player's perception can potentially be different.

Assuming a constraint of imperfect information, the best strategy for a player considering other players' choice of strategies can be computed assuming the defender can compute his best strategy for reaching the Nash equilibrium of a stochastic game for which it is assumed that the defender's sensor is imperfect. For Nash equilibrium, no player can improve his payoff by unilaterally switching to a different strategy. It is implicit that the defender knows that the error probability of his sensor and the players' objectives are directly opposite, i.e., it is a zero-sum game (Shiva et al. 2010) indicating the existence of the equilibrium.

Definition 6.2

The Nash equilibrium represents an action profile for all players in a game with the property that no single player, I , can obtain a higher payoff by choosing a different action from a_i given every other player, j , adheres to a_j .

In (Sastry et al. (1994), a decentralized learning of the Nash equilibrium multiperson stochastic game with incomplete information is introduced, where after each play, the payoffs to individual players are random variables. Nothing is known regarding the distribution of the random payoffs. For learning optimal

strategies, the game is played repeatedly. The primary interest lies in (asymptotically) learning equilibrium strategies, in the sense of Nash equilibrium, with regard to the expected value of the payoff. For the decentralized learning algorithms developed after each play, each of the players updates his strategy based solely on his current action or move and his payoff. None of the players has any information regarding the existence of other players. Thus the game is played with imperfect information.

6.1.6 Graph Theory

Graph theory was introduced very early by Leonhard Euler (1707–1783) when he was asked to find a path that crosses over each of the seven bridges in Königsberg exactly once. Today, graph theory is used for finding communities in networks to detect hierarchies of substructures. In general, graph theory is a mathematical notation used to model pairwise relations between objects. In this context, a graph, $G = (V, E)$, is a pair of vertices (or nodes), V , and a set of edges, E , assumed to be finite, i.e., $|V| = m$ and $|E| = n$. Assuming $V(G) = \{v_1; v_2; \dots; v_m\}$ with, e.g., $m = 5$, and $E(G) = \{e_1; e_2; \dots; e_n\}$ with, e.g., $n = 6$, the corresponding graph is shown in Fig. 6.6.

From Fig. 6.6, it can be seen that graphs are used for designing topological properties for complex networks, e.g., to shape or optimize a network's dynamic performance measures, whereby nodes represent program statements and directed edges represent control or data dependencies between the nodes.

Studying complex network design is germane to cybersecurity with regard to theoretically controlling fraud detection and network intrusion detection. Both require methods for calculating the regularity of a graph to detect behavior anomalies which indicate intrusion detection. Intrusion detection systems have been widely used to detect malicious behavior in network communications and hosts. Thus, intrusion detection and its management are an important capability for distributed intrusion detection solutions, making it possible to integrate and deal with different types of data or collect and synthesize alerts generated from multiple hosts located within the distributed network system environment. Hence, defending complex networks against intrusions is very difficult because a defender must be able to locate the paths into the network and prevent adversaries from using them, while

Fig. 6.6 Simple graph

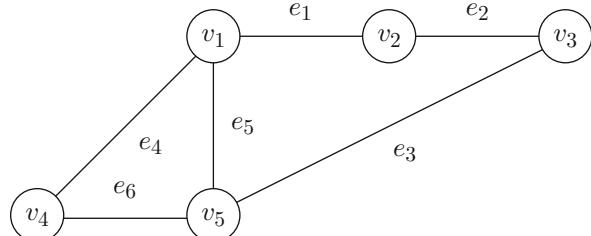
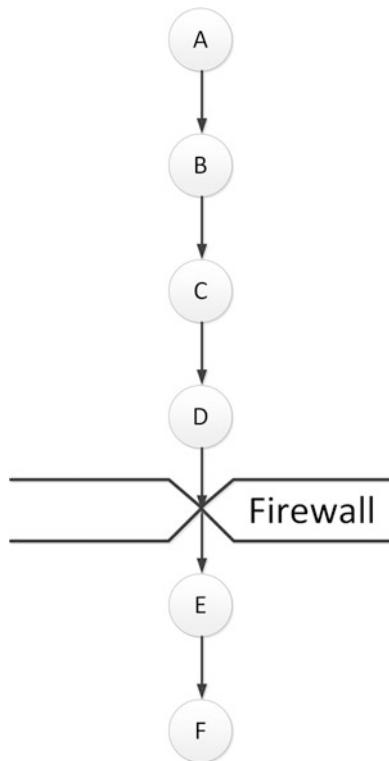


Fig. 6.7 Simple example of a network with a firewall



adversaries need to find only one unprotected path. Therefore, attack graphs are a valuable method for network defenders, illustrating paths an adversary can use to gain access to a targeted network. Defenders can then focus their efforts on patching the vulnerabilities and configuration errors that allow adversaries the greatest amount of access and work to secure those vulnerabilities.

Let's assume a simple network with six nodes, which correspond to states, and edges, which correspond to vulnerability instances. The nodes are class-divided by a firewall, as shown in Fig. 6.7.

The adversary may intrude undetected from Node *A* and can directly compromise Nodes *B*, *C*, and *D*. Assuming the adversary cannot traverse the firewall and compromise Nodes *E* and *F*, thereby completing the process by embedding malware on Node *F*, the attack graph has avoided infection of the mission-critical Node *F*. This is shown in the reachability matrix in Fig. 6.8, for the simple network depicted in Fig. 6.7, where a row represents a source interface on a node, a column represents a target port on a destination interface, and each cell indicates whether or not the source can reach the target.

Fig. 6.8 Reachability matrix;
for details see text

	B	C	D	E	F
A	x	x	x		
B	x	x	x		
C	x	x	x		
D	x	x	x		
E				x	x
F				x	x

Thus, the attack graph workflow, in general, consists of the following parts:

- Correlation quality
- Cyber attack graph construction and its visualization to improve alerts
- Information gathering

The example in Fig. 6.7 uses automatic vulnerability extraction and creation of an attack graph based on unified data models. They identify typical structures of deliberately designed networks, allowing security/vulnerability analyses to be customized specifically for these networks. Thus, it can be determined whether deliberately designed networks have favorable or unfavorable security/vulnerability properties; and response strategies based on these characteristics can be developed. With some enhancement, the deliberate design methods can be used by network engineers to allocate available resources to improve security or reduce vulnerability. But the design methods must be extended in such a way that the performance metric for design includes security and/or vulnerability measures, in addition to other metrics for the network's dynamical performance. Hence, vulnerability and system information in attack graphs can be used to prioritize and tag incoming intrusion detection alerts. Therefore, the attack graph is used during the correlation process to select and optimize correlation results to protect critical resources in networked environments which can be achieved by quantifying the likelihood of potential cyberattacks.

With regard to the problem of probabilistic incorrect computing caused by shared dependencies in nodes, the methodology for security risk analysis based on attack graph nodes and a common vulnerability scoring system allow quick calculation of the probability of cyberattacks. In this context, the method used is the dependency graph for sets of events because dependence is a common feature of a relationship between objects. The relationship between objects can be modeled as a graph, with nodes and edges corresponding to objects and links, respectively. Thus, dependence is measured based on the description of this graph which means that the dependence degree of *Object A* on *Object B* is the probability p of *B* determining *A*. Suppose the dependence degree of *Object A* on *Object B* is $Dep(A \leftarrow B)$, which describes the degree of the determinant. In the case of a cyberattack on the directed edge from *A* to *B*, cyberattacker *A* introduces malware to *B*, so *A* is dependent on *B*.

With the dependency graph method, it is possible to measure the dependence of A on B by computing the sum of the dependent values on each path which results in the computation of the dependence degree $Dep(A \leftarrow B)$.

6.1.7 Importance of Cybersecurity

As the world becomes increasingly more interconnected through digital transformation, users must pay more attention to the security of their digital connections, since the past decade has witnessed a remarkable increase in the use of digital technologies. However, the newest wave of digital technologies is different. This has been accompanied by the fast, constantly evolving spread of security risks. It seems as though every week there are new headlines about cyberattacks bringing an organization's computers or network to its knees, with the resulting bad publicity and embarrassing revelations appearing as front-page news. This raises the question of how to protect organizations and systems from these issues.

The best protection is the development and implementation of plans and procedures to improve intrusion detection *and* prevent/eliminate vulnerabilities. One way to demonstrate the need for those types of procedures is to perform a cybersecurity audit. A better process is to send a clear request for proposal to potential audit suppliers which may move the process forward much more effectively.

The traditional approach in cybersecurity is to focus on the most crucial systems and/or components and to protect them against the biggest known threats, leaving some less important system components undefended and exposed to some less dangerous risks. This approach is insufficient for the currently expanding digital networked systems environment. The reason that cybersecurity professionals believe that traditional approaches to securing cyber-physical systems information are becoming unmanageable is because the possible threat environment has become extremely complex. In this regard, cyber-physical systems (see Sect. 6.5.1) have been identified as vulnerable to cyberattacks because of their network-based accessibility, which makes them vulnerable to remote access. Thus, the consolidation of cyber and physical components within cyber-physical systems enables new categories of vulnerability to develop with regard to:

- Interception
- Replacement
- Removal of information from communication channels

This results in malicious attempts by cyberattackers to affect cyber-physical systems operations by:

- Capture
- Disruption
- Defect creation
- Failure

The reason for this vulnerability can be traced back to the way in which cyber and physical components are integrated into sensor and communication networks. Sensor networks consist of many tiny components, each of which is subject to physical capture. Communication networks are systems of interconnected units that structure information exchange while allowing access to digital technology. This is becoming more and more essential when considering the extreme daily use of smartphones, tablets, gadgets, and other smart devices. Using today's new digital technology, it is easy to access a better quality of information, in greater quantity, at faster speeds via the Internet. But the vulnerability of this cyber-based infrastructure is a huge problem on which cyber criminals are capitalizing through attacks on sensory and communication networks. Thus, cybersecurity is both a critical area and one that is the most vulnerable to exploitation in the context of very complex supply chains and cyber-based operational infrastructures. In the vulnerable space, cyber components provide:

- Computing
- Control software
- Processing
- Sensory support

They facilitate the analysis of big data received from various smart sources, social media collaboration, and a cyber-physical system's overall operation. Therefore, a single successful cyberattack on a critical system node, if unmitigated, can have the potential to affect a significant number of important operational capabilities resulting in (see Sect. 6.1):

- Defective operation
- Denial of service (DoS), a common attack in the cyber domain
- Destruction and exfiltration
- Information corruption
- And others

Hence, cyberattacks causing denial of service may occur by creating an artificial mechanism that keeps the targeted systems unnecessarily busy, delaying or denying regular operational system services, which may be avoided if the intrusion method can be determined, and measures are taken to defend against it. Therefore, the software needs to be designed for the appropriate level of security from the outset; and some cyber-physical systems may need to be checked for resilience before being used. However, numerous solutions are available that analyze patterns and signatures in program codes and behavior of program executions in order to identify the presence of malicious agents or malware, helping system administrators to disable them. The techniques used for intrusion detection (and prevention) can be classified as follows (Zeltser 2015):

- *Behavior Detection:* Observes program execution and attempts to detect malware by looking for suspicious behavior(s), such as:
 - Unpacking of malcode
 - Modifying host files
 - Observing keystrokesNoticing such intrusions allows antivirus tools to be activated and the presence of previously undetected malware on the protected system to be detected. Therefore, behavioral detection makes the use of antivirus tools an intrusion prevention technique.
- *Cloud-Based Detection:* Detects malware by collecting data from protected systems and analyzing it on the provider's infrastructure. This is usually done by capturing relevant details about the file(s) and its execution on the endpoint of a line and providing them to the cloud engine for processing. Moreover, the vendor's cloud engine can derive patterns related to malware characteristics and behavior by correlating data from multiple systems. Hence, a cloud-based engine allows individual users of the antivirus tools offered to benefit from the experience(s) and knowledge of other members of the cloud community regarding intrusion detection and prevention.
- *Heuristics-Based Detection:* Detects malware by statically examining files for suspicious characteristics without an exact signature match. Thus, an antivirus tool might look for the presence of rare instructions or junk code in the examined file(s). The antivirus tool might also emulate running the file to trace what it would do if executed, attempting to do this without noticeably slowing down the running system. A single suspicious attribute might not be enough to mark the file as malicious. Finding several such characteristics, however, might exceed the predetermined risk threshold, leading the antivirus tool to classify a file as malicious.
- *Signature-Based Detection:* Uses key aspects of the examined file(s) to create a static fingerprint of known malware. A signature could represent a series of bytes in the file(s). It could also be a cryptographic hash of the file(s) or its section(s). This method of malware detection has been an essential aspect of antivirus tools since their inception; it remains a part of many antivirus tools to date, though its importance is diminishing. A major limitation of signature-based detection is that this method is unable to mark malicious files for which signatures have not yet been developed. Thus, modern cyberattackers frequently mutate their creations to retain malicious functionality by changing the file's signature.

In general, antivirus vendors have to incorporate multiple layers into their tools to keep up with the intensifying flow of malware samples, as relying on a single approach is no longer a viable option. Malicious files can do anything any other program/file can, such as:

- Erasing a stored file
- Stopping a running program
- Writing a message on a computer screen
- And others

Moreover, malicious files may do nothing at all immediately; they can be embedded to lie dormant, undetected, until some event triggers the file to act. The trigger used can be any of the following, some combination of these, or a random situation.

- Condition
- Count
- Date
- Event
- Time
- Time interval

In fact, malicious file(s) can pose different threats each time or nothing most of the time with something dramatic on occasion. Malicious files (code) can touch everything the user can touch and in the same ways. Users typically have complete control over their own program code and data files; they can read, write, modify, append, and even delete them. However, malicious files (code) can do the same, without the user's permission or even knowledge. There are different types of malicious files, (code) as shown in Table 6.6, which can be used to introduce cyberattacks.

The term virus was coined because the affected system reacts like a biologically infected system, meaning it infects other healthy components/systems by attaching itself to the program code of the respective component/system and either destroying it or coexisting with it. The infection usually spreads at a geometric rate, eventually overtaking an entire system and spreading to all other connected systems.

A common means of virus activation is an attachment to an e-mail message. In this attack, the adversary tries to convince the recipient of an e-mail message to open

Table 6.6 Types of malicious files (code)

File/code type	Characteristics
Logic bomb	Triggers action when a specific condition occurs, such as time, date, count, interval, or some combination of these
Rabbit	Virus or worm that self-replicates without limits with regard to exhausting computing resources
Time bomb	Triggers action at a specified time
Trap door	Allows unauthorized access to functionality
Trojan horse	A login script that solicits the user's login and password and passes the identification information on to the rest of the system for login processing
Virus	Transient or resident viruses are known. A transient virus has life depending on the host's life. A resident virus locates itself in a memory; it can then remain active or be activated as a stand-alone program
Worm	Propagates copies of itself through the network and operates through the network. In comparison, a virus spreads through any medium but usually uses copied programs or data files

the attachment. Once the viral attachment is opened, the activated virus can run its intended task. The virus can be executable code embedded in an executable attachment, but other types of files are equally dangerous. For example, objects, such as graphics or photo images, can contain code to be executed by an editor, so they can be transmission agents for viruses. In general, it is safer to force users to open files on their own rather than automatically.

In the simplest case, a virus inserts a copy of itself into the executable program file before the first executable instruction. Then, all of the virus instructions execute first; after the last virus instruction, control flows naturally to what used to be the first program instruction. Such a situation is shown in Fig. 6.9 (Pfleeger et al. 2015). It should be mentioned that this kind of attachment is simple and effective because the cyberattacker does not need to know anything about the program to which the virus will attach, and often the attached program simply serves as a carrier for the virus. The virus performs its task and then transfers to the original program.

Let's assume the cyberattacker wants to prevent the virus from being detected. He arranges for the virus to attach itself to the program that constructs the listing of files on the disk. If the virus regains control after the listing program has generated the listing but before the listing is displayed or printed, the virus could eliminate its entry from the listing and falsify space counts so that it appears not to exist. This is called a surrounding virus and is shown in Fig. 6.10 (Pfleeger et al. 2015).

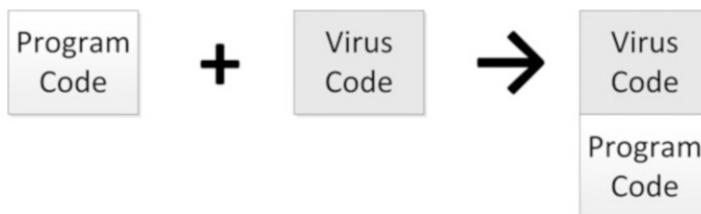


Fig. 6.9 Virus appended to a program code

Fig. 6.10 Virus surrounding a program code

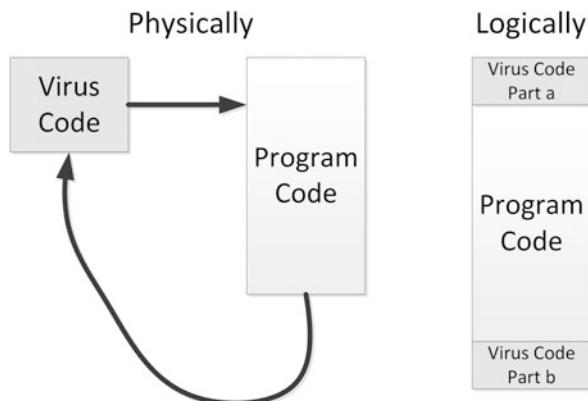
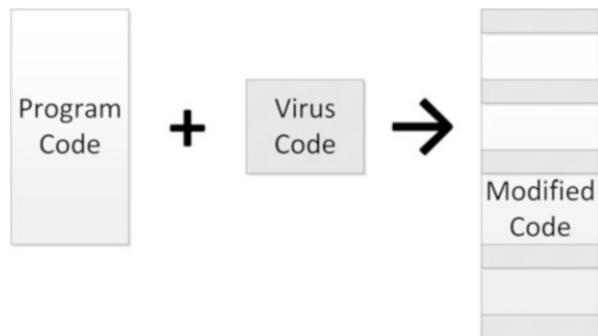


Fig. 6.11 Virus integrated into a program code



Finally, a third situation occurs when the virus replaces some of its target, integrating itself into the original code of the target. This situation is shown in Fig. 6.11 (Pfleeger et al. 2015) where the cyberattacker has to know the exact structure of the original program to know where to insert which pieces of the virus. Finally, the virus can replace the entire target, either mimicking the effect of the target or ignoring the expected effect of the target and performing only the virus effect. In this case, the user is most likely to perceive the loss of the original program.

The only way to prevent virus infection is to not share executable code with an infected source. Nevertheless, there are several techniques for building a reasonably safe community for electronic contact, including the following (Pfleeger et al. 2015):

- *Use Only Commercial Software Acquired from Reliable, Well-Established Vendors:* The good name of even highly reputed enterprises which have significant reputations could be seriously damaged by even one bad incident, so they go through some degree of trouble to keep their products virus free and to patch any problem-causing code right away. Similarly, software distribution companies are careful about products they handle.
- *Test All New Software on an Isolated Computer:* Test new software first on a computer with no hard disk, not connected to a network, and with the boot disk removed. Run the software and look for unexpected behavior. Test the computer with a copy of an up-to-date virus scanner created before running the suspect program. Only if the program passes these tests should it be installed on a less isolated machine.
- *Open Attachments Only When Knowing Them to Be Safe:* An attachment from an unknown source is of questionable safety. Also an attachment from a known source but with a peculiar message may not be trustworthy.
- *Make a Recoverable System Image and Store It Safely:* This clean version will allow a secure reboot because it overwrites the corrupted system files with clean copies. For this reason, the image has to be kept write-protected during reboot. For safety reasons, an extra copy of the safe boot image may be helpful.
- *Make and Retain Backup Copies of Executable System Files:* In the event of a virus, the infected files can be removed and reinstalled from clean backup copies (stored in a secure, offline location).

- *Use Virus Detectors (Virus Scanners) Regularly and Update Them Daily:* Many of the virus detectors available can both detect and eliminate infection from viruses. Several scanners are better than one because one may detect the viruses that others miss. Scanners search for virus signatures; they are constantly being revised as new viruses are discovered. New virus signature files, or new versions of scanners, are distributed frequently. Virus detector signature files should be kept up to date.

As more highly technological devices are introduced to the public, the more the demand for security rises. For this reason, various security schemes have been proposed, such as:

- Anomaly detection
- Probabilistic dependence graph
- Smart tracking firewall

Anomaly detection is a method of detecting anomalous behaviors or data. It mainly focuses on detecting intrusive methods based on their anomalous activities, those that are outside of the regular activity profile in a system. There are several possible approaches to tackling this challenge. The first approach is to focus on the behavior of insider cyberattacks and the design of new anomaly detection methods which utilize solid models of what acceptable behavior is and what a cyberattack is, thereby avoiding a high number of false-positive alarms. They may be caused by typical behavior that is actually normal and authorized, since normal behavior may easily and readily change (Dilek et al. 2015). Other limitations refer to the following properties (Barika et al. 2010; Bitter et al. 2010; Patel et al. 2010):

- Anomaly detection has to be able to characterize normal patterns and create a model of normal behavior; wide-ranging training sets of normal system activities are needed. Any change in a system's normal patterns must lead to a necessary update of the knowledge base.
- If intrusion detection and prevention inaccurately classifies a legitimate activity as a malicious one, the results can be very unfortunate since it will attempt to stop the activity or change it.
- Intrusion detection, no matter how efficient, may be disabled by cyberattackers if they can learn how the system works.
- In heterogeneous environments, there is an issue of integrating information from different sites.
- Another problem involves supplying intrusion detection that will conform to legal regulations, security requirements, and/or service-level agreements in the real world. First, however, the intrusion method must be identified so that the regular operation of the cyber-physical system will remain undisturbed.

The second approach in anomaly detection is to not to revise the existing anomaly detection techniques but to build upon them using novel game theory techniques to

exploit the inside intruder's weakness, in particular the fear of detection (see Sect. 6.1.5). But cyber criminals have always new ideas to disguise harmful data and overcome network protection measures. In this regard they use advanced bypass methods to deliver exploits or other malicious content to a vulnerable destination in a way that makes traffic seem normal and pass through security controls. Because multiple log levels are used which allow to overcome easily most security solutions.

The dependence graph (see Sect. 6.1.6) is a directed graph representing the dependencies of several nodes toward each other. For a given a set of nodes S and a transitive relation $R \subseteq S \times S$ with $(a, b) \in R$, modeling a dependency a needs b to be evaluated first. Hence, the dependency graph is $G = (S, T)$ with $T \subseteq R$ and R as transitive closure of T . Fault detection and localization in systems are methods with which dependability can be measured to ensure a secure function. However, fault event diagnosis systems are not equipped, in any case, to detect fault events due to malicious attacks or naturally occurring events. To resolve these issues, a probabilistic graphical approach can be used that spatially correlates information from the systems and statistical hypothesis testing. A Gaussian Markov Random Field (GMRF) can be used to model a system's random variables and study their dependencies. The dependence graph illustrates the connections using a Markov Random Field (MRF) that is induced by a minimal neighborhood system by inserting an edge between sites that are neighbors. The Gaussian random variables can then be used to approximate fault diagnostics due to malicious intrusions (Landrum et al. 2014).

A smart tracking firewall is a security method for preventing intrusions by malicious nodes that infiltrate a secure wireless mesh network, which is a communication network made up of nodes organized in a mesh topology. It is also a form of wireless ad hoc network. The mesh nodes in the network have the ability to locate and deposit previously intruded nodes into either a blacklist or a graylist. A node blacklisted by a client cannot communicate with the client by either sending or receiving information. A mesh node can archive a malicious node into the graylist when neighboring nodes send alerts about a blacklisted node (Landrum et al. 2014).

Besides the common means of virus activation through an attachment to an e-mail message, the spear-phishing attack is a real particularly perfidious new cyberattack form. This is a mail to a recipient that looks like a message from a friend or colleague. It may, for example, point to a topic field on which the addressee is currently working on and the mail simply refers to the name of a study or publication that may be of interest to the addressed. The e-mail received does not contain a link to a specific page on the Internet nor an attached PDF document, only a final short greeting. In this form of a cyberattack, the attacker knows not only the personal mail address of the attacked person but also details from the attacked person's personal and professional settings, which may easily be filtered out of the so-called social networks, since many people today surrender a lot of themselves. In this way, a cyberattacker succeeds in establishing the identity of a friend or colleague, so that the mail appears completely harmless, which makes it almost impossible to recognize the attack. Since the mail contains no link and no file, the addressed person may google the note mentioned in the mail. In this way, the attacked person accesses

a page prepared by the attacker in which the spy software installed by the attacker is deposited, which from now on scans data from the then infected computer without the attacked person's knowledge. Often the affected person only notes months later that his computer was hacked.

6.1.8 Automotive IT and Cybersecurity

The automobile industry is currently undergoing an unprecedented wave of innovation, as automakers are pioneering innovative technologies that make vehicles safer than ever before. Besides this, the automobile industry is also undergoing a radical transformation from the traditional automaker's business into a digital electronic component manufacturer's business, enhancing and creating new features. This so-called digital transformation is not only redefining business within the automotive industry but is also expanding automotive industry boundaries. Competition is global, and digital technologies have provided resources to go after new opportunities. The reason for this lies in the effective delivery of digital services which requires:

- Transition from a product-centric approach to an ecosystem-centric one
- Seamless integration across different industries, leading to cooperation or coexistence of competition and cooperation

Therefore, automakers, in addition to their automotive products, may have to collaborate with various stakeholders to create a connected vehicles ecosystem, as the stakeholders include:

- Device/component or system manufacturers
- Insurance providers
- Service operators
- Telecommunication operators
- And others

Furthermore, products and services, information, and customer expectations can all be reshaped using new capabilities for mobility, interactivity, and information access. Moreover, connected vehicles become interlinked together through smart devices, such as smartphones, tablets, roadside units (RSU), and others. In the near future, it is predicted that innovative vehicle services will be offered, such as:

- Adaptive cruise control
- Autonomous driving
- Crash avoidance systems

These require connected vehicles, so-called vehicle-to-X (V2X) communication features, such as:

- *Vehicle-to-Infrastructure (V2I)*: This is a concept in which vehicles and roadway infrastructure exchange safety and operational data. In this approach, wireless communication occurs between vehicles and infrastructure, such as smart traffic signals, RSUs (see Sect. 6.5.3), and others.
- *Vehicle-to-Mobile (V2M)*: Technology that uniquely integrates wireless and cellular networks to facilitate intelligent transportation systems applications, such as the AGORA versatile framework for the development of intelligent transportation system applications (Salahuddin and Al-Fuqaha 2013).
- *Vehicle-to-Vehicle (V2V)*: Technology allowing vehicles to communicate with each other. V2V is also known as a vehicular ad hoc network (VANET), a variation of the mobile ad hoc network (MANET), and helps drivers to overcome blind spots, avoid accidents, and other serious dangerous situations.

All key components of intelligent transportation systems (ITS), these components do raise a number of issues/questions with regard to:

- How the IoT is affecting connected vehicles and how to detect and defend against malicious data intrusions.
- How functional safety and security are meshing and becoming more intertwined and what it means for future collaborative developments for the automotive manufacturing companies and their Tier 1 suppliers with regard to securing inter domain communication while trustworthiness is needed for cooperation.
- Cyber risks from the view point of a litigator pursuing a class action law suit.
- Key developments in regulations on data privacy and security across the whole life cycle.
- Security solutions in advanced network architectures and encryption methods which includes vulnerability and incident handling.
- Strategies to properly secure automotive telematics and infotainment systems.

A diagram depicting the digital transformation in vehicles and the associated security challenges is shown in Fig. 6.12.

With regard to the aforementioned issues, the automobile industry is also facing emerging challenges in the area of cybersecurity. The members of the Alliance of Automobile Manufacturers (Auto Alliance) and the Association of Global Automakers believe that by proactively and collaboratively addressing potential cybersecurity challenges, the automobile industry can continue to produce safe vehicles that incorporate modern and robust security options. But defending against cyberattacks often requires collaborative engagement between multiple stakeholders. There are benefits to building partnerships across the vehicle ecosystem, including sharing of cyber threat trends and proven techniques with third parties to defend against cyberattacks which require trustworthiness to support confidence about security levels of involved partners.

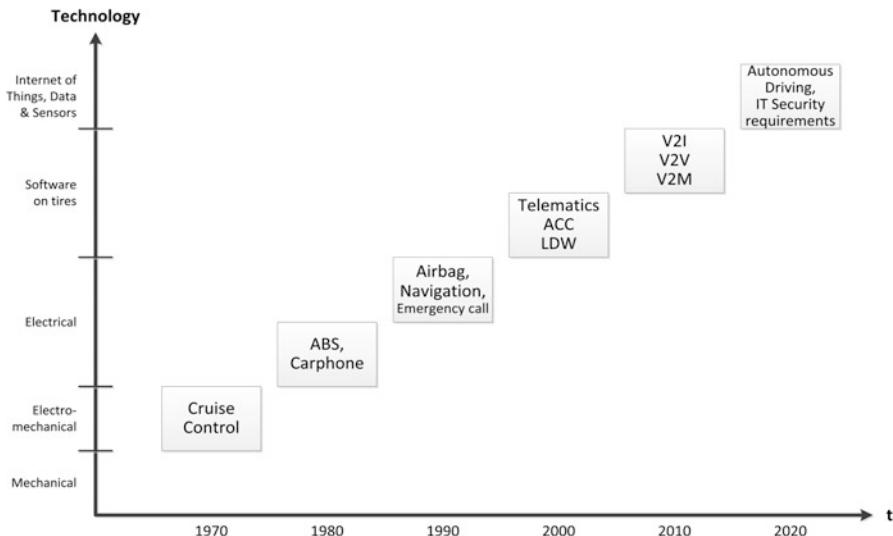


Fig. 6.12 Digital transformations in vehicles and associated security challenges

As written in the “Framework for Automotive Cybersecurity Best Practices” ([URL2 2016](#)):

“...an incident response plan documents processes used to help respond to cybersecurity incidents affecting the motor vehicle ecosystem. A comprehensive response plan that develops increased awareness and capabilities and that establishes communications protocols between automotive manufacturers, suppliers, cybersecurity researchers, and government agencies could assist industry stakeholders in coordinated efforts to address discovered vulnerabilities and enhance product security.”

The forthcoming best practices aim to address incident response plans that may include processes to activate response teams, notify an internal chain-of-command, and trigger response activities to assess and counter cyber attacks. A comprehensive incident response plan provides strategic flexibility for managing many types of cyber incidents and takes into account internal resources and, where appropriate, external resources likely needed to support incident response measures.

The development of protocols for recovering from cybersecurity incidents is also important for ensuring consistent approaches for making available updates to vehicles in a reliable and expeditious manner based on specific circumstances”.

Therefore, security in vehicles in general is a challenge for automakers today because it is a moving target:

- As more smart digital devices connect to each other, the public’s dependency on them is creating more movement toward connecting them with mobile vehicles, such as cars, buses, trucks, trains, aircraft, etc.
- Safety and security issues related to automotive objects are becoming more and more relevant in the realm of Internet-connected devices and objects which require long term capable security.

Security evaluation methods are needed for identifying and removing software security vulnerabilities as well as vulnerability analysis and intrusion detection and prevention systems with regard to security governance and risk posture. Connected vehicles may be targets for cyberattacks because:

- Vehicles are frequently parked in unsecured locations.
- Vehicles can be used to inflict serious bodily injuries.
- Vehicles could be targeted for antisocial activities such as terrorism.

Thus, growing needs exist to understand and address technology and policy issues around cybersecurity with regard to embedded wireless connected technology in vehicles. In addition, today's vehicles are becoming more and more equipped with intelligent electronic control modules which support drivers in tremendous ways, ranging from simple functions such as:

- Dashboard modifications
- Navigation
- Streaming of personal music via smartphones, as well as customized media content
- Vehicle adjustments

to

- Semiautomated driving on highways

to

- OEMs which are competing with each other to integrate the most up-to-date features emerging from the consumer electronics industry, as well as connectivity solutions that enable valuable remote services with their inherent security problems.

However, autonomous driving technologies are increasing the demand for continuous connection of the vehicle's ECUs to a variety of cloud services that would help to improve advanced processing and subsequent vehicle maneuvering strategies, accompanied by the possibility of distributing new software updates and other essential content into every ECU or infotainment system. Hence, ubiquitous internal and external connectivity is undoubtedly the gatekeeper for future needs and possibilities within the automotive industry with regard to security design.

All of these advancements are calling for vehicle cybersecurity, a problem which is not trivial with regard to specific requirements, such as speed, real-time constraints, etc., and contradictory expectations. Industrial standards are still under development, such as IEEE P1556: Security and Privacy of Vehicle and Roadside Communications Including Smart Card Communications. Today, communication is typically done over dedicated short-range communications (DSRC) at the 5.9 GHz level based on the IEEE 802.11p protocol. Therefore, one problem of vehicle

Table 6.7 Vulnerable access points

Communication	In-vehicle hacking	Remote hacking
Channel hacking		
RFID keys: Embedded with RFID tag and a reader in the vehicle. Vehicle can be immobilized if the correct tag is not verified	CDs and USB connectivity, and physical interface for entertainment devices: Entertainment Systems and CAN bus connectivity to update ECU firmware interface with systems within the vehicles	Cellular/telematics connectivity units: Equipped with connectivity used for various functions. Provides access to internal network and ECU
Keyless entry: Remote keyless entry used to open doors and activated alarms can be blocked by interfering transmitters allowing access to vehicles	ODB II port: Provides a regulated access to CAN buses to control key components	Dedicated short-range communication (DSRC): Emerging technology proposed standard for cooperative driving. Can potentially transmit malicious inputs to other vehicles causing damage
Tire pressure monitoring system (TPMS): Alerts drivers about tire pressure readings; can be manipulated showing inconsistent readings		Wi-Fi hotspots: Make vehicle's OBD II port vulnerable to attacks by connecting wirelessly
Bluetooth: Used as standard supporting hands-free callings. Paired with phones it can be a medium for downloading malware		

cybersecurity lies in the advancements in malicious methods and tools emerging in traditional ICT environments which can be applied to automotive systems with no additional cost or effort and which can be a significant threat to safety. Elements such as automotive-specific vehicle communication buses do not offer robust protection against advanced attack vectors. Hence, in Table 6.7, vulnerable access points are summarized with regard to the chosen attack method.

With regard to Table 6.7, cyberattacker's methods of attacking vehicular communication can be manifold because a cyberattacker can:

- Attack against liability-related messages by cheating with its own identity, position, speed, etc.
- Be an inside or an outside cyberattacker, whereby the insider has to be prevented from cheating about its own position; and the outsider has to be prevented from spoofing the position on an honest traffic node to secure positioning.
- Disrupt network operation which results in a denial-of-service attack.
- Intrude bogus information against traffic information, such as “a traffic jam is ahead.”
- Undefended uncovering of identities of other vehicles.

In cases where vehicles carry a certified identity and public key, such as an electronic license plate (ELP), mutual authentication can be done. Authorities are able to cross-certify a vehicle's position by using verifiable multilateration for vehicle identification, as is used in aviation. Multilateration is a surveillance application that accurately establishes the position of transmissions, matches any identity data that is part of the transmission, and sends it to the air traffic management (ATM) system. Multilateration is considered to be a cooperative surveillance technique, combining a dependence on target-derived data for identification and altitude with ground-based calculation of position (URL3 2016). Thus, using this secure, verifiable multilateration (triangle) positioning technique in the automotive domain (Hubaux et al. 2004) results in the following:

- A vehicle located within the triangle cannot prove to be at another position within the triangle except at its true position.
- A vehicle located outside the triangle formed by the verifiers cannot prove to be at any position within the triangle.
- An outside adversary cannot spoof the position of a vehicle such that it seems that the vehicle is at a position different from its real position within the triangle.
- An outside adversary cannot spoof the position of a vehicle such that it seems to be located at a position within the triangle, if the vehicle is out of the triangle.

6.1.9 Attack Value Chain

The latest and greatest advances in technology have created greater efficiency and effectiveness for all kinds of industries. However, the pace of data breaches and intrusions into secure industrial systems, such as computers and communication networks, is accelerating at an alarming rate. The present risks and potential new avenues of compromise and increasing sophistication of intruders are making computers and communication networks more vulnerable. To manage these risks, automakers must enhance and standardize their security procedures including vendors, partners, and even customers looking for potential weaknesses in the attack value chain to secure their own as well as suppliers products and services, as shown in Fig. 6.13. Viruses or malware carried in a smartphone or in an infotainment system can easily invade automotive electronics. Therefore, cybersecurity in vehicles has to maintain zero compromise on security, preventing costly and massive cyber-attack-caused recalls.

Moreover, an unsecure implementation of communication protocols within an infotainment system (see Fig. 6.13) can lead to a remote access tunnel for cyberattackers who are then able to remotely deactivate critical safety elements, such as steering and braking systems, during driving. These types of attacks can be triggered from any place outside the system at any time. Other vulnerable vehicles can be identified with simple ICT methods with regard to an unsecure configuration of a mobile network provider. Therefore, efficient security features within systems and/or components are required which protect against critical threats assumed to

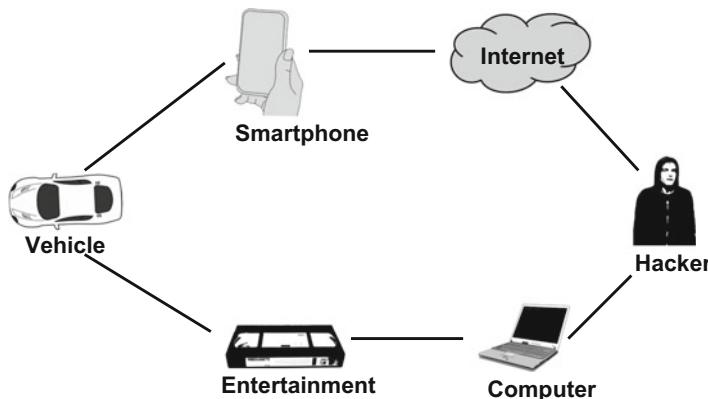


Fig. 6.13 Attack value chain in vehicles

happen. Furthermore, important documentation and source code for securing elements of the vehicle's electric backbone could be bought with minimal effort.

Today's vehicle systems are not designed to continuously upgrade software and hardware to integrate or update security elements. While this could prevent some attack vectors, it could require more processing power than what is available for sustainable testing and validation efforts. Mission-critical ECU components, such as system-on-a-chip (SoC) designs, have appropriate cybersecurity extensions already in place, albeit at a significantly higher cost. It can be assumed that newer software features would increase the cybersecurity level in the automotive domain, such as virtualized or vehicle bus message encryption, which relies on certain hardware-based elements that are not yet integrated into today's vehicles.

Original equipment manufacturers (OEMs), Tier 1 suppliers, and other contributors of complex vehicle cyber-physical systems are facing the same issues with successful and efficient integration of cybersecurity elements. At first glance, the implementation of cybersecurity elements within the automotive industry is seen purely as an additional cost which doesn't innately entice more customers to buy a specific car. Unfortunately, this results in poor integration of security functions over the long term, which will add further costs, both directly and through costly fleet-wide recall events. Thus, cybersecurity is certainly an opportunity for OEMs and Tier 1 suppliers with strong bottom-line implications. In this regard the design and manufacturing of vehicular components and systems as well as vehicles itself require to follow a new design and manufacturing paradigm, which can be stated as security by design, as it has been introduced by the German Industry 4.0 Platform (URL1 2018). The risk of paying penalties or recall costs for security-related issues may soon be as real as costs related to vehicle safety. The primary subcategories of threats to vehicles can be summarized as follows (Bittersohl and Thoppil 2015):

- *Compromised Privacy*: Interception or readout of privacy-related user data that is directly connected with personal details of drivers, which can be stored within cloud services, etc., such as:
 - Billing details
 - Destination targets
 - Driving behavior
- *Dysfunctional Sensor Processing*: Disturbing sensor input for further processing of vehicle maneuvers through modification of bus communication systems or unauthorized software modification directly to ECUs
- *Man-in-the-Middle Attack (MITM)*: Interception of internal and/or external vehicle communication in order to obtain information from ECU-to-ECU communications or other mission-critical software elements
- *Side-Channel Attack*: Utilizing weaknesses in hardware, software, and communication protocols in a system connected to a cyberattack target in order to open an unprotected channel
- *Spoofing*: Faking the presence of communication partners and information that is used to control advanced sensor systems, thereby activating maintenance functions within vehicles and creating new possibilities to modify a vehicle system's configuration

Recent reported events have shown that security breaches into vehicles are sophisticated cyberattacks combining several attack vectors, as shown in Table 6.8. Therefore, the objective is to find the weakest links of the integrated cyber-physical systems. Infotainment systems are often identified as the ideal target as they are based on highly complex and modern operating systems. The huge volume of software code implemented for features such as navigation, radio, video/audio, and external content makes the effort of maintaining secure code more and more complicated if not unmanageable, especially due to the high amount of individual internal or external third-party partners working on such projects. In addition, the integration of vehicle communication protocols and layers into the infotainment system further increases the threat of a cyber attacker gaining access to the more mission-critical elements of the vehicle, which can ultimately result in denial of service (Bittersohl and Thoppil 2015).

Without a doubt, it took significant effort to identify the vulnerabilities essential to achieve far-reaching access to vehicles. Henceforth, research will be based on a combination of different attack categories with the objective of discovering the weakest link within each subdomain. Prior to now, OEMs had no other choice but to recall vulnerable vehicles as the functionality of a remote over-the-air (OTA) update was not yet implemented.

6.1.10 Holistic Cybersecurity Solutions

Despite efforts to protect vehicle cyber-physical systems against cyberattacks, the attacks are growing in number and sophistication. This indicates that a change in the

Table 6.8 Attack value chains

Attack vector value chain	Critical element of attack vector part	Attack category	Vehicle attack targets
Critical communication systems not protected for external access (e.g., Wi-Fi, 4G).	Direct access to critical vehicle communication	Man-in-the-middle	Vehicle bus communication
	Elements with consumer electronic devices		Traffic control unit
Easy access to operating system images and decompiling of software components	Publicly available software images with/without encryption	Side-channel	Infotainment
Modified operating system image transferred to infotainment system without security integrity check for unauthorized modification	Unauthorized software modifications possible.	Side-channel	Comfort systems
	Extraction of critical infrastructure data		Traffic control unit
Readout of cellular network configuration and identification of potential targets	Spoofing	Compromised privacy	Vehicle Wi-Fi
	Extraction of critical infrastructure data		V2X (DSRC)
Modifying CAN chip software through reflash order to send unauthorized messages to other critical ECUs in vehicles	Unprotected external developer/diagnostic tools	Side-channel	Smartphone
	No message authentication		Connected services
Utilizing publicly available diagnostic tools to reverse engineer CAN messages and unlock ECU encryptions.	Effortless decompilation of message protocols.	Vehicle communication bus manipulation	CAN
	No device authentication within vehicle bus system		FlexRay

defense strategy is required as well as acceptance of the fact that there is no panacea to overcome the ever-growing plethora of cybersecurity problems. Thus, a holistic security approach can be used which suggests system administrators look at the full picture and make a thorough analysis of the security threats to the whole system instead of securing it part by part, using a multilayer approach (Shiva et al. 2010):

- *First Layer:* Core hardware and software components. Envision each of these components as being wrapped with a self-checking module, called self-checking hardware/software components
- *Second Layer:* Traditional network security infrastructure built using techniques such as cryptographic algorithms

- *Third Layer:* Secure applications designed with built-in or built-on security approaches utilizing self-checking concepts and components
- *Fourth Layer:* Game theoretic decision module which has the responsibility of choosing the best security strategy for all three inner layers

In the past, research efforts have focused on the second and third layers. Thus, a traditional intrusion detection system (IDS) can be considered as residing in the top layer, which can be made more effective by use of game theory (see Sect. 6.1.5).

Growing distribution of common software, such as AUTomotive Open Source ARchitecture (AUTOSAR) or GENIVI, provides a basis for achieving a holistic cybersecurity approach with backend services. However, the final decision regarding which cybersecurity feature should be integrated depends almost entirely on the OEM. Costs and supplier readiness are main decision drivers for each technology as well as OEM organizational readiness with regard to developing and adhering to cybersecurity policies and guidelines.

6.1.10.1 AUTOSAR

AUTOSAR is a worldwide development partnership of automotive interests founded in 2003 to create and establish open, standardized software architecture for automotive ECUs, excluding infotainment (see Sects. 4.6 and 4.7).

Development goals include scalability to different vehicle and platform variants, transferability of software, consideration of availability and safety requirements, collaboration between various partners, sustainable utilization of natural resources, maintainability throughout the whole product life cycle, and process managing the entire life cycle of a product from inception, through engineering design and manufacturing, to service and disposal of manufactured products.

AUTOSAR is driven by the advent of innovative vehicle applications, contemporary automotive electrical/electronic (E/E) architecture that has reached a level of complexity requiring a technological breakthrough in order to manage it satisfactorily and fulfill the heightened passenger and legal requirements. This need is important for vehicle manufacturers and their leading Tier 1 suppliers who are faced with often conflicting requirements from:

- *Driver Assistance and Dynamic Drive Aspects:* Key items include detection and suppression of critical dynamic vehicle states and navigation in high-density traffic surroundings.
- *Legal Enforcement:* Key items include environmental aspects and safety requirements.
- *Passenger Convenience and Service Requirements:* Comfort and entertainment functional domains.

Leading OEMs and Tier 1 suppliers, having recognized this industry-wide challenge, decided to work together to meet the challenge. Their common objective is to create a development base for industry collaboration on basic functions while providing a platform which continues to encourage competition on innovative

functions. To this end, a development partnership called AUTOSAR was formed, including all vehicle domains with the goals of (URL4 2016):

- Collaboration between various partners
- Definition of an open architecture
- Development of highly dependable systems
- Scalability to different vehicle and platform variants
- Standardization of basic software functionality of automotive ECUs
- Support of different functional domains
- Support of applicable automotive international standards and state-of-the-art technologies
- Transferability of software

The AUTOSAR standard serves as a platform upon which future vehicle applications will be embedded and also serves to minimize the current barriers between functional domains. It will, therefore, be possible to map functions and functional networks to different control nodes in the system, almost independently from the associated hardware.

The technical goals of AUTOSAR:

- Modularity of automotive software elements to enable tailoring of software according to the individual requirements of ECUs and their tasks
- Reusability of functions to help improve product quality and reliability and to reinforce corporate brand image across product lines
- Scalability of function to ensure the adaptability of common software modules to different vehicle platforms and prohibit proliferation of software with similar functionality
- Transferability of functions to optimize the use of resources available throughout a vehicle's electronic architecture

This will help to provide a common software infrastructure for automotive systems of all vehicle domains based on standardized interfaces for the different layers, as shown in Fig. 6.14. This common infrastructure will be comprised of the following elements:

- *Electronic Control Unit (ECU)*: The physical hardware.
- *Runtime Environment (RTE)*: All communication between software components and basic software, including the operating systems (OS) and communication services, is carried out through the RTE layer.
- *Main Software (MSW)*: A combination of:
 - *Basic Software*: Builds on RTE to provide some general utilities which provide the overall functionality of the AUTOSAR infrastructure (software components and RTE on an ECU). Basic software is essential for running the functional part of the software; however, it does not fulfill any functional job itself. The software components do that.

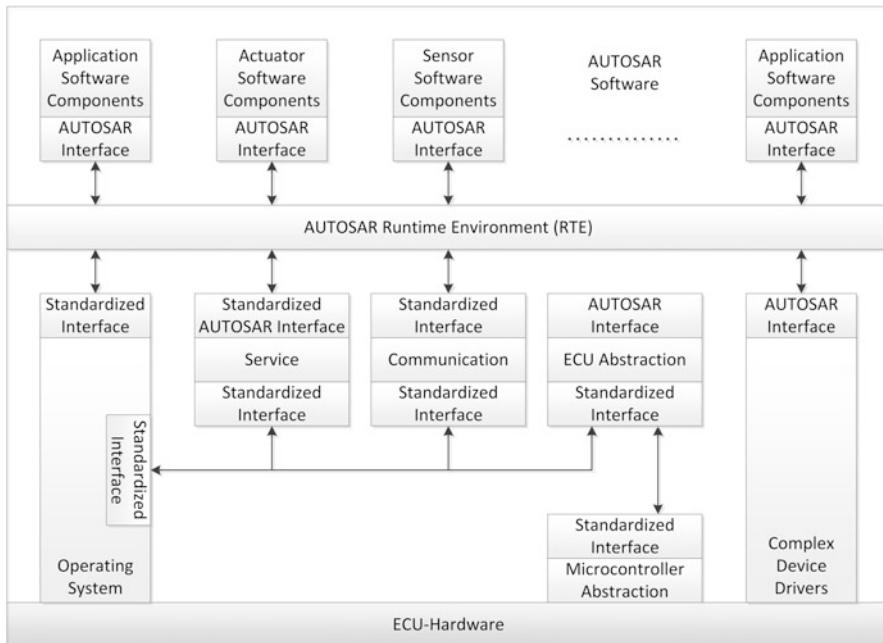


Fig. 6.14 AUTOSAR ECU software architecture (source: www.autosar.org)

- *Software Components*: Base of any software assembly is the implementation of parts of the functionality of the automotive application. Software components are the fundamental building blocks of AUTOSAR systems. Types of software components are:
 - Application software components
 - Actuators/sensor software components
- *Complementary Software (CSW)*: Manufacturer- and model-specific software.

Hence, standardization of functional interfaces across automakers and suppliers and standardization of the interfaces between the different software layers is seen as a basis for achieving the technical goals of AUTOSAR. AUTOSAR provides a standard description format for the interfaces as well as other aspects needed for the integration of the AUTOSAR software components.

The constantly growing complexity of software also increases the specific needs for the network infrastructure in a vehicle. Therefore, in addition to the standard CAN bus, other bus systems have been embedded in vehicles. The use of these bus systems is a challenge for automakers and their suppliers as they seek to protect their systems against cyberattacks. The main focus here is on cyberattacks at the protocol level which may result in the following, which always result in paralyzing the ECUs.

- Denial-of-service attacks
- Falsification of sender addresses using Internet Protocol (IP) spoofing
- Redirection of network traffic using Address Resolution Protocol (ARP) spoofing

IP spoofing (see Table 6.6) is a technique used to gain unauthorized access to computers, whereby the cyberattacker sends messages to a computer with a forged IP address indicating that the message is coming from a trusted host.

Within the Internet Control Message Protocol for IPv6 (ICMP-IPv6), functionalities like ARP are directly integrated. Thus, information about how to assign an IPv6 address or the way a controller sends its data are transmitted unprotected on the network. Hence, a cyberattack is theoretically simple, with an attacker able to impersonate the router and redirect network traffic to read it or change the content. Firewalls and IDSs can, with regard to the required computing power, only protect against such cyberattacks to a limited extent. This requires protocol extensions (Finke et al. 2015), such as the secure neighbor discovery (SEND) protocol, which is a security extension of the neighbor discovery protocol in IPv6 defined in RFC 3971 and updated by RFC 6494. SEND uses cryptographically generated addresses and other new neighbor discovery protocol options for the ICMP-IPv6 packet types used in neighbor discovery protocol (URL5 2016).

AUTOSAR considers that due to V2X applications, the requirement that vehicles interact with off-board systems will enhance the integration of non-AUTOSAR systems; and support of cloud interactions will be the next challenge that AUTOSAR has to face. In such an open access environment to select vehicle systems, a dedicated means of security is required with regard to:

- Architecture
- Cloud interaction
- Onboard communication

This will improve the existing standard, support new technologies, and enhance dynamic security architectures.

6.1.10.2 GENIVI

Compared to AUTOSAR, the nonprofit GENIVI Alliance is committed to driving the broad adoption of specified, open-source, in-vehicle infotainment (IVI) software. Therefore, GENIVI provides automakers with four unique approaches to meeting today's challenges:

1. *Define*: Allows flexible definition of IVI systems that fit customers' latest needs
2. *Partner*: Supports business model evolution and networking across the supply chain
3. *Leverage*: Provides standard, open-source architectures, tools, and software components
4. *Reuse*: Allows reuse of components and redeployment of solutions with no royalty fees

Automakers and their suppliers face at least three significant challenges in developing and delivering IVI functionality to their customers (URL6 2016):

- *Responding to Consumers:* Consumers want IVI functionality that is the same or similar to that found in consumer electronic devices, such as smartphones and tablets. New devices with the latest features are typically launched in the market on an 8- to 18-month cycle versus the 2–5 year cycle for most in-vehicle software. As a result, consumers have introduced a new competitive measure that automakers must use: the time from consumer request to in-vehicle availability.
 - GENIVI's open software approach better aligns consumer electronics and automotive development cycles.
 - GENIVI's individual software components and reusable platform provide automakers and their suppliers with the tools to perform rapid prototyping and to quickly develop and deliver IVI systems that fulfill consumer requests.
- *Complexity and Cost:* Consumer functionality requests push the amount of software in a typical IVI system to over several million lines of code. Hence, automakers have to deal with the increasing complexity and cost of developing, validating, and maintaining software. Many automakers are shifting away from the historical black box approach and are taking more ownership of the design and development process, including maximizing the reuse of legacy code to reduce costs and deploying a software platform on multiple hardware platforms based on the needs of their various models.
 - GENIVI's technical deliverables and open approach promote a wide range of supplier models based on the preferences of the automaker.
 - Automakers can launch a single reusable software platform that with limited integration can run on a wide range of automotive boards, from low- to high-end performance.
- *Customer Ownership:* Automakers are keen to keep their customer relationships sustainable. Large technology companies, such as Apple and Google, have entered the automotive market, introducing demands for user experience, branding, and data usage that limit the automaker-driver relationship. Automakers have their own business model; some prefer a single Tier 1 supplier, while others prefer multiple suppliers taking ownership of certain pieces of the overall system.
 - GENIVI's approach allows automakers to maintain their independence from technology titans pushing their own business models in the automotive industry.
 - GENIVI's flexible architecture and pick-and-mix model give automakers the freedom to include preferred, best-in-class software from multiple suppliers.

GENIVI's technical deliverables consisting of:

- Flexible technical architecture
- Individual software components
- Preintegrated, reusable IVI platform
- Standard interfaces/application programming interfaces (APIs)

which are essential to overcoming the IVI challenges faced by every automaker. Thus, GENIVI technologies are at the forefront of a new generation of IVI solutions. As one of the many GENIVI use cases, BMW has changed from its traditional

approach to IVI software development to where it is today; the first automaker to deliver a complete infotainment product, the so-called entry media and navigation system (EMNS). The EMNS rolled off the assembly line in the fall of 2013 and is now part of the MINI and 1, 3, and 5 BMW series product lines based on the GENIVI Linux platform. Since then, other automakers have selected products with GENIVI solutions making the platform available in four continents around the world. Furthermore, several additional automakers will release GENIVI-equipped systems in their vehicles during the next 2 years.

6.2 IT Security in Automotive Cyber-Physical Systems

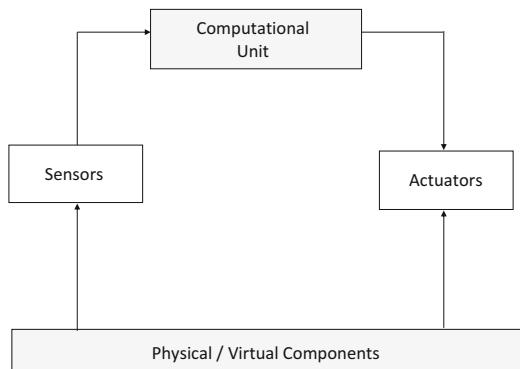
The rapid growth of information and communication technology (ICT) has prompted the expansion of networked systems that address real-world applications. This has led to the integration of computing and communication technologies with physical processes that incorporate CPS, which can be represented (see Sect. 6.1) more generally as shown in Fig. 6.15 by:

- Computed result of the physical system states which could advise the controller to select valid commands
- Control commands which are sent to actuators
- Data acquisition from sensors
- Physical data aggregation in the network

Cyber-physical systems capture novel aspects of networked systems, including integration of distributed computing systems with monitoring and control entities in the physical environment with regard to:

- *Actuating*: Executes various forms of actions determined during computing phases, such as correcting the cyber behavior of the CPS or component, changing the physical process, etc.

Fig. 6.15 Cyber-physical system architecture



- *Computing*: Reasoning and analyzing data collected during sensing/monitoring to check whether the physical process satisfies predefined constraints. If criteria are not being satisfied, corrective actions are proposed.
- *Networking*: Deals with real-time sensor node data aggregation/diffusion for process analytics. Different applications interact concurrently with networking communication.
- *Sensing*: Fundamental capability of a cyber-physical system giving feedback on any past actions which were taken by the cyber-physical system nodes, ensuring correct operation in the future.

Technological advances of CPSs have a tremendous impact on security vulnerability. Therefore, security is a relatively new realm of research. Like any other new field, most of the effort seems to be focused on mapping security solutions from existing domains onto CPS application needs. However, these solutions are usually not very well suited for CPSs because traditional security solutions were not designed for interoperation among heterogeneous applications. Thus the challenge is how to make sure that CPSs are secure while interacting with another system because major types of cyberattacks to CPSs intrude:

- Actuator devices
- Computing devices
- Networking devices
- Sensing/monitoring devices

These attacks are accomplished through (Wang et al. 2010):

- *Compromised Key Attacks*: A key is a secret code which is necessary to interpret secure information. Once a cyberattacker obtains a key, the key is considered to be compromised (Chalkias et al. 2009).
- *Denial-of-Service Attack*: A cyber criminal network attack that prevents legitimate traffic or requests for network resources from being processed or responded to by the system (Pelechrinis et al. 2011). This type of attack usually transmits a huge amount of data to the network making it too busy handling the data to provide normal services.
- *Eavesdropping*: A cyberattack where an adversary can intercept any information communicated by the system.
 - *Passive Attack*: Cyberattacker does not interfere with the workings of the system; it simply observes the system's operation (Kao and Marculescu 2006).
- *Man-in-the-Middle Attack*: False messages are sent to the operator, taking the form of a false negative or a false positive (Saltzman and Sharabani 2009).
 - *False negative*: A test result indicates that a condition failed when it was actually successful, i.e., erroneously no effect has been assumed.
 - *False positive*: A false alarm indicating that a given condition has been fulfilled when it actually has not been fulfilled, i.e., erroneously assuming a positive effect.

Prior work focused on:

- Actuating
- Computing
- Monitoring
- Networking
- Sensing

It focused on reliability and resilience in protecting CPSs against:

- Random independent or benign faults and failures of their cyber and/or physical components (Akella et al. 2010; Johnson 2010).
- Failure to adequately address integrity, confidentiality, and denial-of-service threats (Cárdenas et al. 2008; Cárdenas et al. 2011; Eisenhauer et al. 2006; Fleury et al. 2009; Mo and Sinopoli 2009).

However, conventional computer and network security approaches do not address, in a unified way, how systems outlive malicious cyberattacks which correlate with survivability or how they recover after a cyberattack, which refers to recoverability (Fleury et al. 2009). Thus, securing CPSs goes beyond securing the individual system components separately. Highly skilled cyberattackers use multivector attacks that exploit weaknesses of separate physical and cyber components of the attacked system, none of which may pose a serious threat for the corresponding component. The combined effect, however, may result in a catastrophic event if the attack vectors are dependent.

One of these multivector attacks was the Stuxnet attack (Falliere et al. 2011), which targeted the functions of industrial nuclear centrifuges used in Iran's nuclear program. In the Stuxnet attack, a worm that used zero-day exploits spread to machines using Microsoft® Windows® via local area networks (LANs) or universal serial bus (USB) sticks, carrying a malware payload that infected and reprogrammed programmable logic controllers. It is believed that Stuxnet possessed a broader panoply of cyber weapons.

Thus, there are many ongoing efforts to ensure the security of CPSs which are primarily based on extending mechanisms already used to protect separate cyber and physical components. However, there is no formal security model for cyber-physical systems that addresses security in a unified framework that deals with:

- Hardware threats
- Network threats
- Physical threats
- Software threats

There is a huge number of publications in the literature highlighting the difficulties of securing physical systems, with regard to timing attacks in particular (Fleury et al. 2009; Lamport 2005; Tang and McMillian 2008), noninterferences

(Gamage and McMillin 2009), and execution monitoring (Hamlen et al. 2006; Lamport 1997, 1998). Thus, to secure CPSs, it is important to understand cyberattacks and what can be done to prevent them from becoming successful. Hence, cybersecurity measures which address the risks expected to be present in a CPS or subsystem can be designed and implemented in such a way that access and operation to legitimate activities is not impeded, particularly during times of emergency or restoration activity.

The Institute of Electrical and Electronic Engineers (IEEE) has developed a cybersecurity standard that presents a balance of the above features, introduced as IEEE 802.11 Wireless Network Standard, which is one of the most attractive and fastest-growing networks. The IEEE 802.11 WLAN standard is extended by IEEE 802.11i, a Standard for Wireless Local Area Networks (WLANs), providing improved encryption for networks that use the popular 802.11a, 802.11b, which includes Wi-Fi, as well as 802.11g standards. The 802.11i standard requires new encryption key protocols, known as Temporal Key Integrity Protocol (TKIP) and Advanced Encryption Standard (AES). However, AES requires a dedicated chip, which means hardware upgrades for most existing WiFi networks. Other features of 802.11i are key caching for access, which facilitates fast reconnection to the server for users who have temporarily gone offline, and preauthentication, which allows fast roaming. The 802.11i standard was officially ratified by IEEE in June of 2004 and thereby became part of the 802.11 family of wireless network specifications. Since introducing WiFi, a variety of keys have been deployed:

- *Wired Equivalent Privacy (WEP)*: The first form of authentication used with WiFi. Unfortunately, it was easy to crack, and other systems are now more widely used.
- *Wi-Fi Protected Access (WPA)*: A software/firmware improvement over WEP. The first version, it is also known as WPA1 or WPAv1.
- *Wi-Fi Protected Access II (WPA2)*: Next update to WPAv1, it provides significant improvement in the level of security.

Cybersecurity in the automotive industry refers to securing the manifold automotive ECUs. ECU is a generic term for any embedded system that controls two or more of the electrical systems or subsystems in a vehicle, connected through a CAN bus as shown in Fig. 6.16.

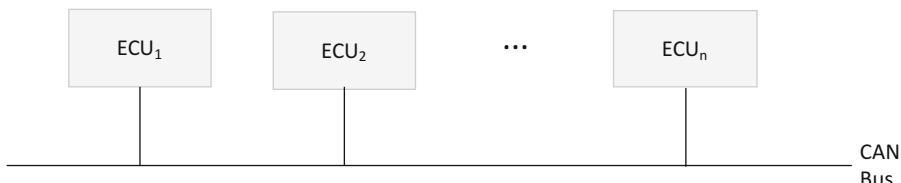


Fig. 6.16 ECUs connected to the CAN bus

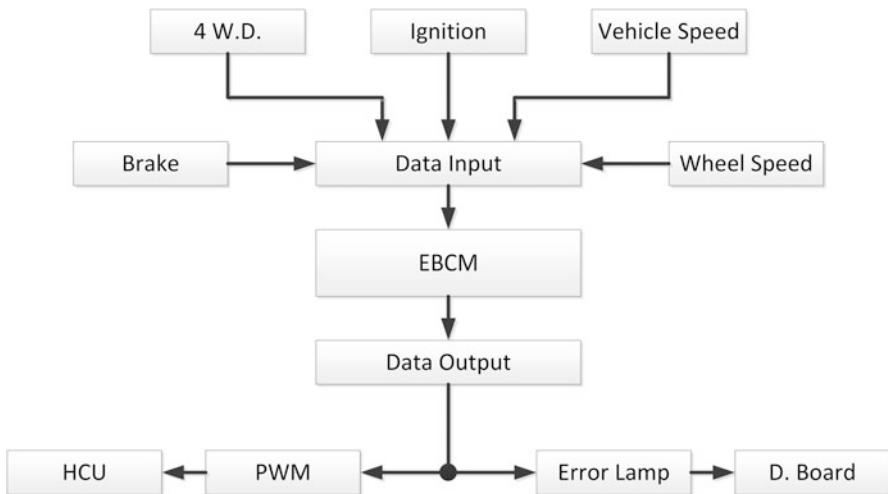


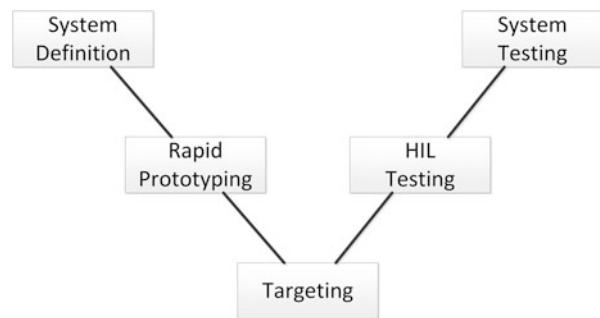
Fig. 6.17 EBCM block structure

The manifold ECU types used in vehicles includes:

- *Body Control Unit (BCU)*: Monitors and controls various electronic accessories in a vehicle's body.
- *Brake Control Unit or Electronic Brake Control Module (EBCM)*: Controls a vehicle's wheels to enhance braking ability on wet, slippery, or icy road surfaces. EBCM regulates the braking systems on the basis of five inputs, as shown in Fig. 6.17 (ni-com 2009).
 1. *Brake*: Input that monitors the status of the brake pedal, i.e., deflection or assertion. Information is acquired in a digital or analog format.
 2. *4W.D.*: Input that monitors the status in digital format as to whether the vehicle is in the 4-wheel-drive mode.
 3. *Ignition*: Input that registers if the ignition key is in place and if the engine is running or not.
 4. *Vehicle Speed*: Input that informs about the speed of the vehicle.
 5. *Wheel Speed*: Application representing a set of four input signals that conveys the information concerning the speed of each wheel, essential to derive all necessary information for the control algorithm.
 6. *HCU*: Hydraulic control unit is a unit in the antilock brake system that controls/regulates hydraulic pressure during an ABS stop.
 7. *PWM*: Pulse width modulation is used in applications such as switching mode voltage regulators, positional motor controls, fuel injector drivers, ignition drivers, and ABS control
 8. *Error Lamp*: Typically the first indicator that the EBCM is damaged and that the ABS system light will illuminate an error on the dashboard.

- *Central Control Unit (CCU)*: Scalable and modular control unit with embedded software; contains mostly internal diagnostics which enhance troubleshooting with distinct messages for fault conditions for the CCU.
- *Door Control Unit (DCU)*: Controls and monitors various electronic accessories in a vehicle's door. Since most of the vehicles have more than one door, DCUs are generally present in each door separately. The DCU associated with the driver's door has some additional functionality which is the result of complex functions, such as locking, driver door switch pad, child lock switches, etc. In most cases, a DCU acts as a master and others act as slaves in communication protocols. Features controlled by DCU are:
 - Automatic window movements
 - Child lock safety feature
 - Global open-close functionality
 - Manual window movements
 - Mirror adjustment
 - Mirror folding
- *Engine Control Module (ECM)*: Controls a series of actuators on an engine to ensure optimal engine performance by reading values from a multitude of sensors within the engine, interpreting the data using multidimensional performance maps (called lookup tables) and adjusting the engine actuators accordingly.
- *Powertrain Control Module (PCM)*: Consisting of the ECM and the transmission control unit, it commonly controls more than 100 factors in a vehicle. Inputs to the PCM come from many sensors, of different types, that are spread around the vehicle. Most of them are oriented toward engine management and performance.
- *Speed Control Unit (SCU)*: Controls the speed of a vehicle. An SCU is a servomechanism that takes over the throttle of the vehicle to maintain a steady speed as set by the driver.
- *Suspension Control Unit (SPCU)*: Responsible for keeping the steering knuckle in place. The steering knuckle connects the wheels to the suspension system, and it also contains the wheel hub or spindle.
- *Telematic Control Unit (TelCU)*: Controls tracking of the vehicle. The TelCU consists of a Global Positioning System (GPS) unit, which keeps track of the latitude and longitude values of the vehicle, an external interface for mobile communication (Global System for Mobile communications (GSM), Global Positioning System (GPS), long-term evolution (LTE) portable radio standard 4G, Wi-Fi), which provides the tracked values to a centralized geographical information system (GIS) database server, an electronic processing unit, a micro-controller to processes the information that also acts on the interface between the GPS, a mobile communication unit, and some amount of memory for saving GPS values in case of mobile-free zones or to intelligently store information about the vehicle's sensor data.
- *Transmission Control Unit (TMCU)*: Controls electronic automatic transmissions. A TMCU generally uses sensors from the vehicle as well as data provided by the ECU to calculate how and when to change gears in the vehicle for optimum performance, fuel economy, and shift quality. In some applications, the TMCU and the ECU are combined into a single unit as a Powertrain Control Module (PCM).

Fig. 6.18 V-diagram used in the ECU design cycle



Managing the increasing complexity and number of ECUs in a vehicle has become a key challenge for automakers and OEMs as modern vehicles have up to 100 ECUs. Moreover, embedded software in ECUs continues to increase in line count, complexity, and sophistication and has reached more than seven million lines of software code today, requiring specific concepts, methods, techniques, and tools for testing security in automotive software in ECUs. Once the code containing the control algorithm is downloaded to the ECU, performance testing of the ECU can be done under extreme conditions, which cannot be achieved in the real world, by performing hardware-in-the-loop (HIL) simulation. In this step, the actual ECU is tested by simulating an engine using the created engine model. In HIL, the software model of the engine is downloaded to real-time hardware; and the appropriate input/output (I/O) interfaces are provided. These I/Os are then connected to the ECU under test. Then various engine conditions can be simulated; and the ECU can be tested to its limits, above and beyond any real-life capabilities of a real engine. The documentation can be done by using any word processing or spreadsheet application. The design process follows the V-model, shown in Fig. 6.18 (ni-com 2009).

6.2.1 Vehicle Network Technologies and Cybersecurity

The software-intensive automotive ECUs control two or more of the electrical systems or subsystems in an automotive vehicle using a bus system for communication. The main forces driving the development of vehicle network technologies have been the advances made in ECU components, governmental regulations imposed, and consumer requests. Among the best known and most common are the CAN, the local interconnect network (LIN) designed for controlling the vehicle ECUs, and the media-oriented systems transport (MOST) designed for all kinds of vehicle multimedia applications, such as audio, video, navigation, and communication systems. Thus, developing new vehicle models increases the number of microcontrollers used which results in an ever-increasing number of nodes within the vehicle network and in turn increase the vulnerability. This makes, cybersecurity of vehicle network technologies is an important factor in the prevention of cyberattacks in vehicles

because ECUs in a vehicle typically receive their input from sensors that send data which is used for computation. Various actors are used to enforce the action determined by the ECUs. The ECUs need to exchange data among themselves during normal operation of the vehicle. For example, the ECM will inform the TMCU of the engine speed; and the TMCU will inform other ECUs when a gear shift occurs. This exchange of data needs to be done quickly, reliably, and securely over the vehicle network. Thus, attacks by adversaries can use the CAN bus to disrupt vehicle control systems in several areas, such as (Kao and Marculescu 2006):

- *Airbag Control System:* Adversaries emulate the behavior of a fully functional airbag control system, including a successful startup check. This code could be included in the network if the airbag system was broken or had been removed or has been electronically deactivated by the attacker.
- *Central Gateway:* Adversaries attack a gateway ECU by implementing basic filtering functions with regard to the internal vehicle communication, forcing a degree of separation between internal and external networks. An implementation flaw of the gateway ECU could be identified and exploited inducing the gateway ECU to pass on arbitrary internal CAN messages to the outside.
- *Warning Light:* Under regular operation, a light flashes in the event of unauthorized opening of a door. Adversaries attack by turning the light off and ensuring it stays off just by sending CAN commands to the comfort subnetwork.
- *Window Lift:* An adversarial attack was conducted in a simulation environment using CAN. In this test, only a few lines of malicious code were added to an arbitrary ECU in the simulated comfort CAN subnetwork. This code deploys when a predefined condition is met; in this case study, it deployed when the vehicle's speed rose over 200 km/h ($\approx 124\text{mph}$). Then, a window opened and would not close until the end of the window lift attack. Similar results were demonstrated in a corresponding physical environment.

In each of these cases, apart from the central gateway attack, adversarial attackers required physical access to the internal CAN network and the ability to insert malicious code into ECUs. In the case of the mentioned central gateway attack, the adversary required the ability to insert malicious code through the OBD interface. These attacks can be analyzed using the US Computer Emergency Readiness Team (CERT) taxonomy (Cebula and Young 2010; Cichonsky et al. 2012) and prevented using the set of short-term countermeasures suggested. These include intrusion detection and facilitating post-incident analysis through proactive forensics support. The OBD connector offers direct access to all CAN buses through a physical port within the vehicle cabin. The interface and messages are standardized which means there is a plethora of cheap, easily available scan tools for the OBD port. Scan tools available are:

- Full-featured versions with built-in software, user interfaces, etc.
- Dumb tools that must interface with another computing platform, such as a phone or a conventional personal computer (PC).

At the Black Hat Asia Security Conference 2015 in Singapore, a programmable device called CANtact was introduced which represents a physical connection between a vehicle's OBD port and a computer's USB port which runs on open-source software. A Python library makes it easy to interact with CAN networks (Akella et al. 2010). CAN frames can be easily encoded as Python objects and sent, received, logged, and inspected. CAN-based standardized diagnosis protocols are supported, such as OBD-II and Unified Diagnostic Services (UDS) ISO 14429, among others. UDS allows the reading and writing of arbitrary memory into a vehicle, making hacking of vehicles much easier as it only requires physical access to the OBD.

With regard to the CAN bus, its protocol contains no direct support for secure communications. Retrofitting the protocol with security mechanisms poses several challenges given the limited data rates available and potential for bus utilization to increase significantly. In (Lin and Sangiovanni-Vincentelli 2012), a security mechanism is described which keeps the bus utilization as low as possible. Through experimental evaluation, it has been shown that the security mechanism can achieve high security levels while keeping communication overhead, e.g., bus load and message latency, at reasonable levels. In another paper (Lin et al. 2013), an integrated mixed integer linear programming formulation was proposed to address safety and security requirements during the explanation of the mapping from the functional model to the CAN-based architecture platform. The mapping design space includes the allocation of tasks to ECUs, the packaging of signals into messages, the sharing of message authentication codes (MACs) among multiple receiving ECUs, and priority classifications of tasks and messages. The security constraints are set to prevent direct and indirect cyberattacks on the MACs. The safety constraints are defined on the end-to-end latency deadlines for safety-critical paths.

In a master's thesis (Bruton 2014), securing CAN bus communication has been investigated by analyzing software-based cryptographic methods that focus on message authentication where the challenges of dealing with a small packet frame is considerable. The scope of the thesis was to investigate the effects using cryptographic approaches for both encryption to provide message content confidentiality, and authentication, to improve security in CAN bus communications without incurring unacceptable delays in communications and without the need for additional hardware resources. With regard to hard real-time constraints of the CAN bus, symmetric encryption techniques are chosen, such as AES which are based on a design principle known as a substitution-permutation network, where a combination of both substitution and permutation is applied to the message which is fast in both software and hardware. Authentication can be achieved on the CAN bus using hash functions for message authentication codes, assuming the hash function employed therein is fast.

In general, security for networked ECUs is an important issue for maintaining the integrity and privacy of data, while also improving network resiliency to cyber physical attacks, which is mostly based on security threats, such as manipulating the system at the information system level and within its surroundings, and others. To ensure security for vehicle CPSs for these types of security threats, several security objectives need to be achieved, as shown in Table 6.9.

Table 6.9 Security objectives and their impacts

Security Objective	Impact
Authenticity	Important proof for securing distributed CPSs and preventing users and devices from impersonating another system or component. Ensures that data, transactions, and communications of a CPS are genuine. Requires that the CPSs can validate that they are who they claim to be and thus avoid intrusion by the means of cyberattacks. This prevents unauthorized access to the sensor nodes or communication network while imposing and enforcing proper restrictions on what authenticated systems and components are permitted to do
Availability	Refers to the ability of always being accessible and usable while a lack of accessibility may cause a denial of service which may result in irreparable damages or malfunction of the system or components around it
Confidentiality	Refers to the capacity of a CPS to prevent the disclosure of information to unauthorized individuals or systems as part of a cyberattack. A CPS must prevent cyberattacks from interfering with the state of the CPS by eavesdropping on the communication channels between the sensor nodes and the controller, as well as between the controller and the actuator nodes
Integrity	Refers to data or resources that cannot be modified without authorization. Integrity is violated if a cyberattacker accidentally or with malicious intent modifies or deletes important data such that the receiving CPS or actuator node receives false data and follows this data believing it to be true
Reliability	Fundamental requirement of a CPS, i.e., a system featuring a tight combination of, and coordination between, the CPS's computational and physical elements. A CPS is designed to process large amounts of data, employ software as a system component, run online continuously, and retain an operator-in-the-loop (OITL) because of human judgement and accountability requirements for safety-critical systems. Based on data-centric runtime monitoring, reliability of a CPS can be automatically evaluated with regard to data detection of abnormal input and output through data quality analysis. As a result, alerts can be sent to the OITL, who can then take actions and make changes to the system based on these alerts in order to achieve minimal system downtime and higher system reliability (Falliere et al. 2011)
Robustness	System property describing the degree to which a system operates correctly in the presence of a disturbance, such as unforeseen or erroneous inputs (Eisenhauer et al. 2006). The notion of robustness was inspired by notions of input-output stability as developed in control theory (Lamport 2005). Moreover, it has been shown that the proposed notion of robustness has to meet two intuitive goals: (1) bounded disturbances lead to bounded deviations from nominal behavior, and (2) the effect of a sporadic disturbance disappears in many finite steps. The proposed notion of robustness for a CPS can be verified in pseudo-polynomial time. The synthesis problem, consisting of designing a controller to enforce robustness, can also be solved in pseudo-polynomial time (Lamport 2005)
Trustworthiness	Estimating the feasible impact of a cyberattack requires evaluation of the system's dependency on its cyber infrastructure and its ability to tolerate potential failure. Further exploration of the cyber-physical relationships within the system and specific or possible attack vectors is necessary to determine the adequacy of cybersecurity efforts (Tang and McMillin 2008)

6.2.2 Cyberattack Taxonomy

Cyberattacks are more difficult to detect and prevent in ECUs and cyber-physical systems compared to cyberattacks on the Internet (Yuzhe et al. 2013). To evade detection, cyberattacks may apply multiple stages to gain access to a vehicle mission-critical system. Moreover, cyberattacks over the years have become both increasingly numerous and sophisticated. This calls for their analysis and categorization, assistance in combating new cyberattacks, and improvement of computer and network security, which necessitates cyberattack taxonomy.

The term taxonomy, in general, is derived from the Greek *taxis*, meaning arrangement or division, and *nomos*, meaning law. In this regard, it is the science of classification according to a predetermined system, with the resulting catalog used to provide a conceptual framework for discussion, analysis, or information retrieval. In theory, the development of a good taxonomy takes into account the importance of separating elements of a group (taxon) into subgroups (taxa) that are mutually exclusive, unambiguous, and, taken together, include all possibilities. Furthermore, as a good practice, the taxonomy should be simple, easy to remember, and easy to use, as mentioned in (URL7 2016). As reported in Kjaerland (2005), taxonomy of cyber-based intrusions can be proposed as it relates to computer crime profiling and highlighting cyberattackers and the attacked systems. Thus, cyberattacks were analyzed using facet theory, which offers a set of principles for guiding research design, has a companion set of multivariate statistical procedures to analyze data, and establishes a framework within which to construct theories (Brown 1985). The analysis included multidimensional scaling with *R*, a programming language and software environment for statistical computing and graphics, which provided functions for both classical and nonmetric multidimensional scaling, with the method of operation, target, source, and impact. Each facet contained a number of elements with an exhaustive description. Hence, taxonomy is proposed to consist of at least four dimensions, providing a holistic taxonomy to deal with the inherent problems in computer and network cyberattacks, as shown in Table 6.10. Within each dimension, various levels of information are provided showing the characteristics and consequences of cyberattacks.

From Table 6.10, it can be deduced that taxonomy should fulfill the following requirements, listed in Table 6.11 (Hansman and Hunt 2005).

As mentioned in Hansman and Hunt (2005), work is needed to improve the classification of blended attacks, which is a limitation within their taxonomy. Another limitation is the lack of vulnerability information which prohibits capturing information to aid in protecting a system from attacks.

An attack-centric taxonomy called Validation Exposure Randomness Deallocation Improper Conditions Taxonomy (VERDICT) has been proposed in (Lough 2001), which focuses on four major causes of security errors:

Table 6.10 Classification, characteristics, and consequences of cyber criminal attacks

Dimension	Cyberattack description
1st	Classifies the cyberattack into a cyberattack class based on the attack vector and the main behavior of the cyberattack. If there is no attack vector, the cyber criminal attack is classified into the closest category
2nd	Classifies the cyberattack targets. Targets can be classified down to very specific targets or a class of targets
3rd	Covers vulnerabilities and exploits, if they exist, used by the cyberattack. They do not have a structured classification due to the infinite number of possible vulnerabilities and exploits
4th	Takes into account the possibility for a cyberattack to have a payload or effect beyond itself. In many cases, a cyberattack will clearly be a certain kind of cyberattack; but yet it will have a payload or cause an effect that is different

Table 6.11 Requirements to develop a pragmatic taxonomy

Requirement	Aims to take into account
Accepted	Taxonomy is structured such that it can become generally approved
Comprehensible	Taxonomy is understood by those who are in the security field, as well as those who only have an interest in it
Completeness	Taxonomy is complete/exhaustive. It should account for all possible attacks and provide categories accordingly While it is hard to prove a taxonomy is complete or exhaustive, it can be justified through the successful categorization of actual attacks
Determinism	Classification procedure is clearly defined
Mutually Exclusive	Taxonomy categorizes each attack into, at most, one category
Repeatable	Classifications are repeatable
Terminology	Complies with established security terminology
Terms	Should be well defined. There should be no confusion as to what a term means
Unambiguous	Each category of the taxonomy must be clearly defined such that there is no ambiguity with respect to an attack's classification
Useful	Taxonomy is used in the security industry and by incident response teams, in particular

- *Improper Deallocation:* Improper destruction of information, or residuals of data, which also includes dumpster diving.
- *Improper Exposure:* Involves improper exposure of information that could be used directly or indirectly for the exploitation of vulnerability.
- *Improper Randomness:* Deals with the fundamentals of cryptography and the improper usage of randomness.
- *Improper Validation:* Refers to improperly validating unconstrained data, which also includes physical security.

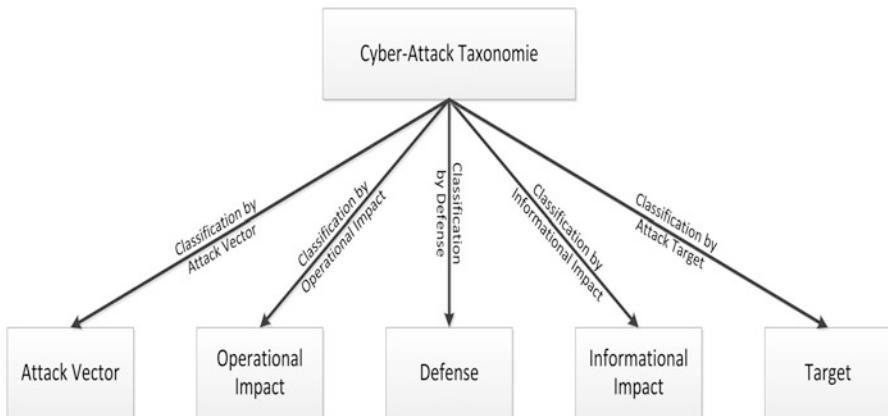


Fig. 6.19 Structure of the cyber attack taxonomy AVOIDIT

In Hansman and Hunt (2005), it is mentioned that the taxonomy described in Lough (2001) lacks pertinent information that would be beneficial for knowledge bodies, such as a CERT, to classify day-to-day attacks and ensure advisories, a taxonomy that can be used as a tool to assist in the identification of all applicable operational cybersecurity risks. Furthermore, the taxonomy described in Lough (2001) lacks classification based on the type of attack, such as Trojan, virus, worm, and others.

The cyberattack taxonomy Attack Vector, Operational Impact, Defense, Information Impact, and Target (AVOIDIT), introduced in (Guttmann and Roback 1995), provides, through application, a knowledge repository used by a defender to classify vulnerabilities that a cyberattacker can use, as shown in Fig. 6.19. AVOIDIT provides details on each cyberattack classification and how a variety of cyberattacks are represented in each category.

The general scheme shown in Fig. 6.19 is expanded in Fig. 6.20, which provides details on each attack classification and how a variety of attacks are represented in each category (Simmons et al. 2014). AVOIDIT could be extended to include new categories within each classification and will provide a cyber criminal attack defender with the appropriate information to make a clear decision in defending against cyberattacks. Advanced approaches to defending against attacks will become available and provide an extensible taxonomy for capturing new defenses. In future work, building a game-theoretic defense strategy, the applicability of AVOIDIT in determining the action space of a cyberattacker will be investigated (Shiva et al. 2010).

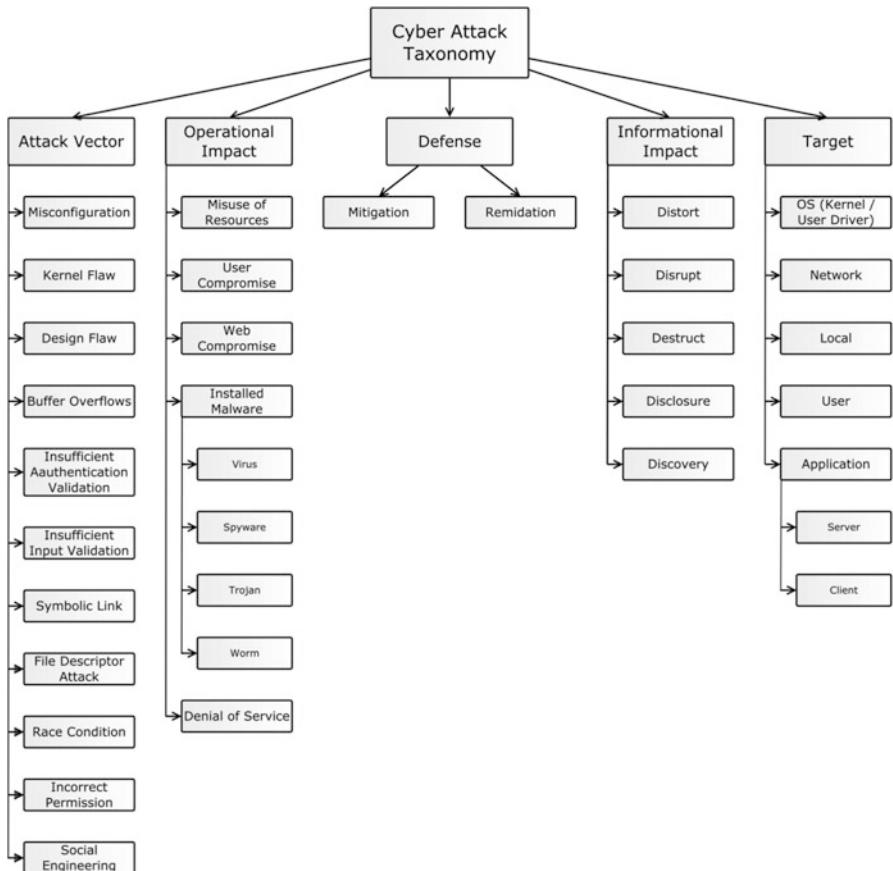


Fig. 6.20 Architecture of the AVOIDIT cyber attack taxonomy

6.3 Hacking and Automotive Attack Surfaces and Vulnerabilities

6.3.1 Hacking

Hacking is a very real vehicle security risk as evidenced by the increasing number of cyberattacks on systems and data. Hacking means that adversaries target trusted security controls as a means of facilitating later cyberattacks whereby hacking adversaries (hackers) may threaten information security on multiple levels simultaneously. Hackers may attack (Shimeall and Spring 2014):

- *Data* used in essential business processes, including compromise, imitation, or redirection of data sources, using websites that closely imitate institutions to obtain authentication information used in later frauds.
- *Individual hosts*, exploiting weaknesses in the operating system or in the application software
- *Users*, either as malicious insiders or as malicious outsiders
- *Networks*, via remote access methods or by exploiting the trust within networks to propagate from an initial intrusion point of compromise

Hackers employ the following strategies:

- *Direct Physical Access*: In this, the simplest hacking attack strategy, the hacker strikes against the target from an intrusion point, without intermediate or third-party hosts involved except for normal traffic routing. This strategy is applied in cyberattacks where the intrusion point is of little value to the hacker or the probability of backtracking is very low.
- *Progressive Access*: The hacking adversary uses a series of intermediate hosts between the intrusion point and the target, each of which is compromised using the same set of exploits.
- *Mass Hacking*: The hacker compromises a group of third-party hosts and uses all of them at once against the targeted host.
- *Misdirection Access*: Generates traffic to confuse or distract the defenders in dealing with their direct cyberattack (Shimeall and Spring 2014).

6.3.2 Automotive Attack Surfaces and Vulnerabilities

As described in Sect. 6.4.2, the number of electronic components in modern vehicles has increased rapidly and continuously during recent years. This has resulted in millions of lines of code executing on several heterogeneous embedded computers with huge connectivity provided by automotive bus systems such as CAN. On one hand, many sensors and actuators have been developed and embedded in vehicles to make passengers feel safer. On the other hand, more advanced entertainment and navigation systems have made their way into vehicles to make traveling more comfortable. Although this technological progress has generated significant benefits in terms of efficiency and cost, it has also created more opportunities for new attack surfaces which increase the vulnerability of vehicles to cyberattacks.

With regard to the Bluetooth® network protocol used in vehicles, an overview of the security architecture and security modes of the Bluetooth® protocol, as well as the vulnerabilities that Bluetooth® networks face, is reported in (Johnson 2010). Three of the most crucial categories correspond to the confidentiality, integrity, and availability (CIA) triad, a model designed to guide policies for information security with regard to threats of:

- Denial of service
- Disclosure of unauthorized information
- Integrity of information

The triad is sometimes referred to as the availability, integrity, and confidentiality (AIC) triad to avoid confusion with the initialism for the US Central Intelligence Agency (CIA).

In the context of the CIA triad:

- *Confidentiality* is a set of rules that limits access to information and is roughly equivalent to privacy.
- *Integrity* is the assurance that the information is trustworthy and accurate.
- *Availability* is a guarantee of reliable access to the information by authorized people.

Availability is best ensured by rigorously maintaining all hardware, performing hardware repairs immediately when needed, and maintaining a correctly functioning operating system environment that is free of software conflicts. However, the powerful directional antennas in Bluetooth-based networks can be used to considerably increase the scanning, eavesdropping, and attack range of almost any kind of Bluetooth® attack.

In Cárdenas et al. (2011), Bluetooth® is considered to be one of the biggest and most viable cyber attack surfaces on modern vehicles, due to the complexity of its protocol and underlying data. Additionally, Bluetooth® has become ubiquitous within the vehicle domain, giving cyberattackers a very reliable intrusion point to test attack scenarios.

In Cárdenas et al. (2008), Bluetooth® capabilities built into test vehicles' telematics units have been investigated. Access to the telematics ECU's UNIX®-like operating system was gained through reverse engineering, and the particular program responsible for handling Bluetooth® functionality was identified. It was verified that the ECU's operating system contained a copy of a popular embedded implementation of the Bluetooth® protocol stack along with a sample hands-free application and a custom-built interface. The interface contained vulnerability that allowed buffer overflow attacks to be mounted by any paired Bluetooth® device and allowed arbitrary code to be executed on the telematic unit. Adversaries use buffer overflows to corrupt the execution stack. By sending carefully crafted input to an application, a cyberattacker can cause the application to execute arbitrary code, possibly taking over the mission-critical cyber-physical system's functionality. Buffer overflow attacks generally rely on two techniques, usually in combination:

- Having the operating system mishandle data types
- Writing data to particular memory addresses

This means that strongly typed programming languages and environments that disallow direct memory access usually prevent buffer overflows from happening. Available techniques to prevent buffer overflows include:

- Code auditing
- Compiler tools, such as StackShield, StackGuard, and Libsafe, etc.
- Nonexecutable stacks which are supported by many operating systems
- Patches with regard to bug reports relating to applications upon which the code is dependent

The US federal government mandated the OBD-II port, under the dashboard, which provides a direct and standard hard-wired communication link to ECUs through which access is allowed to read and reset a vehicle's fault codes. Also, access to information from various units through the diagnostic connector is possible so that all systems can be diagnosed and programmed. User-upgradable subsystems, such as audio players, are attached to these same networks by a variety of short-range wireless devices, such as wireless tire pressure monitoring system (TPMS), as well as Bluetooth® devices and more, which also represent new partial attack surfaces. However, vehicles equipped with driving aid systems, such as an electronic stability program (ESP) or adaptive cruise control (ACC), allow deep interventions in the driving behavior of the vehicle, too. Furthermore, electronic drive-by-wire vehicle control systems fully depend on the underlying automotive data networks. Moreover, vehicle communication networks assure safety against technical interference; but they are mostly unprotected against malicious cyberattacks. This increasing coupling of unsecured automotive components together with new multimedia networks, such as MOST, and the integration of wireless interfaces, such as GSM or Bluetooth, causes various additional security risks in the context of attack surface intrusion points. Summing up, it can be stated that today's vehicles are pervasively computerized with regard to their increasingly sophisticated services and embedded communication features and, hence, potentially much more vulnerable to cyberattacks than in the past. As a result, the attack surface intrusion points must be multifaceted, as shown in Fig. 6.21, including safety-critical components such as brakes, engine, transmission, and others.

Cyberattacks against vehicle safety-critical systems result in physical control of the various components of the vehicle and access to the internal vehicle network. This allows the adversary to inject code into the vehicle networks to directly or indirectly control the desired ECUs. Researchers from the University of California San Diego and the University of Washington were able to execute code remotely on a telematics unit of a vehicle by exploiting vulnerability in the Bluetooth stack on an ECU and by separately compromising a cellular modem.

Vehicle manufacturers also provide some kind of external digital multimedia port, typically a USB port or an iPod/iPhone docking port, allowing users to control their vehicles' media systems using their personal audio players or phones. Consequently, an adversary might deliver malicious code by encoding it onto a CD or a song file along with using social engineering to convince the user to play

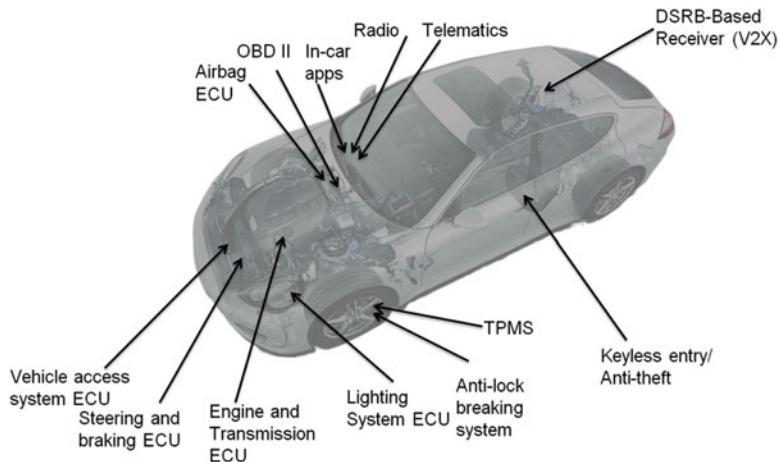


Fig. 6.21 Anatomy of the 15 most hackable and exposed attack surface intrusion points on a next-generation vehicle, modified after (Intel Security 2015)

it. Alternatively, a user's phone or iPod might be compromised out of band and malicious software installed onto it that attacks the vehicle's media system when connected (Checkoway et al. 2011).

As reported in Valasek and Miller (2014), a compromised ECU cannot control the safety features of a vehicle. This ECU's task is typically only related to receiving and processing radio signals. Therefore, a cyber physical attack usually requires a second step which involves injecting malicious code into the internal vehicle network in an attempt to communicate with safety-critical ECUs, such as those responsible for steering, braking, and acceleration. In some vehicles, this may be trivial; but in many designs, the ECU which was compromised remotely will not be able to directly send messages to these safety-critical ECUs. In this case, the cyberattacker will have to somehow get messages bridged from the network of the compromised ECUs to the network where the target ECU resides. This might require tricking the gateway ECU or compromising it outright.

The researchers from the University of California San Diego and the University of Washington (Checkoway et al. 2011) demonstrated a way to compromise the bridge ECU in their vehicle to get from the less privileged CAN network to the one containing the ECU in charge of braking. After the attacker has wirelessly compromised an ECU and acquired the ability to send malicious code to a desired target ECU, the attacker may communicate with safety-critical ECUs, making them behave in some way that compromises vehicle safety. This involves reverse engineering the messages on the network and figuring out the exact format to perform some physical action.

Since each manufacturer, and perhaps each model and even each year, use different data in the messages on the bus, the message reverse engineering process requires a large amount of work and is manufacturer specific. For example, the

messages to lock the brakes on one manufacturer's vehicle likely won't work on a vehicle from a different manufacturer. Furthermore, some ECUs only listen to certain messages and may have safety features built into them, such as not responding to certain messages while the vehicle is in motion (Valasek and Miller 2014). Thus, it is important to know, without a detailed investigation, whether it is possible to affect cyber physical vehicle features through malicious software injection since it essentially relies on the implementation of the ECUs. Therefore, Valasek and Miller (2014) report an approach similar to measuring the remote attack surface. For each vehicle, they list the computer-controlled features. In the Toyota Prius, for example, the collision prevention system is designed to stop the vehicle when certain CAN messages are received. This is a safety feature and can be exploited. So while all vehicles may or may not be vulnerable to safety-critical actions through CAN data injection (Valasek and Miller 2014), it can be assumed that those with advanced computer-controlled features are more susceptible since they are designed to take physical actions based on data received on the internal network.

In the case of telematics services, value-added automatic features, such as those listed below, are provided over a long-range wireless link.

- Crash response
- Remote diagnostics
- Stolen vehicle recovery

For this purpose, telematic systems integrate internal automotive subsystems with a remote command center via a wide-area cellular network connection.

Some service providers have taken this concept even further by proposing a car-as-a-platform (CaaP) model for third-party development and applications related to in-car connected platforms, offering a selection of features in connected vehicles (cars) with a special focus on entertainment apps and safety-management features. Entertainment is one of the most popular features available for the connected car. Entertainment features include integrations with apps, such as Pandora®, Yelp®, Facebook®, and others. Hughes Telematics has described plans for developing an app store for automotive applications (Mollmann 2009), while Ford recently announced that it will open its SYNC® telematics system as a platform for third-party applications (Goodwin 2009). SYNC®, an automaker-installed integrated in-vehicle communications and entertainment system, allows users to:

- Control music
- Make hands-free telephone calls
- Perform other functions with the use of voice commands

The system consists of applications and user interfaces developed by Ford and other third-party developers. With regard to a cellular modem built in to the vehicle, Ford is planning to execute OTA software updates just as Tesla already does. Ford

can already do OTA updates to SYNC3 using Wi-Fi when the vehicle is connected at home. Other telematics systems, such as General Motors' OnStar®, provide value-added features, such as:

- Automatic crash response
- Remote diagnostics
- Stolen vehicle recovery over a long-range wireless link

To do so, these telematics systems integrate internal automotive subsystems with a remote command center via a wide-area cellular connection.

Furthermore, there are many proposed V2V and V2X communications systems (CAMP09 2008; CAMP10 2008; CAMP05 2005; VTTI 2007) that will broaden the attack surface intrusion points further. More possible attack surfaces of connected vehicles are given in Fig. 6.21. Overall, these trends suggest that a wide range of attack vectors will be available by which an adversary might compromise a vehicle's electric/digital components and gain access to internal vehicular networks with unknown consequences. The two kinds of attack vectors by which adversaries might gain access to a vehicle's internal networks are, as previously mentioned, the physical access and the numerous wireless interfaces embedded in today's vehicles. These interfaces accept outside input through which it is possible to remotely compromise key ECUs via externally facing vulnerabilities, remotely control a vehicle over the Internet, and others. With regard to physical access, an adversary can, with even momentary access to the vehicle, insert a malicious component into a vehicle's internal network via the ubiquitous OBD-II port (Kosher et al. 2010).

The next step proposed by some service providers is a connected-car-as-a-digital-platform (CCaaDP) model. The vehicle itself is a connected platform that enables multiple protocols to communicate with each other and connects to the cloud through the user's mobile cellular service and hardware. The current models feature multiple communication systems that connect the following to the drivers display.

- Airbag
- Camera/radar systems
- Driver assist
- Engine
- Safety systems
- Tire pressure

There is also a network that connects the passenger area to the information system along with entertainment control, which will open an additional attack surface. These systems are moving from just wired systems, such as CAN, MOST, and Flexray™, to more standard systems such as Ethernet, a new wired solution which can support low weight, unshielded cable capable of 100 Mbps in full duplex mode for connectivity. This wired system is being sought by many automakers to allow the vehicle to be a platform core, such as a data center, with one in-platform interconnect

system and the edge of the network to be wireless or a USB interface (Chatterjee 2012). This single network configuration simplifies the communication options by creating a single protocol for the data transfer. This allows for industry qualification, such as the standards in the global automotive industry, including TS16949 compliance/ISO 9001 certification, in-car EMC performance, and AEC-Q100, to be addressed in one pass, thus offering the automotive One-Pair Ethernet Alliance Special Interest Group (OPENSIG). OPENSIG is promoting the switch to in-car Ethernet.

Hence, the connected vehicle (car) is driving the automotive semiconductor market. Factory-installed networking connections are increasing due to integration of systems with sensor networks that are not accessible post vehicle assembly. The automaker-installed rate may be as high as 60% in the near future. Costs of these systems have been dramatically reduced, and they are available both in mass market vehicles and luxury applications, which also opens new attack surface intrusion points, resulting in increased vulnerability.

Therefore, the following questions need to be answered as they relate to security in vehicle CPSs:

- Which methods and tools can be used for security testing and evaluation in the automotive industry?
 - Many methods and tools are available for vehicle security testing and evaluation; however, these methods alone are not able to address all security problems that might arise from the implementation phase. Hence, an evaluation methodology is needed, to determine which method(s) could be used for security testing and which could address the security problems arising from the implementation phase in vehicle software (Chalkias et al. 2009).
- How can various methods be combined to systematically perform security testing?
 - Different methods are required for security testing followed by ad hoc approaches. A single method may always lack a foolproof strategy for eliminating all kinds of security problems. The solution is, therefore, to combine the advantages of other methods into the presence of one. Therefore, a systematic approach to combining various methods can be beneficial. By systematic security testing, it may also be possible to prove that a potential vulnerability can be exploited (Chalkias et al. 2009).

Security testing methods available in the automotive domain which prevent intrusion points for cyberattacks are:

- *Functional Security Testing*: Investigates functional correctness and the robustness testing of security functionalities (Chalkias et al. 2009). For example, cryptographic algorithms to be implemented should be checked for their correctness. Implementations of cryptographic algorithms are often tested with official test vectors. Developers mostly rely on specifications and official test vectors during code development. Some doors of opportunity remain for cyberattackers

to exploit potential vulnerabilities that may arise from other sorts of random test vectors. These kinds of security vulnerabilities are missed by functional security testing teams. Adhering to MISRA C/MISRA C++ safety coding standards can reduce the number of such potential vulnerabilities in software. MISRA guidelines have been widely adopted to ensure the quality of safety- and security-critical software in automotive, aerospace, defense, industrial, medical, and rail applications. By following MISRA rules, developers can be assured of using the most stringent software coding guidelines to mitigate liability and risk in software applications on which human lives depend, and to avoid coding practices that can introduce security vulnerabilities (URL8 2016).

- *Fuzzing and Penetration Testing:* Vehicle CPSs with malformed inputs that might be able to uncover unsafe weaknesses and vulnerabilities (Xiao et al. 2008) with regard to possible attacks through external ports and physical devices can be tested. The available code for automotive software is not open source. Reverse engineering is currently used to retrieve binary code, and all types of security tests are performed with the help of third-party debuggers, for example, OllyDbg, a 32-bit assembler level analyzing debugger for Microsoft Windows, and IDA Pro, a Windows, Linux®, or Mac OS® X-hosted multiprocessor disassembler and debugger, and others. Penetration testing investigates possible attacks through external ports and physical devices and is a sophisticated way of testing the whole system by a security tester with his/her knowledge of security testing (Chalkias et al. 2009). It involves testing hardware and software with single a or a combination of various security testing methods such as:
 - Code review
 - Manual inspection
 - Static analysis
- *Vulnerability Scanning:* A test system with a known set of vulnerabilities that could be either unsafe functions or unsafe configurations (Chalkias et al. 2009). The Open Vulnerability Assessment System (OpenVAS) is a framework of several services and tools offering a comprehensive and powerful vulnerability scanning and vulnerability management solution that scans for open ports in automotive IT and software. OpenVAS products are free software. Most components are licensed under the GNU General Public License (GNU GPL). The architecture of OpenVAS is shown in Fig. 6.22 (URL9 2016).

The most essential blocks shown in Fig. 6.22 have the following meanings:

- *OpenVAS Command Line Tool:* Contains the command line tool, Open VAS management protocol, which allows the creation of batch processes to drive the OpenVAS manager. It runs on Windows, Linux, etc. and is a plugin for Nagios®. Nagios can be considered to be an industry standard for monitoring IT infrastructures (www.nagios.org).
- *Greenbone Security Assistant (GSA):* Is a client for OpenVAS management protocol and OpenVAS administration protocol and serves HTTP and HTTPS.

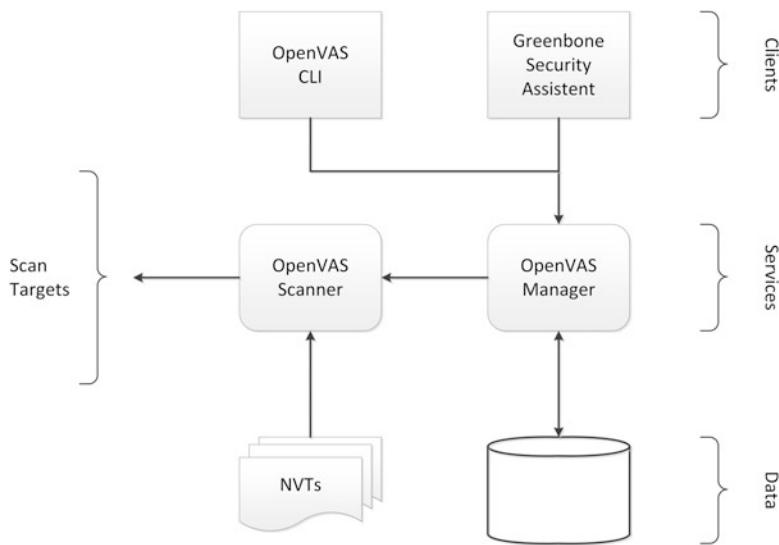


Fig. 6.22 Open Vulnerability Assessment System (OpenVAS) framework, modified after (URL9 2016)

- *OpenVAS Scanner:* Uses the OpenVAS transfer protocol on the server side and the OpenVAS manager on the client side.
- *Network Vulnerability Tests (NVTs):* Work on the detection of certain product vulnerability evaluations. The actual detection NVTs should result in a Common Platform Enumeration Code (CPEC) code for the product.

Finally, in Fig. 6.23, risks of security attacks are summarized with regard to attack vectors which represent the path or means by which a hacker can gain access to CPSs and communication networks to intrude a malicious outcome. The attacker's goal is to exploit system or component vulnerabilities, including immediate and long-term risks. In general, it can be stated that recent hacking attacks will enable a steep rise in the demand for automotive security solutions that repel malware intrusion. To some extent, firewalls and antivirus software can block attack vectors; but no intrusion prevention method today is totally attack proof given that a defense method that is effective today may not remain so for long. Hackers are constantly updating attack vectors and seeking new ones by looking to gain unauthorized access to vehicle CPSs and communication networks.

Immediate and long-term risk as a function of attack vectors, attack goals, and the vulnerable system are shown in comparison in Fig. 6.23 is:

- *Man-in-the-Middle Attack (MITA):* Involves an attacker positioning himself between the two nodes A and B which will communicate without the knowledge of each other. Hence, the man-in-the-middle (node C) makes node A believing that he is node B. Thereafter, he makes node B believing that he is node A. In this way node C handles all communication between nodes A and B without revealing this fact, and he can copy, alter, or compromise any messages sent.

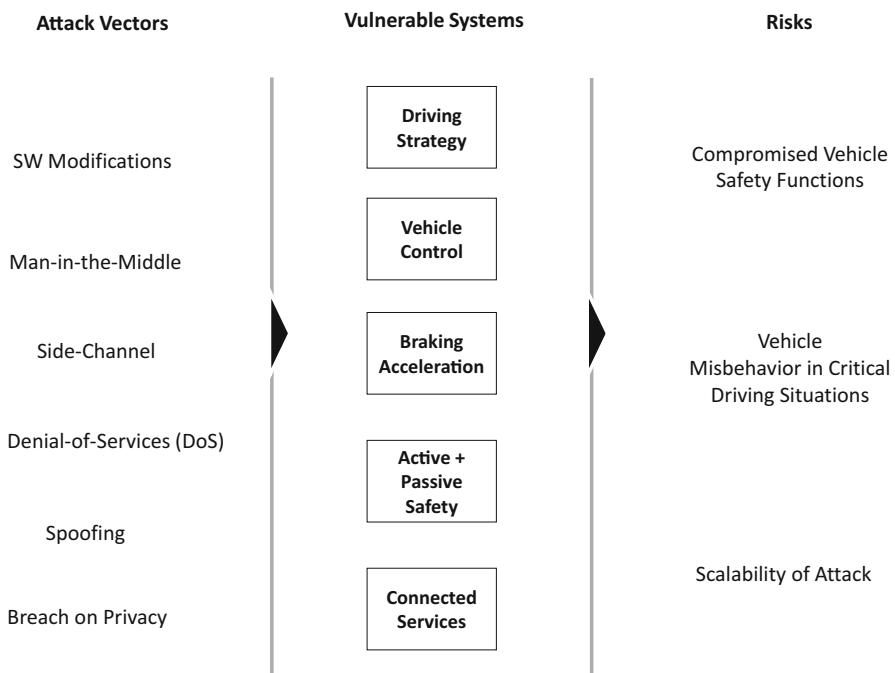


Fig. 6.23 Risks of cybersecurity w.r.t. cyberattacks in vehicles

- *Side Channel Attack (SCA):* Attack based on information gained from the physical implementation of a cryptosystem, rather than brute force or theoretical weaknesses in the algorithms. Some side-channel attacks require technical knowledge of the internal operation of the system on which the cryptography is implemented, although others such as differential power analysis (DPA) are effective as black box attacks.
- *Brute Force Attack:* Refers to attempts to obtain logon credentials by guessing usernames and passwords. Some risks exist for services that allow remote access, brute force attackers use password guessing tools and scripts containing default password databases, dictionaries, or rainbow tables that contain commonly used passwords and may try all combinations of a character set. Brute force attacks are typically one-by-one attacks executed by an expert attacker against selected targets (Johnson 2016).
- *Denial-of-Service (DOS) Attack:* Type of attack where the attackers attempt to prevent legitimate users from accessing the service. In a DoS attack, the attacker usually sends excessive messages asking the network or server to authenticate requests that have invalid return addresses. The network or server will not be able to find the return address of the attacker when sending the authentication approval, causing the server to wait before closing the connection. When server closes the connection, the attacker sends more authentication messages with invalid return addresses. Hence, the process of authentication and server wait will begin again, keeping the network or server busy (URL1 2017).

- *Spoofing*: Situation where a cyber attacker (or his program) successfully masquerades as another by falsifying data and hence gaining an illegitimate advantage.
- *Compromised Privacy*: A term used to classify matter, knowledge of which has, in whole or in part, passed to an unauthorized person or persons or which has been subject to risk of such passing.

Cyber attacks have changed. Broad, scattershot attacks designed for mischief have been replaced with advanced persistent threats focused on acquiring valuable data. Modern cyberattacks are often conducted across multiple vectors and stages. They have a plan to get in, signal back from the compromised network, and extract valuable data despite network security measures. Traditional defense-in-depth security measures, such as next-generation firewalls, antivirus, web gateways, and even newer sandbox technologies only look for the first move – the inbound attack. Advanced cyber attacks are designed to evade traditional network security.

6.4 Intrusion Detection and Prevention

6.4.1 Intrusion Detection

Intrusion detection can be defined (Heady et al. 1990) as any set of actions that attempts to compromise the CIA of a resource (see Sect. 6.3.2). Thus, it is a violation of the security constraints of the respective system. But as reported by Kumar and Spafford (1994), any definition of an intrusion is imprecise as security policy requirements do not always translate into a well-defined set of actions because intrusion detection is a methodology by which intrusions are detected. This methodology can be divided into two categories:

- *Anomaly Intrusion Detection*: System activities are observed which periodically generate profiles that capture their behavior, and older data is updated regularly to indicate its anomaly. As input audit records are processed, the observed system periodically generates a value indicative of its abnormality which may happen in a case where there is too much deviation from the regular profiles; and the intrusion detection system reports an intrusion. However, this can lead to false-positive alarms, depending on the conditioning or sensitivity of the intrusion detection system. False positives are events that are reported as malicious but in reality they are not.
 - *Advantage of Anomaly Intrusion Detection*: No predefined rules for detection of intrusions are required; hence new attacks can be detected.
 - *Disadvantages of Anomaly Intrusion Detection*: False positives can arise, leading to inconvenience for the users. Establishment of regular profile usage is required but is often hard to achieve.
- *Misuse Intrusion Detection*: Based on well-defined patterns of input events, assuming that the state transition of the system leads to an intruded state when exercised with the intrusion pattern, weaknesses in the system and application software can be exploited. The objective is to frame the intrusion detection

problem as a pattern-matching problem and to develop efficient algorithms for such matching. But simply specifying an intrusion pattern without the initial state specification is often insufficient to capture an intrusion scenario fully (Shieh and Gligor 1991).

Another classification scheme is based on the intrusion types presented in Denning (1987) and Smaha (1988) and is shown in Table 6.12, which introduces intrusion types, their characteristics, and detection possibilities.

Let A_1, A_2, \dots, A_n be n measures used to determine if an intrusion is occurring on a system at any given moment, whereby each A_i measures a different aspect of the system with

$$A_i = \begin{cases} 1 & \text{implying that the measure is anomalous} \\ 0 & \text{otherwise} \end{cases}$$

Table 6.12 Intrusion types and their detection

Intrusion type	Characteristics	Detection
Attempted break-in	Breaking into a system might generate an abnormally high rate of password failures with regard to a single account or the system as a whole	Atypical behavior profiles or violations of security constraints
Denial of service	An intruder able to monopolize a resource might have abnormally high activity with regard to the resource, while activity for all other users is abnormally low	Atypical use of system resources (e.g., networks)
Inference by legitimate user	A user attempting to obtain unauthorized data from a database through aggregation and inference might retrieve more records than usual	Atypical behavior profiles using I/O resources
Leakage by legitimate user	A user trying to leak sensitive documents might log into the system at unusual times or route data to remote printers not normally used	Atypical usage of I/O resources
Masquerading or successful break-in	A log into a system through an unauthorized account and password might have a different login time, location, or connection type from that of the account's legitimate user An intruder's behavior may differ considerably from that of the legitimate, e.g., a user using most of his time browsing through directories and executing system status commands whereas the legitimate user might edit, compile, or link programs	Atypical behavior profiles or violations of security constraints
Trojan horse	A program is substituted for a legitimate program	Atypical CPU time or I/O activity
Virus	May cause an increase in the frequency of executable files rewritten or storage used by executable files	Atypical CPU time or I/O activity

Let H be the hypothesis that the system is currently undergoing an intrusion. The reliability and sensitivity of each anomaly measure A_i is determined by

$$p(A_i = 1|H)$$

and

$$p(A_i = 1|/H).$$

The combined belief in H is

$$p(H|A_1, A_2, \dots, A_n) = p(A_1, A_2, \dots, A_n|H) \times \frac{p(H)}{p(A_1, A_2, \dots, A_n)}$$

which requires the joint probability distribution of the set of measures conditioned on H and $/H$.

In (Lunt et al. 1992), covariance matrices are used to account for the interrelationships between measures. If the measures A_1, A_2, \dots, A_n are represented by vector A , then the compound anomaly measure is determined by

$$A^T C^{-1} A$$

where C is the covariance matrix representing the dependence between each pair of anomaly measures A_i and A_j .

The foregoing methodology on intrusion detection is now broadened by the issue of intrusion prevention, the process of performing intrusion detection and attempting to stop the possible incidents detected. Therefore, the issue is one of introducing intrusion detection and prevention systems that are primarily focusing on identifying possible incidents, logging information about them, attempting to stop them, reporting them to security administrators, and documenting existing threats. Hence, intrusion detection and prevention have become a necessary issue to the security infrastructure of nearly every mission-critical system. The types of intrusion detection and prevention system (IDPS) techniques can be differentiated by the types of events that they monitor and the ways in which they are deployed, as shown in Table 6.13.

Securing automotive mission-critical components is a very important objective because these components are targeted by cyberattackers who want to gain access to the sensitive information of mission-critical components, system configurations, vulnerabilities, and others. Therefore, specific protective actions are of particular importance, such as encryption and other actions for transmitting data physically or logically over separate network components. This includes verifying that the components are working as desired, monitoring the components for security issues, performing regular vulnerability assessments, responding appropriately to vulnerabilities, and testing and deploying intrusion detection and prevention system updates. Resource constraints should also be taken into consideration by defining specialized sets of requirements for the following:

Table 6.13 Intrusion detection and prevention system types

IDPS Type	Characteristics
Host based	Monitoring characteristics of a single host and events occurring within that host for suspicious activity
Network based	Monitoring network traffic for particular network segments or devices and analyzing network and application protocol activity to identify suspicious activity
Network behavior analysis	Examines network traffic identifying threats that generate unusual traffic flows, such as distributed denial-of-service (DDoS) attacks, certain forms of malware, and policy violations (e.g., client system providing network services to other systems)
Wireless	Monitoring wireless network traffic and analyzing it to identify suspicious activity involving the wireless networking protocols themselves

- *Life Cycle Costs*: Initial and maintenance costs whereby the life cycle concept must be made in the context of achievement of the capability required to meet the operational conditions.
- *Management*: Design and implementation of reliability, interoperability, scalability, and product security requirements, as well as operation and maintenance, including software updates, and training, documentation, and technical support.
- *Performance*: Maximum capacity and performance features of intrusion detection and prevention.
- *Security Capabilities*: Information gathering, logging, detection, and prevention of intrusions.

6.4.2 Intrusion Prevention

Intrusion prevention technologies are differentiated from intrusion detection technologies by the characteristic that intrusion prevention system (IPS) technologies respond to a detected threat by attempting to prevent it from succeeding. Several response techniques are used for intrusion prevention, which can be divided into the following groups (Scarfone and Mell 2007):

- *IPS Stops Intrusion Attack Itself*: Examples of how this could be done are as follows:
 - Block access to target or possibly other likely targets from offending user account, IP address, or other intrusion attacker attribute.
 - Block all access to targeted system, service, application, or other resource.
 - Terminate network connection or user session that is being used for intrusion attack.
- *IPS Changes Security Environment*: IPS could change configuration of other security controls to disrupt an intrusion attack. Common examples are:
 - Cause patches to be applied to a host if IPS detects that the system has vulnerabilities.

- Reconfigure a network device, e.g., firewall, router, switch, to block access by the intrusion attacker or to the target, and alter a system-based firewall on a target to block incoming attacks.
- *IPS Changes Intrusion Attack's Content:* Some IPS technologies can remove or replace malicious portions of an intrusion attack to make it benign.
 - A simple example is an IPS that removes an infected file attachment from an e-mail and then permits the cleaned email to reach its recipient.
 - A more complex example is an IPS that acts as a proxy and normalizes incoming requests, which means that the proxy repackages the payloads of the requests, discarding header information. This might cause certain intrusion attacks to be discarded as part of the normalization process.

With regard to potential vehicle cyber criminal intrusion attacks, WLAN technology is the most important technology for use with intrusion prevention systems. Most WLANs use the IEEE 802.11 family of WLAN standards. IEEE 802.11 WLANs have two fundamental architectural components, see Fig. 6.27:

- An ACCESS POINT that logically connects STATIONs with a distribution system, which is typically a system's wired infrastructure.
- A STATION, which is a wireless endpoint device.

Some WLANs also use wireless switches which are devices that act as intermediaries between ACCESS POINTS and the distributed systems. The purpose of a switch is to assist in managing the WLAN infrastructure. In WLANs without wireless switches, the ACCESS POINTs connect directly to the distributed systems. The IEEE 802.11 standard also defines the following two WLAN architectures:

- *Ad Hoc Mode:* Peer-to-peer mode that does not use ACCESS POINTs, involving two or more STATIONs communicating directly with one another.
- *Infrastructure Mode:* ACCESS POINTs connect wireless STATIONs to a distributed system, typically a wired network.

Each ACCESS POINT and STATION on a WLAN can be identified by its Media Access Control (MAC) address a unique 48-bit value that is assigned to a wireless network interface card.

Some of the wireless intrusion detection and prevention techniques terminate connections between ill-conditioned or misconfigured STATIONs and an authorized ACCESS POINTs or between an authorized STATION and an ill-conditioned or misconfigured ACCESS POINT. This is typically done by sending messages to the endpoints, telling them to disassociate the current session. The IPS then refuses to permit a new connection to be established. Most IPSs are able to specify the prevention capability configuration for each type of alert. This usually includes enabling or disabling prevention, as well as specifying which type of prevention

capability should be used. Others have a learning or simulation mode that suppresses all prevention actions and instead indicates when a prevention action would have been performed. This allows monitoring and fine-tuning of the configuration of the prevention capabilities before enabling prevention, which reduces the risk of performing prevention actions on benign activity.

Thus, the main task of intrusion prevention is to defend a CPS by detecting an attack and possibly repelling it. Detecting hostile attacks depends on the number and type of appropriate actions, which can be obtained from publicly available data, found in the National Vulnerability Database (NVD), the US government repository of standards based vulnerability management data, or the CVE database, a dictionary of publicly known information security vulnerabilities and exposures. Both of these databases are sponsored by the US Department of Homeland Security/US Office of Cybersecurity and Communications/Computer Emergency Readiness Team and help in understanding the severity of the current security threat landscape (see Sect. 6.1.1). Therefore, intrusion prevention requires well-selected investigations of threats because adversaries are seeking out and exploiting network, device, and application vulnerabilities to attack, causing serious problems for the vehicle attacked. Thus, intrusion detection and prevention strategies are becoming a critical issue for automakers, OEMs, and suppliers.

The main activities of an IDPS are summarized in Fig. 6.24. If a cyberattack is suspected, an alarm list of possible attacks is created, and the component or subsystem the intruder is attempting to attack is locked (Landrum et al. 2014). As can be seen in Fig. 6.24, preprocessing describes processing performed on raw data, transforming this data into a format that is more easily and effectively processed for the purpose of intrusion detection. There are a number of different tools and methods used for preprocessing. One is feature extraction, which pulls out specified data that is significant in some particular context, such as intrusion. The ruleset shown in Fig. 6.24 contains three components:

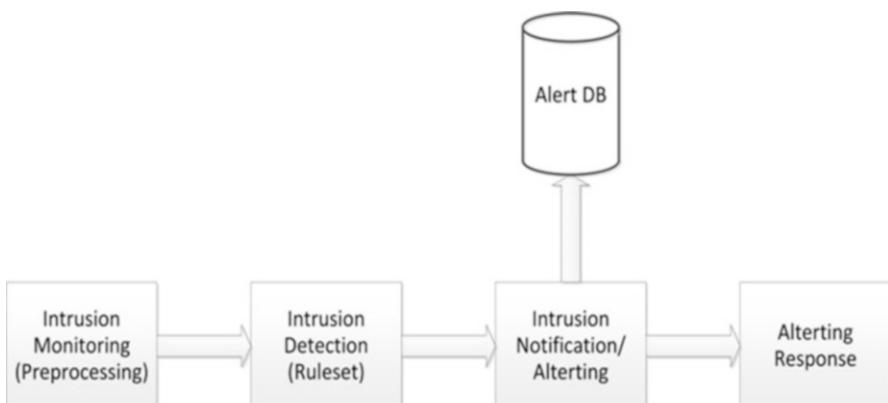


Fig. 6.24 Intrusion detection and prevention systems tasks

- Set of rules
- Database
- Interpreter for the rule

A rule can be defined as an ordered pair of symbol strings. The ruleset has a predetermined, total ordering; and the database is a collection of intrusion-related patterns. The interpreter operates by scanning the ordered pair of pattern strings of each rule until one is found that can be successfully matched against the intrusion-related pattern of the database.

If an intrusion is identified, the notification feature of the intrusion prevention system, shown in Fig. 6.24, starts an alert response as an operational routine encapsulating the identified intrusion scheme. Hence, the intrusion prevention architecture, shown in Fig. 6.24, is a key element in controlling the information flow between attack surfaces and mission-critical systems, as shown in Fig. 6.25.

In addition to the foregoing, the IDPSA architecture scheme, shown in Fig. 6.26, illustrating detecting and preventing unknown vulnerabilities is a task which expands the ruleset-based approach in Fig. 6.24 through an artificial neural network. Executing this approach require again data gathering and pre-processing which means that all incoming data is collected, transformed and normalized to standard entities. Thereafter, feature extraction from this data is required in which feature entities are objects of information that could be used like performance evaluation for number of packets transferred between vehicles, delay in transfer of packets, number of dropped packets and more. Other basic features could be the information in the header of the packets transferred which could include, for example:

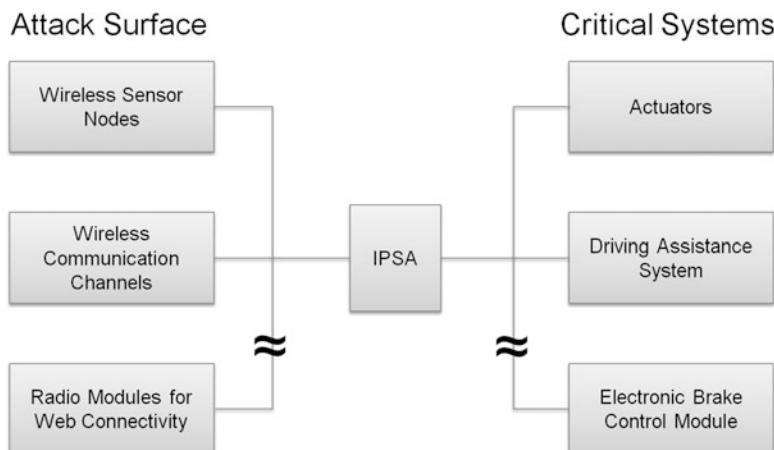


Fig. 6.25 Intrusion prevention system architecture (IPSA) lies in between attack surfaces and mission-critical systems of vehicle cyber-physical systems

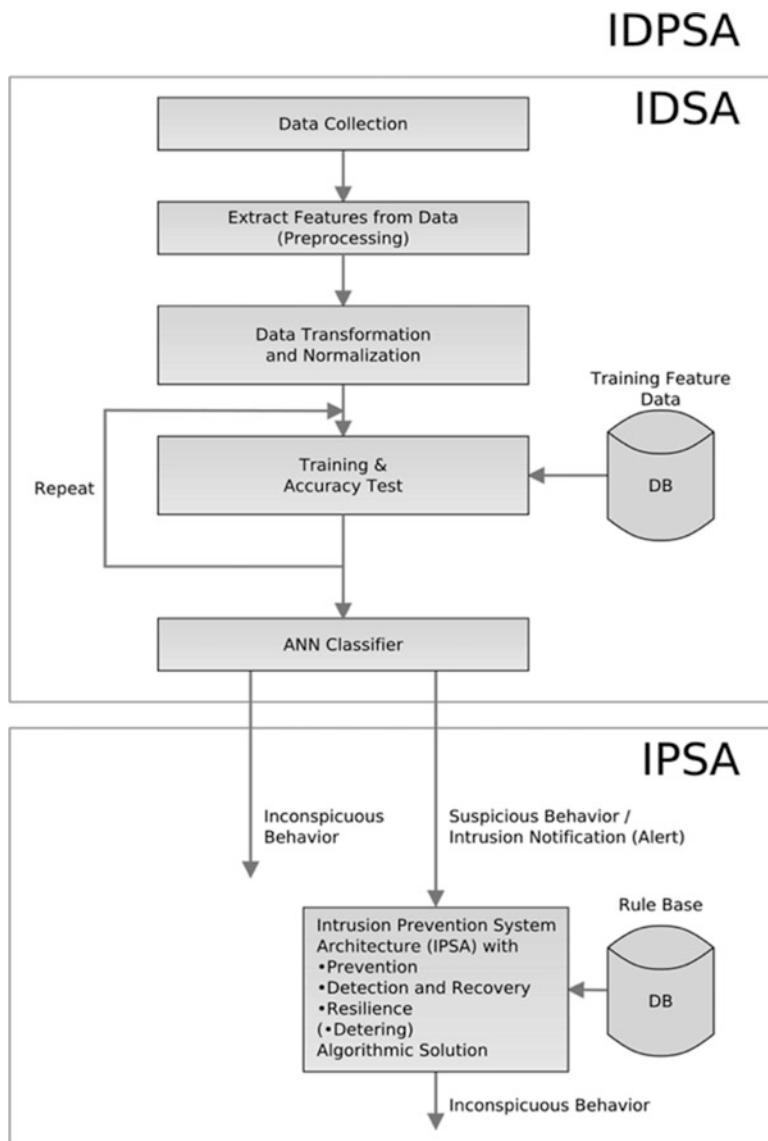


Fig. 6.26 Intrusion detection and prevention system architecture (IDPSA)

- IP address
- Payload size and type
- Port

- Source and destination MAC
- Time to live

The type of artificial neural net is the next important step. In IDSA a feed forward neural network (FFNN) type is used, consisting of an input layer with as many neurons as number of features used for classification, two hidden layer with, for example, less number of neurons and a final output layer. The FFNN requires training based on specified features. The step after training the FFNN is to test it in place with the features assigned to normal and abnormal behavior based on a performance metrics which describe the accuracy of the detection rate and false alarm rate of the IDSA. Accuracy is calculated by ratio of correct classification to the total test data set. Detection rate is the ratio of the number of correct detection to the total number of attacks. In this context an abnormal or anomaly behavior can be received using a statistical based threshold approach.

As reported in (Karim and Proha 2014), numerous static, dynamic, and hybrid solutions are available for analyzing patterns and signatures in program codes and the behavior of program executions in order to identify the presence of malicious agents in the system under test, thereby helping to disable them. In real-time CPSs, which are used for mission-critical tasks, intrusion can be detected through static timing analysis.

In Zimmer et al. (2010), three mechanisms for time-based intrusion detection are described that detect the execution of unauthorized instructions in real-time CPS environments. Such intrusion detection utilizes information obtained by static timing analysis. For real-time CPSs, timing bounds on code sections are available as they are already determined prior to the schedulability analysis. The Zimmer et al. (2010) paper demonstrates how to provide microtimings for multiple granularity levels of application code. Through bound checking of these microtimings, techniques have been developed to detect intrusions (i) in a self-checking manner by the application and (ii) through the operating system scheduler (OSS), which are novel contributions in the real-time CPSs domain.

Another option is testing for stability and resiliency because the complex software systems found in today's vehicles are prone to attacks. Automakers and their OEMs need to fully assess vehicle security to ensure a stable and resilient system. To test for stability and resiliency, several methodologies are used:

- *Functional and Performance Test*: Validates security components under valid traffic and cyberattack conditions.
- *Impairment Test*: Validates performance when communication is impaired; typically used with delayed, dropped, or erroneous packets.
- *Resiliency Test*: Validates operation under degraded or failure conditions, such as sensor failure, actuator failure, etc.
- *Stress Test*: Validates system or components beyond normal operational capacity to observe how the system or components operate.

With regard to tests, another important strategy is the security penetration test (SPT). This test aims to identify weaknesses in IT systems of a defined target environment on the basis of a systematic methodology. When implementing SPTs, the same techniques, tools, and expert knowledge are used, which are also used by real attackers. Hence, experienced penetration testers are required which use automated and manual test procedures to present realistic attack scenarios. In addition to technical analyzes, social-level attacks can also be part of a SPT to test the security awareness of employees of a company with regard to the dissemination of information and the conscious or unconscious use of unauthorized applications. Depending on the targeted object, the vehicle, the following strategies for SPTs can be distinguished as described by TechTarget networking.de:

- *External Penetration Strategy*: External tests deal with attacks on the network. The methods used are carried out from outside the vehicle to be attacked, i.e., through the Internet. This test can be carried out with no or complete knowledge of the vulnerable technical environment. Typically, this penetration test begins with public available information about the vehicle, subsequent network spanning, and others.
- *Internal Penetration Test Strategy*: Internal tests are carried out within the vulnerable technical environment. The penetration test simulates an attack on the internal network. The focus here is to understand what might happen if the network was successfully penetrated or what an authorized user could do to capture specific information resources of the compromised network. One important attack is sniffing which is used to a considerable extent with internal penetrations tests. The sniffer or the computer is directly connected to the network in promiscuous mode, which allows a considerable amount of information to be collected. For sniffing, a variety of free and commercial tools are available, such as Wireshark (the former Ethereal), the Microsoft Message Analyzer (the successor to Netmon), or the Viavi Observer Analyzer.
- *Blind Test Strategy*: In blind tests, one tries to simulate the actions and procedures of a real hacker. As with a real hacker attack, the test team has only limited or no information about the vehicle before performing the penetration test. The penetration test team uses public available data to collect information about the targeted object and perform the penetration tests. These blind tests can provide a lot of information about the targeted object that would otherwise remain unknown – for example, this type of penetration tests can raise problems such as additional Internet access points, directly connected networks, and public available confidential/protected information. However, blind tests are more time-consuming and expensive because the necessary effort of the test team for the target search is higher.
- *Double-Blind Test Strategy*: Double-blind tests are an important test component, since it is possible to check the security monitoring and identification of security incidents as well as the escalation and reaction procedures of the targeted object.
- *Targeted Testing Strategy*: In the case of targeted or systematic tests, sometimes referred to as a lights-turned-on approach, the penetration test team are involved

in the test. The test activities and the information regarding the target and network design are generally known. Targeted penetration testing can be more efficient and cost-effective if the goal of the test is more focused on the technical side or design of the network, rather than on incident response and other workflows of the targeted object. In contrast to blind tests, a systematic test can be carried out in less time and with less effort. The only difference is that this may not provide a complete picture of the targeted object's vulnerabilities and reactivity.

In addition to the aforementioned methods, a large number of distributed computing resources connected by a network representing a so-called “cloud” can be used to deliver essential vehicle applications with regard to connected vehicle needs. Thus, in a connected vehicle, the cloud allows challenges in the vehicle ecosystem to be met, which will increase the value of current business and induce new third parties to take part in the cloud (see Sect. 6.5.4).

Furthermore, vehicle owners will also be able to connect to the vehicle remotely from other devices which, unfortunately, will open the door to new intrusion points for cyberattacks.

6.5 Functional Safety and Security

The growing complexity and networking of today’s automotive systems increases the importance of functional safety and security. Safety and security issues have been treated separately for the most part.

Safety systems are set up and operated totally disjoined from other systems, having their own physically separate system and gateways when connecting with others. For functional safety, the absence of reaction is required and has to be proven, usually resulting in a limited read-only access to the safety system.

Trends such as remote access via the Internet require rethinking this separation and setting up concepts for systems that allow common usage safely and securely. This can be achieved by embedding security measures to guarantee the correct execution of functional-safety-relevant operations. This requires that communication systems offer flexible frameworks that on the one hand run the correct utilization of resources needed for safety and, on the other hand, offer respective services, such as access rights or authentication, to other applications.

Using redundancies by integrating safety-critical, security-relevant, and standard operations within a single communication network also allows for cost-efficient solutions. Hence, these trends break up the isolated structure of networking and, therefore, enable new risks and threats concerning safety and security, and set new challenges for the safety and security measures in automotive systems.

6.5.1 Security for Wireless Mobile Networks

As previously mentioned, wireless technologies are bringing significant changes to communication networking and services. Due to their unique features, such as a

shared medium, limited resources, and dynamic topology, wireless ad hoc networks are vulnerable to a variety of potential attacks. However, common security measures employed for wired networks are not enough to protect the nodes of the networks against complex attacks. Therefore, a new line of defense, the intrusion detection approach, has been added. In this section, the wireless mobile networks, along with their security issues, are introduced. The most obvious characteristic of wireless networks is that communication takes place over a wireless channel, usually a radio channel. Such a channel suffers from a number of vulnerabilities:

- *Address Spoofing*: Scenario in which a network node uses the address of another node to exploit privileges granted to the legitimate authorized user of the identity. In WLANs, this can be done by changing the media access code MAC address of a network interface.
- *Eavesdropping*: Placing an antenna at an appropriate location, a cyberattacker can overhear information that the authorized user transmits or receives. Eavesdropping is often used to carry out attacks, notably passive attacks.
- *Location Tracking*: Tracing calls made by a cellular network or using network sensors.
- *Medium Access Control*: Following the rules of a MAC protocol in an attempt to obtain more than a fair share of a WLAN bandwidth.
- *Unauthorized Transmission*: Injecting forged or replayed frames. Attack goal can be to illegitimately join the WLAN.

Passive attacks consist of listening to the communication network and analyzing the captured data without interacting with the network. Such cyber physical attacks can be illustrated by the weakness of wired equivalent privacy WEP, (see Sect. 6.2) a security protocol, specified in the IEEE Wi-Fi standard 802.11b, that is designed to provide a WLAN with a level of security and privacy comparable to what is usually expected of a LAN. WEP seeks to establish protection similar to a wired network's physical security measures by encrypting data transmitted over the WLAN to protect against misdeeds. In case of an unprotected WLAN, the cyberattacker does not need to have physical access to any device to connect to the network. Hence, WEP is intended to transform this simple access into a difficult one by increasing the level of difficulty of attacking WLANs which comes from:

- *Broadcasting Nature of Radio Communications*, because eavesdropping on wireless transmissions is simple. This can be prevented by encrypted messages. There are two main families of encryption techniques: stream ciphers and block ciphers.
- *Connecting to the WLAN*, which does not require physical access to the network access point. Thus any device can try to illegitimately use the services provided by the WLAN, which can be prevented by authentication of the mobile STAs before allowing their connection to the WLAN.

Authentication of an STA is based on a simple challenge-response protocol. Once authenticated, the STA communicates with the access point by means of encrypted

messages. The key used for encryption is the same as the one used for authentication. The encryption algorithm specified by WEP is based on the four-line stream cipher Rivest Cipher 4 (RC4). Stream ciphers produce a long pseudorandom byte sequence out of a short secret seed value. This pseudorandom sequence is fused with the clear text message using the *XOR* operation to generate the encrypted message. WEP works in the same way. The sender of a message M initializes the RC4 algorithm with the secret key and connects the pseudorandom sequence K generated by RC4 logically through the *XOR* operation with M . The receiver of the encrypted message $M \oplus K$ uses the same secret key to initialize the RC4 algorithm that produces the same pseudorandom sequence K whereby K is connected through *XOR* operation to the encrypted message to obtain the message:

$$(M \oplus K) \oplus K = M.$$

As mentioned in Butayán and Hubaux (2007), this description is not precise enough. There is more to be taken into account than what WEP does when encrypting messages. It can be seen that if encryption is appropriate, then every message would be encrypted with the same pseudorandom sequence K .

Let's assume that a cyberattacker is eavesdropping on two encrypted messages, $M_1 \oplus K$ and $M_2 \oplus K$. With regard to the *XOR* operation of these two messages, we receive

$$(M_1 \oplus K) \oplus (M_2 \oplus K) = M_1 \oplus M_2$$

which is equivalent to one message being encrypted with the other, but clear messages are far from being pseudorandom sequences. Thus, $M_1 \oplus M_2$ is a weak encryption; and the cyberattacker is likely to be able to break it using the statistical properties of the clear messages.

To address this problem, WEP appends an initialization vector (IV) to the secret key before initializing the RC4 algorithm, where the IV changes for every message, as described in Butayán and Hubaux (2007). This ensures that the RC4 algorithm produces a different pseudorandom sequence for every message. The receiver should also know that the IV will be able to decrypt the messages received. For this reason, the IV is sent in a clear message together with the encrypted message. Figure 6.27 illustrates the WEP encryption and decryption procedure after Butayán and Hubaux (2007).

From Fig. 6.27, it can also be seen that before encryption, the sender attaches an integrity check value (ICV) to the clear message. The purpose of this value is to enable the receiver to detect any malicious modifications of the message by a cyberattacker. In case of WEP, ICV is a CRC value computed for the clear message. As a CRC value alone cannot enable the detection of malicious modifications, because the attacker can compute the new CRC value for the modified message, the CRC value is also encrypted in WEP. The rationale is that in order to modify the message in an unnoticeable way, the cyberattacker must now encrypt the new CRC

Fig. 6.27 Encryption and decryption in WEP with SK as the security key, modified after (Butayán and Hubaux 2007)

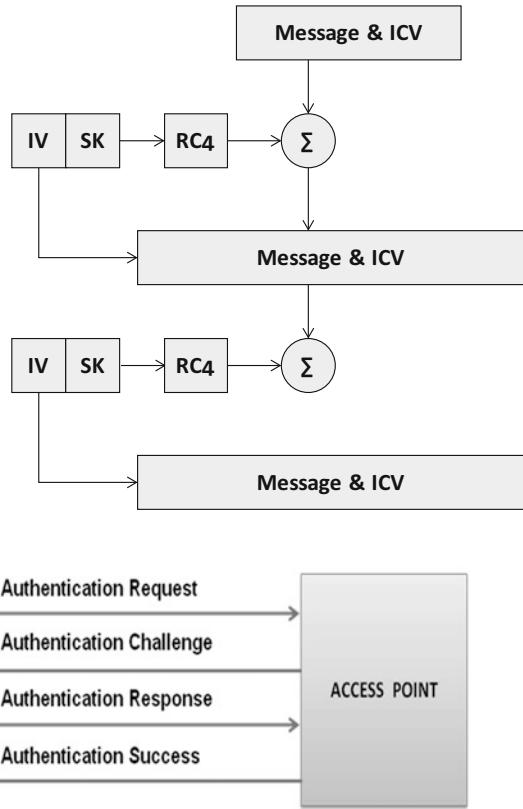


Fig. 6.28 WEP authentication

value but cannot do this without the knowledge of the secret key (Butayán and Hubaux 2007).

WEP also includes a device-level authentication mechanism through which STATION must provide, to the ACCESS POINT, a proof of ownership of the key they share for which four messages are exchanged, as shown in Fig. 6.28.

STATION makes a request. ACCESS POINT shown in Fig. 6.28 sends a challenge, such as a 128-bit random value. STATION sends a response, e.g., a 128-bit random value encrypted with the WEP stream cipher. ACCESS POINT decrypts the response. If the decrypted response matches the original challenge value, then a positive authenticate response is returned to STATION. WEP authentication is one way, i.e., the ACCESS POINT is not authenticated by STATION (Das et al. 2012). After completion of the authentication phase, subsequent traffic is not authenticated. Therefore, the protocol is vulnerable to the authentication spoofing attack. A cyberattacker may obtain the key by using XOR operation for the intercepted challenge value and its response. The key stream may be used by the cyberattacker to create proper responses to new challenges (Housley and Arbaugh 2003).

6.5.2 Security for Sensor Networks

Recent technological advances have made it possible to deploy wireless sensor networks consisting of a large number of functional sensor nodes that communicate over short distances through wireless links (Akyildiz et al. 2002). The desirable features of sensor networks have motivated many researchers to develop protocols and algorithms to support the various applications of sensor networks. A common use of sensor networks in the automotive domain is to sense and monitor cyber-physical systems and/or components. Two access control approaches are in use for wireless sensor networks:

- *Uni-Access Scheme*: Mainly used to access one sensor node at a time. The user can directly access the data on any sensor node in the network without going through the base station, and a sensor node can protect its data so that only authorized users can access it.
- *Multi-Access Scheme*: Applies public key cryptography to achieve an additional feature, which allows a user to access data on many sensor nodes via a single query.

In sensor networks, one can differentiate between two attack forms in (Das et al. 2012):

- *Attacks on Communication*: A cyberattacker can easily perform a denial-of-service (DoS) attack by jamming the wireless channel and disabling the network operation. This attack is easy to intrude and common to the protocols in every sensor network.
- *Attacks on Sensor Nodes and Users*: Once a sensor node is compromised, the cyberattacker has full control of it. The attacker can learn keys and all sensed data stored on the compromised sensor node.

Typical security problems in sensor nodes include:

- *False Node*: An intruder may insert a node into the sensor network that feeds false data or prevents the passage of true data. Such problems are known to occur in distributed network systems as well as ad hoc networks.
- *Legitimate Addition of a Node to an Existing Sensor Network*: If a sensor node needs to be replaced or another sensor node needs to be added to an existing sensor network, securely integrating the new sensor node into the existing sensor network is an issue.
- *Passive Information Gathering*: If communication between sensors or between sensors and base stations is in the clear, then an intruder with an appropriately powerful receiver and antenna can easily pick up the data stream. If the thumbed information is encrypted, then it is important to know which cryptographic approach has been used by the compromised sensor node.

- *Subversion of a Node:* A particular sensor might have captured information stored on it, such as the key, which might be obtained by the intruder. If a sensor node has been compromised, then the issue is how to exclude that sensor node, and that sensor node only, from the sensor network.

Furthermore, sensor network security has some unique features that do not exist in other networks. For example, a sensor node has limited memory space so that the number of keys that can be stored in its memory, as well as the variables for asymmetric cryptographic algorithms, is limited. Moreover, any security solution with a static configuration may not be suitable for ad hoc sensor networks because sensor nodes have mobility, and the sensor network topology may change frequently. Sensor nodes have to continuously detect possible intrusions because their neighbor nodes are not fixed. Similarly, a malicious node with mobility can roam in a sensor network and attack different parts of the network (Xiao 2006).

However, the encryption-decryption techniques devised for traditional wired networks cannot feasibly be applied directly to wireless networks; wireless sensor networks (WSNs) in particular. Utilizing any encryption scheme requires transmission of extra bits, hence extra processing, memory, and power which are important resources for the sensor nodes. Applying security mechanisms, such as encryption, could also increase delay, jitter, and packet loss in WSNs (Saleh and Khatib 2005). Moreover, some critical issues arise when applying encryption schemes to WSNs, such as how the keys are:

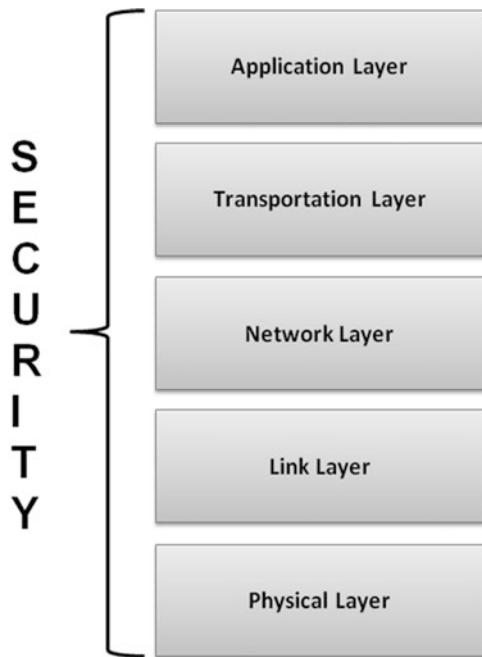
- Assigned to a new sensor added to the network
- Generated or disseminated
- Managed
- Renewed, to ensure robust security for the network
- Revoked

As minimal human or no human interaction with the sensor nodes is a fundamental feature of WSNs, how the keys can be modified from time to time is an important issue for encryption because adoption of preloaded keys or embedded keys would not be an efficient solution (Pathan et al. 2006).

A holistic approach reported in Avancha (2005) aims to improve the performance of WSNs with regard to security, longevity, and connectivity under changing environmental conditions. The holistic approach to security is concerned with involving all layers of WSNs to ensure the overall security of a network, as shown in Fig. 6.29.

For such a network, a single security solution for a single layer might not be an efficient solution, where employing a holistic approach could be the best option. The holistic approach has some basic principles, such as security has to be ensured for all layers of the protocol stack. If no physical security for the sensors is ensured, the security measures must be able to exhibit a graceful degradation if some of the sensors in the network are compromised, out of order, or captured by an adversary. Security measures should be developed to work in a decentralized fashion. If security is not considered for all of the security layers, e.g., if a sensor is somehow

Fig. 6.29 Holistic view of security in wireless sensor networks



captured or jammed in the physical layer, the security for the overall network breaks despite the fact that there are some efficient security mechanisms working in other layers. By building security layers using a holistic approach, protection is established for the overall network (Pathan et al. 2006).

6.5.3 Platform Security

Platform security refers to the security architecture, tools, and processes that ensure the security of an entire computing platform (hardware, software, network, storage, and other components) by using a centralized security architecture or system. Platform security secures all components and layers within a platform. This allows for the elimination of individual security measures and the use of multiple applications/services to secure different layers of an ICT environment. Security at the platform level simplifies the security process for information technology and developers. However, once the security is cracked, the entire platform is vulnerable. Thus, a trusted platform module (TPM) is required, which is a hardware device that is basically a secure cyber-physical controller with added cryptographic functionality. It works with supporting software and firmware to prevent unauthorized access to a platform. The TPM contains a hardware engine capable of performing up to 2048-bit Rivest-Shamir-Adleman (RSA) encryption/decryption. The TPM uses its built-in RSA engine during digital signing and key wrapping operations.

6.5.4 Cloud Computing and Data Security

Cloud computing is a new information technology infrastructure in which both, the application, delivered as a service to users at anytime, anywhere whenever the Internet is available, and the computing resources, hardware and systems software in data centers, may be provided. Services provided by the cloud can be at different levels, described by the X-as-a-Service (XaaS) model, whereby X could be:

- Hardware
- Infrastructure
- Platform
- Software

Cloud computing is a type of Internet-based computing that provides shared computer processing resources and data to computers and other devices on demand. When deciding to use cloud computing, users need to be aware that in addition to the services provided, i.e., ability and system performance, security should be of particular concern. Therefore, cloud-based security services are an important issue when migrating from dedicated hardware solutions to cloud-based security services using the XaaS model. At its core, cloud computing is used to describe data acquisition or distribution through the Internet and wireless networks. In the XaaS model, the application data is generally hosted in the cloud and made available to users via an Internet interface. XaaS users often download a thin client, which gives them access to the application via a web browser. The increase in virtual and cloud networks is boosting demand for cloud-based security because data and applications are now portable and distributed across a wide variety of networks. This means that security applications need to live as software in the cloud, rather than on dedicated hardware devices, thus protecting specific potential intrusion points of the vehicle network.

Today's vehicle users will be able to access applications from a screen in the vehicle, thereby enjoying the same level of digital services that they have in their homes, at work, or on the go via smart devices.

Moreover, cloud computing could also bring additional benefits to the average vehicle in many ways. One of these is actually related to drive dynamics. New vehicles often have electronically adjustable suspensions; with cloud computing, they could be more automated, providing a much better customized drive. The same is possible for electronically disconnecting sway bars and other off-roading features on some Jeeps and multipurpose sport utility vehicles SUVs. Another area is bringing personalized data into the cabin of a vehicle. For drivers, this means that their data from online calendars and contacts, a personal music library, and other data will travel with them and be available right at their fingertips. Thus, data protection in cloud computing is a crucial security issue. Hence, before moving into the cloud, users should clearly identify the data to be protected and classify data based on its security implications. Therefore, the security classification must be specified because different types of data may have different value and hence different security implications for confidentiality, integrity, and availability (CIA).

In cloud computing, data security has become more complicated because users may be confronted with all kinds of cyberattacks with regard to the intrinsic cloud characteristics. This requires an understanding of the potential security threats to identify where the cyberattackers may come from and what kind of cyberattacks they may launch. As reported in (Das et al. 2012), there are two types of cyberattackers:

- *Insiders*: Are users with authorized access privileges inside the cloud organization or at the cloud service provider's site and possibly the cloud service provider itself. They can launch serious attacks by:
 - Obtaining control of the virtual machines
 - Gaining access to sensitive information by logging all communication information of other cloud users, thereby abusing their privileges

Therefore, cloud users should establish a trusted relationship with cloud service providers. The occasional misbehavior of a cloud service provider may be any, or a combination, of the following:

1. Colluding with a small number of malicious users for the purpose of harvesting data files and their contents.
2. Deciding to hide data corruption caused by server hacks or Byzantine failures, thereby maintaining reputations. The Byzantine model (Dolev 1982) assumes a system with n components and an adversary that may compromise up to $k < n$ components. Therefore, the identified vulnerabilities V_j are

$$V_j = f(t_i, q_j)$$

where

$$V_j \subseteq V$$

is a set of

$$j \in [0 : k]$$

faulty components of q_j . The threat transition function $D_f(t_j)$ then ^is

$$D_f(t_i) : V_j \xrightarrow{t_j, a} V_s$$

where

$$V_j \subseteq V_s$$

which is an adversary that has compromised the components of V_j and was restricted to attacking those states with

$$V_s \supseteq V_j$$

This defines the allowable system transitions that the adversary can exploit. But faulty components cannot be recovered with this model.

- Gaining data information by eavesdropping and monitoring network traffic (Kao and Marculescu 2006).
- Neglecting, keeping, or deliberately deleting rarely accessed data files, thereby saving resources.

For valuable and/or sensitive data or services, cloud users should implement their own security protection mechanisms, such as cryptographic protection.

Outsiders: Cloud computing could be vulnerable to malicious attacks from the Internet. Outsider attackers can launch passive attacks, such as eavesdropping on the network traffic, and active attacks, such as phishing legitimate users credentials, manipulating network traffic, and probing the cloud structure.

In cloud computing, sensitive data pooled in the cloud demands that the cloud data storage and sharing service be responsible for secure, efficient, and reliable distribution of data to a potentially large number of authorized users (Das et al. 2012). One way of providing a secure data access service is through cryptographic methods. The data owner and data user encrypt data before storing it in the cloud, retaining the secret key.

In the literature, related mechanisms can be found in the areas of shared cryptographic file systems and access control of outsourced data (Capitani di Vimercati 2007; Kallahalla et al. 2003; Goh et al. 2003).

Since diverse mobile technologies are available, mobile cloud computing supports and adapts itself to multiple mobile platforms and devices. Thus, mobile-device-centric cloud computing consists of an infrastructure formed by the mobile devices themselves. In this context, security concerns may depend on how the infrastructure is organized to deliver mobile cloud security. A cloud support service specific to mobile devices has been investigated through a dedicated infrastructure and a related model (Satyanarayanan et al. 2009).

Cloud providers, for example, Amazon, Azure, and Google, manage the security and availability of their cloud infrastructure like any other larger enterprise. They monitor and investigate security incidents or events. Cloud service providers (CSP) must therefore distinguish between legitimate penetration tests (see Sect. 6.4.2) by customers and real attacks. If customer tests trigger the wrong countermeasures, connections can be routed into a DDoS black hole or intrusion prevention systems can be activated. This not only costs the CSP time and valuable resources, because of the shared infrastructure of cloud systems, but can also have a negative impact on other customers.

Before testing, it is important to know the limits of cloud penetration testing. This means, for example, that one is aware of its responsibility. This changes depending on what type of system is checked because IaaS, PaaS, or SaaS each have different requirements. An IaaS environment, for example, allows a much more aggressive approach than SaaS, which is mainly due to the fact that IaaS often has numerous users (vehicles) working and/or connected on the system and that a failure would have a massive impact on them which is not the case with SaaS. A concentration test can take a system completely offline. This is not a problem if the OEM or Tier 1 supplier company owns the server completely, but a huge problem when other users are taken offline.

6.5.5 Functional Safety

Functional safety is part of the overall safety of a vehicle system, or a component of it, that depends on the cyber-physical system or its components for operating correctly in response to its inputs, including safe management of likely operator errors, hardware failures, and environmental changes. Functional safety is intrinsically end-to-end in scope, which means that it has to treat the function of a system or subsystem or component as part of the function of the whole system. This means that while functional safety standards focus on electrical, electronic, and programmable systems (E/E/PS), the end-to-end scope, in practice, of functional safety methods has to extend to the non-E/E/PS parts of the system that the E/E/PS actuates, controls, or monitors (URL11 2016).

Functional safety is achieved when every specified safety function is carried out and the level of performance required of each safety function is met. This is normally achieved by a process that includes the following steps as a minimum (URL11 2016):

- *Identify the Required Safety Functions:* This means hazards and safety functions have to be known or identified.
- *Assess the Risk Reduction Required by the Safety Function:* This involves a safety integrity level (SIL), performance level (PL), or other quantification assessment. An SIL applies to an end-to-end safety function of the safety-related system, not just to a component or part of the system.
 - *Automotive Safety Integrity Level (ASIL)* is a risk classification scheme defined by ISO 26262, *Functional Safety for Road Vehicles Standard* which is an adaptation of the SIL used in IEC 61508 for the automotive industry. This classification helps define the safety requirements necessary to be in line with the ISO 26262 standard. ASIL is established by performing a risk analysis of a potential hazard by looking at the severity, exposure, and controllability of the vehicle operating scenario. The safety goal for that hazard in turn carries the ASIL requirements. There are four ASILs identified by the standard: ASIL A is comparable to SIL-1, ASIL B/C is comparable to SIL-2, and ASIL D is comparable with SIL-3. For SIL-4, no comparison exists with ASIL. ASIL D dictates the highest integrity requirements for the product and ASIL A the

lowest. However, ISO 26262 does neither provide normative nor informative mapping of ASIL to SIL. ASIL is a qualitative measurement of risk, while SIL is quantitatively defined as the probability or frequency of dangerous failures, depending on the type of safety function. Thus, in IEC 61508, higher-risk applications require greater robustness to dangerous failures. Hazards that are identified as quality management (QM) do not dictate any ASIL safety requirements (URL12 2016).

- *Ensure Safety Function Performs to the Design Intent:* This includes under conditions of incorrect operator input and failure modes. The design and life cycle are managed by qualified and competent engineers carrying out processes to a recognized functional safety standard. In Europe, that standard is IEC EN 61508 or one of the industry-specific standards derived from IEC EN 61508 or some other standard, such as ISO 13849.
- *Verify the System Meets the Assigned SIL (ASIL, PL, or agPL):* This can be done by determining mean time between failures (MTBF) and the safe failure fraction (SFF), along with appropriate tests. SFF is the probability of the system failing in a safe state. The critical or dangerous state is identified from a failure mode and effects analysis (FMEA) or failure mode effects and critical analysis (FMECA) of the system under test.
 - *MTBF:* Predicted elapsed time between inherent failures of a system during operation which can be calculated as the arithmetic mean time between failures of a system using the following equation:

$$\text{MTBF} = \frac{\sum(\text{start of downtime} - \text{start of uptime})}{\text{number of failures}}$$

- *SFF:* Takes into account any inherent tendency to fail toward a safe state. SFF is the sum of the rate of safe failures plus the rate of detected dangerous failures divided by the sum of the rate of safe failures plus the rate of detected and undetected dangerous failures. It is important to realize that the only types of failures to be considered are those which could have some effect on the safety function. SFF can be calculated using the following equation:

$$\text{SFF} = \frac{(\sum \lambda_S + \sum \lambda_{DD})}{(\sum \lambda_S + \sum \lambda_D)}$$

where

λ_S : Rate of safe failure

$(\sum \lambda_S + \sum \lambda_D)$: Overall failure rate

λ_{DD} : Rate of detected dangerous failure

λ_D : Rate of dangerous failure

- *FMEA:* The first step of a system reliability study involves reviewing as many components, assemblies, and subsystems as possible to identify failure modes and their causes and effects. For each component, the failure modes and their

resulting effects on the rest of the system are recorded in a specific FMEA worksheet. FMEA can be a qualitative analysis but may be put on a quantitative basis when mathematical failure rate models are combined with a statistical failure mode ratio database (URL13 2016).

- *FMECA*: An extended FMEA indicates that a criticality analysis is performed, too.
- *Conduct Functional Safety Audits*: Examine and assess the evidence that the appropriate safety life cycle management techniques were applied consistently and thoroughly in the relevant life cycle stages.

Neither safety nor functional safety can be determined without considering the vehicle cyber-physical system as a whole and the environment with which it interacts. Functional safety is inherently end-to-end in scope.

6.6 Car Hacking Examples

Today's vehicles can be understood as a complex network of ICT systems. As vehicles become increasingly computerized, their attack surfaces also grow and increase. Worldwide security research demonstrates a huge number of vulnerabilities in vehicle electronic systems, showing that automakers have not placed enough emphasis on developing secure vehicular ECUs and communication systems. ECUs receive inputs from sensors and make adjustments to a series of actuators controlling, e.g., the operation of the engine's physical components. This allows for ignition timing and the fuel/air mixture to be dynamically adjusted in real time, which can save fuel and optimize performance. Prior to the use of ECUs for engine management, these functions were controlled mechanically (Eyal 2007).

To understand the magnitude of the security problems facing today's vehicles, it is first necessary to address the interconnectivity of today's vehicle components which are managed by a vehicle's onboard computer systems. Once hackers have access to personal and other information from vehicle systems, they are able to find myriad new ways to use it. For example, GPS information could be used to track a driver's habits and schedule.

Vehicle hacking is the manipulation of the code in a vehicle's ECU to exploit a vulnerability and gain control of other ECU units in the vehicle. An excellent timeline of recent vehicle hacks is given in (Currie 2015), which we have the author's permission to use.

When the CAN system bus was developed in the mid-1980s, its designers certainly did not envision that the bus would one day be targeted by attackers seeking to take over or otherwise manipulate the function of an automobile (see Sect. 6.4.2). As recently as 10 years ago, hacking vehicles received less media attention and was not a worry to most vehicle users. With regard to the last decade, and particularly the last several years, vehicle hacking has become a real concern. In a 2015 study by Kelley Blue Book®, for which members of the vehicle-buying public were polled, it was found that 78% of study participants believed vehicle

hacking “will be a frequent problem in the next 3 years or less” (PR Newswire 2015). This perception among the general public is mostly a result of several recent high-profile vehicle hacks. The timeline below summarizes some of the more notable vehicle hacks that have recently occurred.

6.6.1 2010: Vehicles Disabled Remotely via Web Application

One of the first widely reported accounts of vehicle hacking occurred in 2010 when a disgruntled former employee of an Austin, Texas, car dealership sought revenge against his former employer (Poulsen 2010). This attack did not involve any hacking of the actual vehicles themselves. Nonetheless, the attacker was able to physically disable the vehicles of owners without their knowledge or consent. The former dealership employee used stolen credentials to log into a web application that allowed remote access to functions of customers’ vehicles, including the engine immobilizer and the horn (Poulsen 2010). This web application’s intended purpose was to let dealership personnel immobilize the vehicles of customer who failed to make their loan payments on time. In fact, it ended up being used to cause mayhem as vehicle owners found themselves locked out of their vehicles with the horns constantly honking (Poulsen 2010).

The web application used by the dealership, in this case, was WebTeckPlus from Pay Technologies, LLC (Payteck 2003). The WebTeckPlus application provides a web portal for dealership employees to interface with PayTeck electronic controllers installed in customers’ vehicles. The PayTeck hardware consists of an electronic keypad and controller that is installed inside the customer’s vehicle. The controller is wired into the vehicle’s engine immobilizer and horn. Each time a customer makes a payment on time, they are given a new code to enter into the electronic keypad. If the correct code is entered, the vehicle will continue to function normally. If payment is late and a code is not entered into the keypad on time, the controller will activate the engine immobilizer, rendering the vehicle useless. The WebTeckPlus application also allows a dealership employee to log in and remotely disable a particular customer’s vehicle at will, if necessary. The PayTeck hardware and WebTeckPlus software allow dealerships to save time and money by avoiding having to repossess vehicles.

This hack can best be summarized as an unauthorized intrusion of a web-based application, which is certainly nothing new. The perpetrator faced computer intrusion charges (Poulsen 2010). However, this particular incident highlights the link between a vehicle’s critical control systems and the digital transformation of the modern connected world, showing how that link can potentially be exploited by someone with malicious intent.

6.6.2 2010 and 2011 CAESS Experimental Analysis

In 2010, a group of researchers from the Center for Automotive Embedded Systems Security (CAESS) – a joint venture between the University of California, San Diego and the University of Washington – released a research paper entitled *Experimental Security Analysis of a Modern Automobile* (Koscher et al. 2010). The team conducted a range of lab experiments and road tests and found that it was possible to manipulate a vehicle’s functions by injecting messages on the CAN bus (Koscher et al. 2010). The researchers successfully demonstrated that a would-be attacker could disable the brakes, selectively brake individual wheels on demand, stop the engine, falsify information on the vehicle’s speedometer, and more (Koscher et al. 2010).

Although the CAESS team highlighted serious security flaws in a modern vehicle system, their research was largely met with criticism. At the time, automakers and the media alike claimed that it was neither realistic nor plausible for an attacker to have wired access to a vehicle’s CAN bus to be able to carry out this type of attack in the real world (Miller and Valasek 2014, 2015, p. 5ff).

The following year, in 2011, the CAESS team published a new research paper entitled *Comprehensive Experimental Analyses of Automotive Attack Surfaces* (Checkoway et al. 2011). This paper was a response to the media scepticism surrounding the team’s previous findings. The team acknowledged that the previous threat model of an attacker having physical access to a vehicle’s internal network had *justifiably been viewed as unrealistic* (Checkoway et al. 2011). This time, the researchers sought to analyze the external attack surface of a modern vehicle and determine whether an attack could be carried out remotely.

In analyzing the attack surface of a modern car, the CAESS team created the illustration shown in Fig. 6.30.

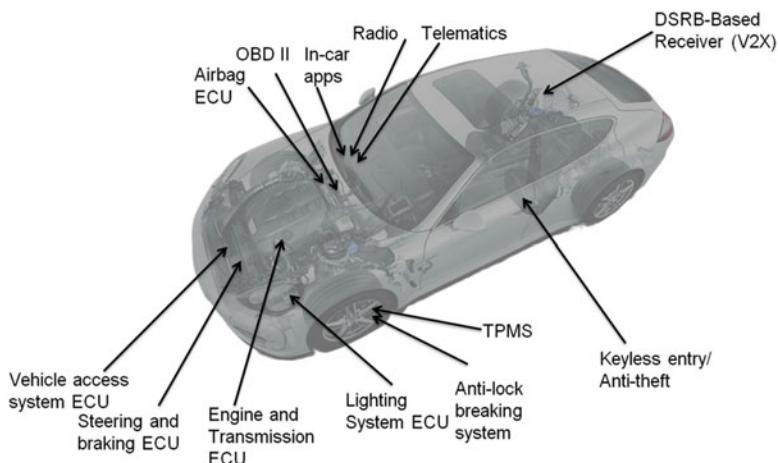


Fig. 6.30 Digital I/O channels on a modern vehicle

The Fig. 6.30 shows the different I/O channels on a modern vehicle, with each one representing a potential entry point for an attacker. The “lightning bolt” symbols represent possible sources of remote wireless access and control. As automakers continue to increase the connectivity of their vehicles, the attack surface only broadens. The vehicle’s cellular, Bluetooth, and Wi-Fi systems make particularly attractive entry points for a would-be attacker.

Ultimately, the CAESS team found that it was possible to remotely exploit their test vehicle via a range of different vectors, including the radio’s MP3 parser, the vehicle’s Bluetooth system, and the cellular connection used for the vehicle’s telematics system (Checkoway et al. 2011). From there, CAN messages could be injected on the bus as had been demonstrated in the group’s previous findings.

This research was hailed by some as ground breaking because “it showed that vehicles were vulnerable to attacks from across the country, not just locally” (Miller and Valasek 2015, p. 5ff). Despite having answered their critics, the CAESS team’s findings failed to garner much media attention or response from the automotive industry. This was due in part to the fact that the researchers did not share how their exploits could be replicated, nor did they reveal the specific vehicle they tested (Miller and Valasek 2015, p. 5ff). While it is understandable that the research team would choose not to release the details of their exploits so as not to aid the “bad guys,” this also made the findings a lot easier for automakers and the general public to shrug off.

6.6.3 2013 Miller and Valasek Physical Hack

A more recent high-profile case of vehicle hacking came from researchers Charlie Miller and Chris Valasek. Working with an \$80,000 grant from the Defense Advanced Research Projects Agency (DARPA), Miller and Valasek were tasked with finding security vulnerabilities in automobiles and published their findings in 2013 (Greenberg 2013). They conducted a series of real-time demonstrations for journalists and security professionals, before going on to present their findings at the 2013 DEF CON® 21 Hacking Conference in Las Vegas, Nevada. Specifically, Miller and Valasek targeted the systems of a 2010 Ford® Escape and a 2010 Toyota® Prius (Greenberg 2013). They were essentially able to reverse engineer the vehicles’ CAN bus communications to demonstrate “everything from annoyances like uncontrollably blasting the horn to serious hazards like slamming on the Prius’ brakes at high speeds” (Greenberg 2013). The graphic in Fig. 6.31 lists many of the vehicular functions that Miller and Valasek were able to manipulate on their 2010 Toyota Prius test vehicle.

Some of these capabilities, for example, being able to jerk the steering wheel or slam on the brakes, propelled car hacking from a nuisance to a serious safety concern for automakers.

Miller and Valasek’s method involved using a laptop PC running Windows XP hooked into the vehicle’s OBD-II port via a series of cables (Miller and Valasek 2015, p. 23). The OBD-II port is traditionally used by mechanics and repair shops to

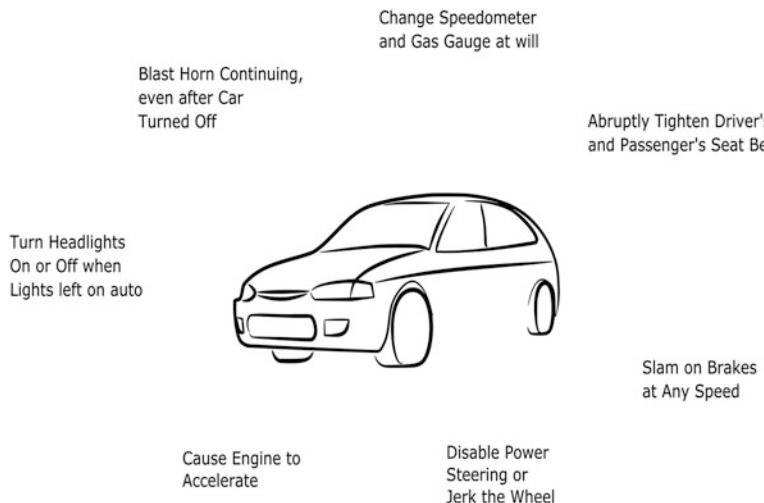


Fig. 6.31 Anatomy of an Automotive Hack (Greenberg 2013)

retrieve fault codes and diagnose problems with a vehicle, but it also represents an attractive point of entry for a vehicle security researcher or an attacker performing reconnaissance (see also Sect. 6.2.2). Miller and Valasek used a proprietary ECOM cable from EControls which was hooked to their laptop via a USB port. They then fashioned a custom ECOM-to-OBD-II connector to allow them to interface with the car's OBD-II port (Miller and Valasek 2015, p. 22). At this point, all Miller and Valasek had to do was listen and observe the CAN messages transiting the CAN bus to begin building a picture of which message corresponded to which vehicular function. The next step was to use the connected laptop to replay captured CAN packets, recording the vehicle's response each time. Finally, they crafted modified CAN packets and were able to manipulate the behavior of the vehicle (Miller and Valasek 2014, p. 26).

It is worth emphasizing again that in Miller and Valasek's 2013 car hacking demonstration, the researchers had physical access to the vehicle's CAN bus. The rationale behind this was that, according to the researchers, it had already been shown by prior scholarly research (Checkoway et al. 2011) that various interfaces, such as Bluetooth or a vehicle's telematics unit, could be hacked to allow for remote code execution (Miller and Valasek 2015, p. 4ff). Considering the challenge of gaining remote access to be trivial, the researchers sought to find out what could be accomplished after access had been gained (Miller and Valasek 2015, p. 4ff).

Following the release of Miller and Valasek's findings in 2013, the general public and big automakers seemingly failed to recognize the triviality of the prerequisite of gaining remote access to a vehicle's systems. Miller and Valasek faced scepticism for having demonstrated security flaws that required an attacker to be physically

located inside the vehicle with a laptop hooked up to the car's data port and, as in the case of the Prius demonstration, with the dashboard completely disassembled for ease of access (Greenberg 2013). Indeed, in response to Miller and Valasek's work, Toyota's safety manager, John Hanson, argued that "[Toyota's] focus, and that of the entire auto industry, is to prevent hacking from a remote wireless device outside of the vehicle" (cited in Greenberg 2013), indicating that Toyota was largely unimpressed by the hacking demonstration. Hanson went on to state, "we believe our systems are robust and secure" (cited in Greenberg 2013).

6.6.4 2015 Miller and Valasek Remote Hack

Charlie Miller and Chris Valasek made headlines again in 2015, this time for successfully demonstrating that an unaltered passenger vehicle – a 2014 Jeep® Cherokee, in this case – could be remotely exploited without the need for any physical access (Miller and Valasek 2015, p.6ff). Unlike their 2013 hack of a Toyota Prius and Ford Escape, this new research mimicked a real-world attack scenario by demonstrating both the ability to gain remote access and the ability to remotely execute code. Unlike the 2013 hack, which was largely met with incredulity by automakers, the 2015 hack prompted Fiat® Chrysler® Automobiles (FCA) to recall some 1.4 million vehicles for a critical security update and forced Sprint® Corporation to enhance the security of its cellular carrier network (Miller and Valasek 2015, p.87).

Miller and Valasek's Jeep hack took advantage of the vehicle's onboard connectivity features, in addition to the familiar lack of security controls on the CAN bus. Access was obtained through vulnerability in Uconnect®, a system that governs the vehicle's infotainment, navigation, built-in apps, and cellular communications (Greenberg 2015a). What made the Uconnect system so attractive to the pair of researchers was that in addition to being a hotbed of connectivity, Uconnect also contains a microcontroller in its head unit which can communicate with other modules on the vehicle's CAN bus (Miller and Valasek 2015, p.20). The hack also took advantage of a weakness in Sprint's cellular network, to which the vehicle's onboard telematics system was connected. The telematics system is used for real-time traffic data, in-car Wi-Fi, and other remote connectivity functions (Miller and Valasek 2015, p.32).

Through port scanning, they found Uconnect's D-Bus port (6667) to be open. D-Bus, also known as Diagnostic Bus, is a messaging system used to communicate between processes (Miller and Valasek 2015, p.28). Under normal conditions, the D-Bus service should not be subject to user input or manipulation, as it is intended for internal systems messages only. Miller and Valasek then found that prior to Sprint's fix, any 3G device on the Sprint network could communicate with the open D-Bus port on any Uconnect-enabled vehicle (Miller and Valasek 2015, p.46). For their attack, Miller and Valasek used a laptop computer tethered to a 3G cellular phone on the Sprint network. The laptop was then able to communicate directly with vehicles running the vulnerable Uconnect system (Miller and Valasek 2015, p.46).

Knowing the IP address of a specific vehicle allowed for a targeted attack; however, they also found that an Internet port scan of port 6667 across IP ranges 21.0.0.0/8 and 25.0.0.0/8 would yield responses from vulnerable Uconnect systems in vehicles nationwide (Miller and Valasek 2015, p.46). The researchers' scans of Internet-facing vulnerable devices turned up a wide range of vehicles across the country, from the Dodge, Ram, Jeep, and Chrysler brands, spanning multiple model years (Miller and Valasek 2015, p.47).

With access to the vehicle's Uconnect system obtained, Miller and Valasek then pivoted to the CAN-connected microcontroller in the Uconnect head unit. They were able to flash the controller with a new firmware version, one that they had reverse engineered to include their malicious code (Miller and Valasek 2015, p.50). With their modified firmware residing on the CAN bus, they were then able to send commands to many different vehicle components and control systems. During a press demonstration, Miller and Valasek showed that they were able to remotely set the air conditioning to its maximum cold setting, turn the radio on at full volume, and cover the windshield with wiper fluid making it difficult for the driver to see. More worryingly, they could also disable the transmission, control the throttle, and disable the brakes (Greenberg 2015a).

This latest automotive hacking demonstration propelled vehicle security into the general public's consciousness in a way that had not been seen previously. Shortly after news of the Jeep Cherokee hack hit the media, a Kelley Blue Book study of the car-buying public found that 72% of respondents were "aware of the recent Jeep Cherokee hacking incident" (PR Newswire 2015). Perhaps more tellingly, 41% of respondents said they would "consider this recent vehicle hacking incident when buying/leasing their next car" (PR Newswire 2015).

For the first time, an automotive hack had the very real potential to cost a large automaker a significant amount of money. Fiat Chrysler Automobiles, facing a reputation hit and possible loss of future customers, made the wise but costly decision to patch any vehicles that were vulnerable to Miller and Valasek's exploit. By some estimates, the amount which this critical security update costed FCA in labor hours alone was in excess of \$10 million (Cobb 2015). Miller and Valasek have long stated that their shared goal has been to provide their research to the automotive industry and security community "so that we can learn to build more secure vehicles in the future, so that drivers can trust they are safe from a cyber attack" (Miller and Valasek 2015, p.88). Certainly, hitting an automaker's bottom line is an effective way to accomplish this greater good.

6.7 Exercises

What is meant by the term *digital transformation*?

Give an example of the characteristics of digital transformation.

What is meant by the term *information technology*?

Give an example of the characteristics of information technology.

What is meant by the term *cybersecurity*?

Give an example of the characteristics of cybersecurity.

What is meant by the term *application security*?

Give an example of the characteristics of application security.

What is meant by the term *information security*?

Give an example of the characteristics of information security.

What is meant by the term *network security*?

Give an example of the characteristics of network security.

What is meant by the term *security threats*?

Give an example of the characteristics of security threats.

What is meant by the term *countermeasures* with regard to *cybersecurity*?

Give an example of the characteristics of countermeasures with regard to cybersecurity.

What is meant by the term *likelihood of risk*?

Give an example of the characteristics of likelihood of risk.

What is meant by the term *risk management in cybersecurity*?

Give an example of the characteristics of risk management in cybersecurity.

What is meant by the term *security risk*?

Give an example of the characteristics of security risks.

What is meant by the term *vulnerability*?

Give an example of the characteristics of vulnerability.

What is meant by the term *vulnerable space*?

Give an example of the characteristics of a vulnerable space.

What is meant by the term *vulnerable access points*?

Give an example of the characteristics of vulnerable access points.

What is meant by the term *cyber attack*?

Give an example of the characteristics of cyberattacks.

What is meant by the term *anomaly detection*?

Give an example of the characteristics of anomaly detection.

What is meant by the term *denial of service*?

Give an example of the characteristics of denial of service.

What is meant by the term *artificial intelligence*?

Give an example of the characteristics of artificial intelligence.

What is meant by the term *control theory*?

Give an example of the characteristics of control theory.

What is meant by the term *epidemic theory*?

Give an example of the characteristics of epidemic theory.

What is meant by the term *game theory*?

Give an example of the characteristics of game theory.

What is meant by the term *graph theory*?

Give an example of the characteristics of graph theory.

What is meant by the term *probabilistic dependence graph*?

Give an example of the characteristics of a probabilistic dependence graph.

What is meant by the term *logic bomb*?

Give an example of the characteristics of a logic bomb attack.

What is meant by the term *Trojan horse*?

Give an example of the characteristics of Trojan horses.

What is meant by the term *virus*?

Give an example of the characteristics of viruses.

What is meant by the term *worm*?

Give an example of the characteristics of worms.

What is meant by the term *vehicle-to-infrastructure*?

Give an example of the characteristics of vehicle-to-infrastructure.

What is meant by the term *vehicle-to-mobile*?

Give an example of the characteristics of vehicle-to-mobile.

What is meant by the term *vehicle-to-vehicle*?

Give an example of the characteristics of vehicle-to-vehicle.

What is meant by the term *OEM*?

Give an example of the characteristics of OEMs.

What is meant by the term *remote hacking*?

Give an example of the characteristics of remote hacking.

What is meant by the term *attack value chain*?

Give an example of the characteristics of attack value chains.

What is meant by the term *man-in-the-middle attack*?

Give an example of the characteristics of a man-in-the-middle attack.

What is meant by the term *compromised-key attack*?

Give an example of the characteristics of a compromised-key attack.

What is meant by the term *electronic control unit*?

Give an example of the characteristics of an electronic control unit.

What is meant by the term *CAN*?

Give an example of the characteristics of CAN.

What is meant by the term *cyberattack taxonomy*?

Give an example of the characteristics of a cyberattack taxonomy.

What is meant by the term *attack surface*?

Give an example of the characteristics of attack surfaces.

What is meant by the term *onboard diagnostics*?

Give an example of the characteristics of onboard diagnostics.

What is meant by the term *vulnerability scanning*?

Give an example of the characteristics of vulnerability scanning.

What is meant by the term *intrusion detection*?

Give an example of the characteristics of intrusion detection.

What is meant by the term *intrusion prevention*?

Give an example of the characteristics of intrusion prevention.

What is meant by the term *WLAN security*?

Give an example of the characteristics of WLAN security.

What is meant by the term *sensor node security*?

Give an example of the characteristics of sensor node security.

What is meant by the term *WEP authentication*?

Give an example of the characteristics of WEP authentication.

What is meant by the term *platform security*?

Give an example of the characteristics of platform security.

What is meant by the term *cloud computing*?

Give an example of the characteristics of cloud computing.

What is meant by the term *functional safety*?

Give an example of the characteristics of functional safety.

What is meant by the term *mean time between failure*?

Give an example of the characteristics of mean time between failure.

What is meant by the term *mean address spoofing*?

Give an example of the characteristics of an address spoofing.

What is meant by the term *eavesdropping*?

Give an example of the characteristics of eavesdropping.

What is meant by the term *medium access control*?

Give an example of the characteristics of medium access control.

What is meant by the term *false node*?

Give an example of the characteristics of false nodes.

What is meant by the term *platform security*?

Give an example of the characteristics of platform security.

What is meant by the term *insiders*?

Give an example of the characteristics of insiders.

What is meant by the term *outsiders*?

Give an example of the characteristics of outsiders.

What is meant by the term *Byzantine model*?

Give an example of the characteristics of the Byzantine model.

What is meant by the term *mean time between failure*?

Give an example of the characteristics of mean time between failure.

What is meant by the term *SIL*?

Give an example of the characteristics of SIL.

What is meant by the term *ASIL*?

Give an example of the characteristics of ASIL.

What is meant by the term *car hacking*?

Give an example of the characteristics of car hacking.

References and Further Reading

- (Akella et al. 2010) Akella, R., Tang, H., McMillin, B.: Analysis of Information Flow Security in Cyber-Physical Systems. In: Internat. Journal of Critical Infrastructure Protection, Vol. 3, pp. 157–173, 2010
- (Akyildiz et al. 2002) Akyildiz, I. E., Su, W., Sankkarasubramaniam, Y., Cayirci, E.: Wireless Sensor Networks: A Survey. In: Comput. News, Vol. 16, No. 4, pp 393–402, 2002
- (Avancha 2005) Avancha, S.: A Holistic Approach to Secure Sensor Networks. Ph. D. thesis, 2005
- (Barika, et al. 2010) Barika, F., Hadjar, K., El-Kadhi, N.: Artificial neural network for mobile IDS solution, In: Security and Management, pp. 271–277, 2010
- (Bitter et al. 2010) Bitter, C., Elizondo, D. A., Watson, T.: Application of Artificial Neural Networks and Related Techniques to Intrusion Detection. In: IEEE World Congress on Computational Intelligence, pp. 949–954, IEEE Press 2010.
- (Bittersohl and Thoppill 2015) Bittersohl, C., Thoppil, T. G.: Automotive Cyber Security, P3 Inc., 2015

- (Brown 1985) Broqn, J.: An Introduction to the Use of Facet Theory. In: Facet Theory, pp. 17–57, Springer Publ. 1985
- (Bruton 2014) Bruton, J. A.: Securing CAN Bus Communication: An Analysis of Cryptographic Approaches. Master Thesis National University of Ireland, Galway, 2014
- (Butayán and Hubaux 2007) Butayán, L., Hubaux, J.-P.: Security and Cooperation in Wireless Networks. Cambridge University Press, 2007
- (CAMP05 2005) CAMP05 Vehicle Safety Communications Consortium. Vehicle Safety Communications Project Task 3 Final Report 2005. <http://www.intellidriveusa.org/documents/vehicle-safety.pdf>
- (CAMP09 2008) CAMP09 Vehicle Safety Communications Consortium. Vehicle Safety Communications – Applications 1st Annual Report, Sept. 2008. <http://www.intellidriveusa.org/documents/09042008-vsc-a-report.pdf>
- (CAMP10 2008) CAMP10 Vehicle Safety Communications Consortium. Cooperative Intersection Collision Avoidance System Limited to Stop Sign and Traffic Signal Violations Midterm Phase I Report, Oct. 2008. <http://www.nhtsa.dot.gov/staticfiles/DOT/NHTSA/NRD/Multimedia/PDFs/Crash%20Avoidance/2008/811048.pdf>
- (Cárdenas et al. 2008) Cárdenas, A., Amin, S., Sastry, S.: Secure Control - Towards Survivable Cyber-Physical Systems. Proceed. 28th IEEE International Conference on Distributed Computing Systems Workshops, pp. 495–500, 2008
- (Cárdenas et al. 2011) Cárdenas, A., Amin, S., Lin, Z., Huang, Y., Huan, C., Sastry, S.: Attacks against Process Control Systems: Risk Assessment, Detection, and Response. Proceed. 6th ACM Symposium on Information, Computer and Communications Security, pp. 355–366, 2011
- (Cebula and Young 2010) Cebula, J., Young, L. R.: A Taxonomy of Operational Cyber Security Risks. Software Engineering Institute Technical Note CMU/SEI-2010-TN-028, 2010
- (Chakrabarti et al. 2007) Chakrabarti, D., Leskovec, J., Faloutsos, C., Madden, S., Guestin, C., Faloutsos, M.: Information Survival Threshold in Sensor and P2P Networks. In: INFOCOMM, IEEE, pp. 1316–1324, 2007
- (Chalkias et al. 2009) Chalkias, K., Baldimtsi, F., Hristu-Varsakelis, D., Etephanides, G.: Two Types of Key-Compromise Impersonation Attacks against One-Pass Key Establishment Protocols. In: Communications in Computer and Information Science, Vol. 23, Part 3, pp. 227–238, 2009
- (Chatterjee 2012) Chatterjee, P.: The Connected Car as a Platform. In: EDN Network, December 2012
- (Checkoway et al. 2011) Checkoway, S., McCoy, D., Kantor, B., Anderson, D., Shacham, H., Savage, S., Koscher, K., Czeskis, A., Roesner, F., Kohno, T.: Comprehensive Experimental Analysis of Automotive Attack Surfaces. <http://www.autose.org/pubs/cars-usenixsec2011.pdf>
- (Cichonsky et al. 2012) Cichonsky, P., Millar, T., Grance, T., Scarfone, K.: Computer Security Incident Handling Guide. National Institute of Standards and Technology (NIST) Special Publication 800-61, Revision 2, 2012
- (Cobb 2015) Cobb, S.: Cybersecurity and Manufacturers: What the Costly Chrysler Jeep Hack Reveals. <http://www.welivesecurity.com/2015/07/29/cybersecurity-manufacturing-chrysler-jeep-hack/>
- (Currie 2015) Currie, R.: Developments in Car Hacking. SANS Institute 2015. <https://www.sans.org/reading-room/whitepapers/ICS/developments-car-hacking-36607>
- (Daley and Gani 1999) Daley, D. J., Gani, J.: Epidemic Modelling: An Introduction. Cambridge University Press, 1999
- (Das et al. 2012) Das, S. K., Kant, K., Zhang, N.: Handbook on Securing Cyber-Physical Critical Infrastructure. Elsevier Publ. 2012
- (De Capitani di Vimercati 2007) De Capitani di Vimercati, S., Foresti, S., Jajodia, S., Paraboschi, S., Samarati, P.: Over Encryption: Management of Access Control Evolution on Outsourced Data. In: Proc. of VLDB, pp. 123–134. 2007
- (Denning 1987) Denning, D. E.: An Intrusion Detection Model. In: IEEE Transactions on Software, Vol: SE-13 Issue: 2, pp. 222–232, 1987

- (Dilek et al. 2015) Dilek, S., Caku, H., Aydin, M.: Applications of Artificial Intelligence Techniques to Combating Cyber Crimes - A Review. *Internat. J. of Artificial Intelligence and Applications (IJAIA)*, Vol. 6, No. 11, pp. 21–39, 2015
- (Dolev 1982) Dolev, D.: The Byzantine Generals Strike Again. *Journal of Algorithms*, Vol. 3(1), pp.14–30, 1982
- (Eisenhauer et al. 2006) Eisenhauer, J., Donelly, P., Ellis, M., O'Brien, M.: Roadmap to Secure Control Systems in the Energy Sector. *Energetics Inc. Columbia, MD*, 2006
- (Eugster et al. 2004) Eugster, P. T., Guerraoui, R., Kermarrec, A., Massouli, L.: From Epidemics to Distributed Computing. In: *IEEE Computer*, Vol. 37, pp. 60–76, 2004
- (Eyal 2007) Eyal, N.: Vehicle Lab – Engine Control Unit, 2007. <http://www.vehicle-lab.net/ecu.html>
- (Falliere et al. 2011) Falliere, N., O'Murchu, L., Chien, E.: W32. Stuxnet Dossier. Symantec Corporation, 2011
- (Finke et al. 2015) Finke, T., Schoop, D., Melcher, H.: Extension of Security AUTOSAR architecture to recognition and Countermeasures in terms of relevant attack scenarios Automotive Ethernet. Thesis Work in German; University of Applied Sciences Esslingen, 2015
- (Fleury et al. 2009) Fleury, T., Khurana, H., Welch, V.: Towards Taxonomy of Attacks against Energy Control Systems. Proceed. 2nd Annual IFIP Working Group. Internat. Conference on Critical Infrastructure Protection, pp. 71–85, 2009
- (Gamage and McMillian 2009) Gamage, T., McMillin, B.: Enforcing Information Flow Properties using Compensating Events. In: Proceed. 42nd Hawaii Internat. Conference on System Sciences, pp. 1–7, 2009
- (Goh et al. 2003) Goh, E., Shacham, H., Modadugu, N., Boneh, D.: SiRiUS: Securing Remote Untrusted Storage. In: Proc. of NDSS, pp. 131–145, 2003
- (Goodfellow et al. 2016) Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, 2016. www.deeplearningbook.org
- (Goodwin 2009) Goodwin, A.: Ford Unveils Open-Source Developer Platform. 2009. http://reviews.cnet.com/8301-13746_7-10385619-48.html, Oct. 2009
- (Gordon and Ford 2006) Gordon, S., Ford, R.: On the Definition of Classification of Cybercrime. *Journal in Computer Virology*, Vol.2, No. 1, pp. 13–20, 2006
- (Greenberg 2013) Greenberg, A.: Hackers Reveal Nasty New Car Attacks-With me Behind the Wheel. <http://www.forbes.com/sites/andygreenberg/2013/07/24/hackers-reveal-nasty-new-car-attacks-with-mebbehind-the-wheel-video/>
- (Greenberg 2015) Greenberg, A.: Hackers Remotely Kill a Jeep on the Highway-With me in it. <http://www.wired.com/2015/07/hackersremotely-kill-jEEP-highway/>
- (Gupta 2016) Gupta, V.: Control of Cyber-Physical Systems: Recent Results and New Challenges, 2016; http://www.ieeeCSS-oll.org/sites/default/files/final_gupta_acc.pdf
- (Guttmann and Roback 1995) Guttmann, B., Roback, E. A.: An Introduction to Computer Security: The NIST Handbook. DIANE Publ. 1995
- (Hamlen et al. 2006) Hamlen, K., Morrisett, G., Schneider, F.: Computability classes for enforcement mechanisms. In: *ACM Transactions on Programming Languages and Systems*, Vol. 28, No. 1, pp. 175–205; 2006
- (Hansman and Hunt 2005) Huntsman, S., Hunt, R.: A Taxonomy of Network and Computer Attacks. In: *Computers and Security*, Vol. 24, Issue 1, pp. 31–43, 2005
- (Heady et al. 1990) Heady, R., Luger, G., Maccabe, A., Servilla, M.: The Architecture of a Network Level Intrusion Detection System. Technical Report University of New Mexico, Department of Computer Science, 1990
- (Housley and Arbaugh 2003) Housley, R., Arbaugh, W.: Security Problems in 802.11-based Networks. In: *Commun. ACM* Vol. 46, No. 5, pp. 21–34, 2003
- (Hubaux et al. 2004) Hubaux, J. P., Chapkun, S., Luo, J., Raya, M.: The Security and Privacy of Smart Vehicles. In: *Journal IEEE Security and Privacy*, Vol. 2, No. 3, pp. 49–55, 2004
- (Intel Security 2015) Intel Security White Paper Automotive Security Best Practice. 2015; <http://www.mcafee.com/de/resources/white-papers/wp-automotive-security.pdf>

- (IXIA 2014) IXIA Securing the Connected Car, Whitepaper 915–3513-01 Rev. A, 2014: www.ixiacom.com
- (Jin et al. 2012) Jin, X., Dan, M., Zhang, N., Yu, W., Fu, X., Das, S. K.: Game Theory for Infrastructure Security: The Power of Intent-Based Adversary Models. In: Das, S. K., Kant, K., Zhang, N.: Handbook on Securing Cyber-Physical Critical Infrastructure, pp. 31–53. Morgan Kaufmann Publ., 2012
- (Johnson 2010) Johnson, T.: Fault-Tolerant Distributed Cyber-Physical Systems: Two Case Studies. Master Thesis University of Illinois, ECE Dept., 2010
- (Johnson 2016) Johnson, M.: Cyber Crime, Security and Data Intelligence. Routledge Publ. 2016
- (Kallahalla et al. 2003) Kallahalla, M., Riedel, E., Waminadham, R., Wang, Q., Fu, K.: Scalable Secure File Sharing on Untrusted Storage. In: Proc. of 2nd USENIX Conference of File and Storage Technologies, pp. 29–42. 2003
- (Kao and Marculescu 2006) Kao, J. C., Marculescu, R.: Eavesdropping Minimization via Transmission Power Control in Ad-Hoc Wireless Networks. In: 3rd Annual IEEE Communications Society on Sensor and Ad-Hoc Communications and Networks, pp. 707–714, 2006
- (Karim and Proha 2014) Karim, E., Proha, V. V.: Cyber-Physical Systems Security. In: Applied Cyber-Physical Systems, pp. 75–84. Eds.: Shuh, S. S., Tanik, U., J., Carbone, J. N., Rogglin, A.; Springer Publ., 2014
- (Kephart and White 1993) Kephart, J. O., White, S. R.: Measuring and Modeling Computer Virus Prevalence. In: Proceed. IEEE Symposium on Security and Privacy, pp. 2–15, 1993
- (Kermack and McKendrick 1927) Kermach, W. O., McKendrick, A.: A Contribution to the Mathematical Theory of Epidemics. Proceed. Royal Society of London, Vol. A, No. 1, pp. 700–721, 1927
- (Kjaerland 2005) Kjaerland, M.: A Taxonomy and Comparison of Computer Security Incidents for the Commercial and Government Sectors. In: Computers and Security, Vol. 25, pp. 522–538, 2005.
- (Koscher et al. 2010) Koscher, K., Czeskis, A., Roesner, F., Patel, S., Kohno, T., Checkoway, S., McCoy, D., Kantor, B., Anderson, D., Shacham, H., Savage, S.: Experimental Security Analysis of a Modern Automobile. In: IEEE Symposium on Security and Privacy, pp. 448–461, 2010
- (Kumar and Spafford 1994) Kumar, S., Spafford, E. H.: An Application of Pattern Matching in Intrusion Detection. Computer Science Technical Reports, Paper 1116, Purdue University, 1994
- (Landrum et al. 2014) Landrum, R., Pace, S., Hu, F.: Cyber-Physical Systems Security–Smart Grid Example, pp. 135–154. In: Cyber-Physical Systems. Ed.: F. Hu. CRC Press 2014
- (Lamport 1997) Lamport, L.: Proving the Correctness of Multiprocessing Programs. In: IEEE Transactions on Software Engineering, Vol. 3(2), pp. 125–143, 1997
- (Lamport 1998) Lamport, L.: Proving Possibility Properties. In: Theoretical Computer Science, Vol. 206(1–2), pp. 341–352, 1998
- (Lamport 2005) Lamport L.: Real-Time Model Checking is Really Simple. Proceed. 13th Advanced Research Working Conference on Correct Hardware Design and Verification Methods, pp. 162–175, 2005
- (Landram et al. 2014) Landram, R., Pace, S., Hu, F.: Cyber-Physical System Security - Smart Grid Example. In: F. Hu: Cyber-Physical Systems - Integrated Computing and Engineering Design. pp. 145–154, CRC Press 2014
- (Lin and Sangiovanni-Vincentelli 2012) Lin, C. W., Sangiovanni-Vincentelli, A.: Cyber-Security for the Controller Area Network (CAN) Communication Protocol. In: IEEE Proceed. Internat. Conference on Cyber Security, pp. 1–7, 2012
- (Lin et al. 2013) Lin, C. W., Zhu, Q., Phung, C., Sangiovanni-Vincentelli, A.: Security-aware mapping for CAN-based real-time distributed automotive systems. In: IEEE Proceed. Internat. Conference on Cyber Security, pp. 115–121, 2013
- (Lough 2001) Lough, G. L.: A Taxonomy of Computer Attacks with Applications to Wireless Networks. Dissertation submitted to the Faculty of the Virginia Polytechnic Institute, 2001
- (Lunt et al. 1992) Lunt, T. F., Tamaru, A., Gilham, F., Jagannathan, R., Neumann, P. G., Javitz, H. S., Valdes, A., Garvey, T. D.: A Real-Time Intrusion Detection Expert System (IDES) –

- Final technical Report, SRI Computer Science Laboratory, SRI International, Menlo Park, CA, 1992
- (Luo et al. 2010) Luo, Y., Szidarovsky, F., Al-Nashif, Y., Hariti, S.: Game Theory Based Network Security. In: Journal of Information Security, pp. 41–44, 2010
- (Miller and Valasek 2014) Miller C., Valasek C.: A Survey of Remote Automotive Attack Surfaces. IOActive 2014. Available from: https://www.ioactive.com/pdfs/IOActive_Remote_Attack_Surfaces.pdf
- (Miller and Valasek 2015) Miller, C., Valasek, C.: Remote Exploitation of an Unaltered Passenger Vehicle. <http://illmatics.com/Remote%20Car%20Hacking.pdf>
- (Mo and Sinopoli 2009) Mo, Y., Sinopoli, B.: Secure Control against Replay Attacks. Proceed. 47th Conf. on Communication, Control, and Computing, pp. 911–918, 2009
- (Mollman 2009) Mollmann S.: From Cars to TVs, Apps are Spreading to the Real World. <http://edition.cnn.com/2009/TECH/10/08/apps.realworld/>
- (Möller 2016) Möller, D. P. F.: Guide to Computing Fundamentals in Cyber-Physical Systems – Concepts, Design Methods, and Applications, Springer Publ., 2016
- (ni-com 2009) ECU Designing and Testing Using National Instruments Products. White Paper, National Instruments 2009
- (Nurse et al. 2014) Nurse, J. R. C., Buckley, O., Legg, P. A., Goldsmith, M., Cresse, S., Wright, G. R., Whitey, M.: Understanding Insider Threat: A Framework for Characterizing Attacks. IEEE Security and Privacy Workshops, pp. 214–222, IEEE 2014
- (Patel et al. 2010) Patel, A., Qassim, Q., Shukor, Z., Nogueira, J., Junior, J., Wills, C.: Autonomic Agent-Based Self-Managed Intrusion Detection and Prevention System, In: Proceed. South African Information Security Multi-Conference, pp. 223–234, 2010
- (Pathan et al. 2006) Pathan, Al-S. K., Lee, H.-W., Hong, C. S.: Security in Wireless Sensor Networks: Issues and Challenges. In: Proceed. Internat. Confer. Advanced Technology, pp. 1043–1048, 2006
- (Payteck 2003) How PayTeck Works. www.payteck.cc/aboutpayteck.html
- (Pfleeger et al. 2015) Pfleeger, C. P., Pfleeger, S. L., Margulies, J.: Security in Computing. Prentice Hall 2015
- (Pelechrinis et al. 2011) Pelechrinis, K., Iliofoitou, M., Krishnanurthy, S. V.: Denial of Service Attacks in Wireless Networks: The Case of Jammers. In: IEEE Communications Surveys and Tutorial, Vol. 13, No. 2, pp. 245–257, 2011
- (Poulsen 2010) Poulsen, K.: Hacker disables more than 100 cars remotely. Wired online. March 17th 2010. Available from: www.wired.com/threatlevel/2010/03/hacker-bricks-cars
- (PR Newswire 2015) <https://www.prnewswire.com/news-releases/nearly-80-percent-of-consumers-think-vehicle-hacking-will-be-frequent-problem-in-near-future-according-to-new-kelley-blue-book-survey-300121740.html>
- (Salahuddin and Al-Fuqaha 2013) Salahuddin M. A. B., Al-Fuqaha, A.: AGORA: A Versatile Framework for the Development of Intelligent Transportation System Applications. In: Wireless Sensor and Mobile Ad-Hoc Networks: Vehicular and Space Applications, pp. 163–184, Eds.: B. Benhaddou, A. Al-Fuqaha, Springer Publ. 2013
- (Saleh and Khatib 2005). Saleh, M., Khatib, I. A.: Throughput Analysis of WEP Security in Ad Hoc Sensor Networks. In: Proc. 2nd International Conference on Innovations in Information Technology, 2005
- (Saltzman and Sharabani 2009) Saltzman, R., Sharabani, A.: Active Man in the Middle Attacks – A Security Advisory. Whitepaper IBM Rational Application Security Group. IBM Corporation 2009
- (Satyanarayanan et al. 2009) Satyanarayanan, M., Bahl, P., Caceres, R., Davies, N.: The Case for VM-based Cloudlets in Mobile Computing. IEEE Pervasive Compt. Vol. 8 No. 4, 14–23, 2009
- (Sastry et al. 1994) Sastry, P. S., Phansalpar, V. V., Thathachar, M. A. L.: Decentralized Learning of Nash Equilibria in Multi-Person Stochastic Games with Incomplete Information. In: IEEE Transact. On Systems, Man, and Cybernetics, Vol. 24, No. 5, pp. 769–777, 1994

- (Scarfone and Mell 2007) Scarfone K., Mell, P.: Guide to Intrusion Detection and Prevention Systems. National Institute of Standards and Technology (NIST) Special Publication 800–94, 2007
- (Shieh and Gligor 1991) Shiva, S. W., Gligor, V. D.: A Pattern Oriented Intrusion Model and its Applications. In: Proceed. IEEE Computer Society Symposium on Research in Security and Privacy, pp. 327–342, 1991
- (Shimeall and Spring 2014) Shimeall, T., Spring, J.: Introduction to Information Security: A Strategic-Based Approach. Elsevier Publ. 2014
- (Shiva et al. 2010) Shiva, S., Roy, S., Dasgupta, D.: Game Theory for Cyber Security. In: CSIIRW Conf. Proceed., ACM Press 2010
- (Simmons et al. 2014) Simmons, C., Ellis, C., Shiva, S., Dasgupta, D., Wu, Q.: AVOIDIT : A cyberattack Taxonomy. In: 9th Annual Symposium on Information Assurance (ASIA), pp. 14-1-14-, 2014
- (Smaha 1988) Smaha S. E.: Haytack: An Intrusion Detection System. In: Proceed. 4th Aerospace Computer Security Applications Conference, pp. 37–44, 1988
- (Tang and McMillin 2008) Tang, H., McMillian, B.: Security Property Violation in CPS through Timing. In: Proceed. 28th Internat. Conference on Distributed Computing Systems Workshops, pp. 519–524, 2008
- (Valasek and Miller 2014) Valasek, C., Miller, C.: A Survey of Remote Automotive Attack Surfaces. Technical White Paper, IOActive Inc., 2014
- (VTTI 2007) VTTI - Virginia Tech Transportation Institute. Intersection Collision Avoidance - Violation Task 5 Final Report, 2007. <http://www.intellidriveusa.org/documents/final-report-04-2007.pdf>
- (Wang et al. 2010) Wang, E. K., Ye, Y., Xu, X., Yiu, S. M., Hui, L. C. K., Chow, K. P.: Security Issues and Challenges for Cyber Physical Systems. IEEE/ACM Conference on Green Computing and Communications and IEEE/ACM Intern. Conference on Cyber, Physical and Social Computing, pp.733–738, IEEE Publ., 2010
- (Xiao 2006) Xiao, Y.: Security in Sensor Networks. Auerbach Publ., 2006
- (Xiao et al. 2008) Xiao, K., Ren, S., Kwiat, K.: Retrofitting Cyber-Physical Systems for Survivability through External Coordination. In: Proceed. 41st Internat. Conference on Systems Science, pp. 454–466, 2008
- (Yuzhe et al. 2013) Yuzhe, L., Ling, S., Peng, D., Quecedo, E.: Jamming Attack on Cyber-Physical Systems : A Game Theoretic Approach. In: IEEE 3rd Annual Conference on Cyber Technology in Automation, pp. 252–257, 2013
- (Zeltser 2015) Zeltser, L.: Antivirus Software uses Several Different Virus Detection Techniques. TechTarget Network, 2015
- (Zimmer et al. 2010) Zimmer, C., Bhat, B., Mueller, F., Mohan, S.: Time-Based Intrusion Detection in Cyber-Physical Systems. In: Proceed. 1st ACM/IEEE International Conference on Cyber-Physical Systems, pp. 109–118, 2010

Links

- (URL1 2016) www.dhs.gov/science-and-technology/cyber-security-division
- (URL2 2016) <https://autoalliance.org/connected-cars/cybersecurity/>
- (URL3 2016) http://www.icao.int/APAC/Documents/edocs/cns/mlat_concept.pdf
- (URL4 2016) https://www.autosar.org/fileadmin/user_upload/standards/classic/3-0/AUTOSAR_TechnicalOverview.pdf
- (URL5 2016) https://en.wikipedia.org/wiki/Secure_Neighbor_Discovery
- (URL6 2016) <https://www.genivi.org/challenges>
- (URL7 2016) <http://searchcontentmanagement.techttarget.com/definition/taxonomy>
- (URL8 2016) <https://ldra.com/automotive/>
- (URL9 2016) <http://www.openvas.org/about.html>

- (URL10 2016) <https://www.automotiveisac.com/best-practices/>
- (URL11 2016) https://en.wikipedia.org/wiki/Functional_safety
- (URL12 2016) <http://www.exida.com/Resources/Term/Automotive-Safety-Integrity-Level-ASIL>
- (URL13 2016) https://en.wikipedia.org/wiki/Failure_mode_and_effects_analysis
- (URL1 2017) <https://www.techopedia.com/definition/24841/denial-of-service-attack-dos>
- (URL1 2018) <https://www.plattform-i40.de/I40/Navigation/DE/Industrie40/Handlungsfelder/Sicherheit/sicherheit.html> (in German)



Mobile Apps for the Connected Car

7

This chapter is one of the innovations in the field of the automotive industry, apps for connected cars. Section 7.1 reports on the current situation of the global automotive and IT market and its management and systems. Section 7.2 describes the current trend of agile software development in the automotive industry. Agile software development places a strong focus on fast response to customer requirements, turning speed and agility into a key competitive advantage. In Sect. 7.3 the importance of smartphones and the corresponding app market is introduced as well as their unprecedented growth in numbers. Section 7.4 focuses on the iOS operating system which offers a rich set of features and a powerful application programming interface (API) for mobile app development and is fully integrated with Apple's iCloud ecosystem. Section 7.5 explains the background of Xcode, an integrated development environment (IDE) for developers which allows multi-language and multi-target platform development for native macOS, iOS, tvOS, and watchOS applications. The supported languages are Apple's Objective-C and Swift with the Cocoa framework, but it is also possible to develop C and C++ applications. Section 7.6 introduces Android applications which are usually developed in the Java language using the Android Software Development Kit. Android is a powerful operating system competing neck-to-neck with Apple's iOS. Section 7.7 focuses on the topic how car manufacturers are embracing the smartphone technologies by integrating Apple's and Google's hardware and software into the car's infotainment system. Apple's technology is called CarPlay, Google's Android Car. In Sect. 7.8 the required programming languages for mobile app development such as Objective-C, Swift, and Java App Development are introduced. Section 7.9 introduces the requirements of the use case example of the car ride-sharing models carpooling and cab sharing. Wrapping up, in Sect. 7.10 the source code of some of the key classes in the several applications is discussed. Finally, in Sect. 7.11 exercises are included in a form of a questionnaire to be answered. The last section is followed by references and suggestions for further reading.

7.1 Automotive IT

Automotive IT deals with IT systems that are being deployed in an automotive company. The automotive industry has one of the most sophisticated IT infrastructures in the world. Typically, more than 2% of the total revenue is spent on IT. One can differentiate between the different domains where automotive IT systems are being deployed:

- Engineering
- Procurement
- Production
- Sales and Aftersales

as well as central functions like human resources (HR), finance controlling, and others.

Central IT systems for administrative functions are typically standardized and customized off the shelf system like SAP, Peoplesoft, and others (Weber 2012; Laudon et al. 2010). Engineering and production uses a mixture of commercial off the shelf systems (COTSs) and highly customized versions that are often very different from the core systems (Ludewig and Licher 2013). This has been discussed in more detail in Chap. 3 introducing the engineering IT systems. Daimler, for example, deploys a Siemens product life cycle management (PLM) solution called Teamcenter that over the year was customized and adapted in various ways (see Fig. 7.1). This Daimler-specific version is called “Smaragd.” On top of this many

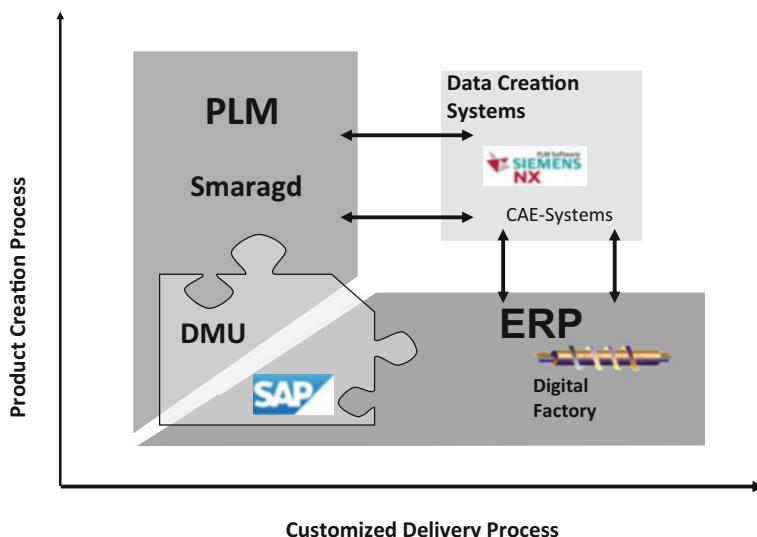


Fig. 7.1 IT systems in an automotive company

special solutions were developed like the role-based engineering workplace and the common engineering client (CEC) framework.

The typical IT landscape in an automotive company is a very specific mixture of large-scale standard software packages for central functions, grown and customized versions of standard packages, and a multitude of individual applications that were developed for special purposes. The management of this heterogeneous IT landscape is a real challenge. Figure 7.2 illustrates the service model of Daimler's IT organization. At the center of this model lies the demand management, service delivery and operations processes and the interface to the business units who are treated as customers of the internal IT organization. These core service activities are embedded into several supporting activities and processes like enterprise architecture, IT standards, skill management, financial/controlling management, and so forth.

The main objectives are:

- Cost savings
- Performance and scalability
- Stability and reliability

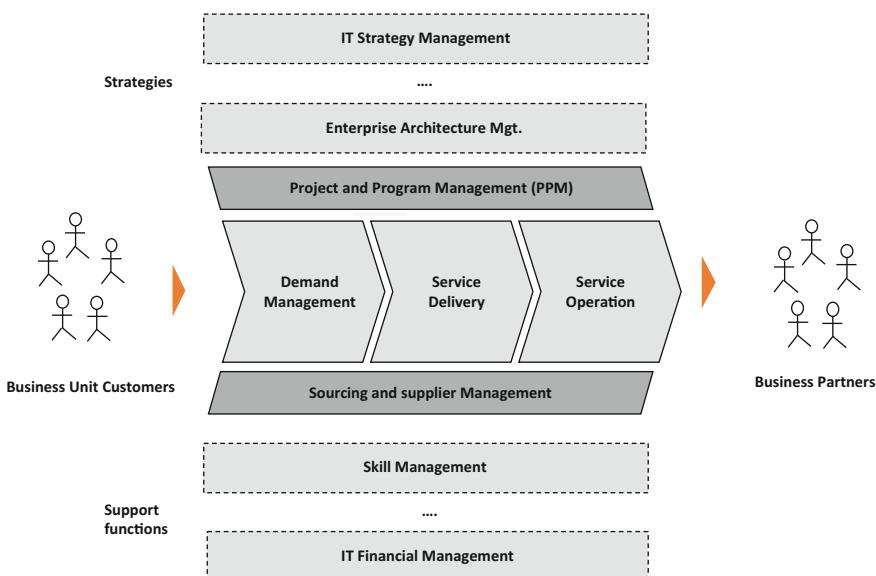


Fig. 7.2 Service model in IT management

7.1.1 IT Management and Systems in the Automotive Industry

IT management has always been a compromise between the foregoing mentioned different goals. Figure 7.2 shows the service model of the Daimler IT organization. Business units bring in their requirements; different key process improvements are defined and managed by means of a balanced scorecard (Kaplan and Norton 1996). One will find similar operating models in other automotive OEMs. Major innovations in the last years were driven by the introduction of shared service centers which served multiple divisions simultaneously and more efficiently as well as the the adoption of offshore development (Mangiapane and Büchler 2015). While offshore development in the early days was primarily driven by cost considerations (e.g., the per hour cost of a development team in India is significantly less than in Germany or the USA), quick ramp up and the availability of talent were major drivers in later stages.

The year 2000 problem, where old COBOL code had to be updated, was a big boost for the offshore industry (Grechenig et al. 2010).

Over the years, a very heterogeneous IT landscape emerged that was not only difficult to maintain but also needed huge efforts to run and keep the knowledge in-house. The problem can be easily seen if one looks at the numbers of different applications deployed in a company. Volkswagen had more than 10,000 different applications from the very large backbone systems like SAP to tiny applications running in only one department with a very specific scope. Daimler counted more than 6000. Each of these different applications needs a specific IT infrastructure such as database systems, operating system, middleware, integrated development environment (IDE) tools, and others.

Stable operations are only possible if all interfaces and dependences are managed properly. Some of the applications might have not been used at all but still had to be managed. The companies launched programs to reduce the number of applications and standardize as much as possible. Daimler developed a classification scheme that became the basis for the company's "journey to excellence (J2E)" initiative. All applications were categorized and classified according to their user base, business impact, cost, maintenance, dependencies with other applications, and future readiness (Mangiapane and Büchler 2015). If applications did not provide much business impact, but on the other hand were difficult and costly to maintain, they were removed from the portfolio, replaced, or completely shut down. That way it was possible to reduce the number of applications significantly without compromising the general functionality. Development projects could be focused and company-wide standards established.

The core backbone systems today, as shown in Fig 7.1, include:

- SAP as the central platform for enterprise resource planning (ERP) like finance, controlling, procurement and production control
- Teamcenter-Smaragd for PLM
- Different CAD authoring tools like NX or CATIA
- IBM DOORS for requirements engineering
- Microsoft products for unified messaging, e-mail, operating system, workflow management, etc.

Also, the development and operation environments were standardized using either the Java platform with the Eclipse IDE or C# and .NET with the Visual Studio tool suite. The middleware is standardized around so-called technology stacks (Weber 2012; Schäfer 2010; Masak 2010).

The proactive infrastructure (PAI) stack is the backbone for multi-tier Java Enterprise applications in Daimler. This technology stack evolved over more than 15 years. It is based on IBM's WebSphere Application Server, includes authentications and security frameworks, and also standardized the persistence layer with interfaces to IBM's DB2 database, Oracle's database server, or Microsoft's SQL Server.

All new Java development for desktops and enterprise applications has to use this target infrastructure by default. Exceptions are normally not allowed; one has to find really good arguments for not using the standard stack.

These initiatives have led to a stable, reliable, and predictable IT infrastructure, comprising a few thousand applications that serve hundreds of thousands of employees worldwide and connect sales divisions, workshops, and thousands of business partners and suppliers worldwide. The core goals are cost efficiency, performance, smooth operations, predictable releases, and easy maintenance.

The smartphone and social media boom have also changed the automotive industry fundamentally. Typical characteristics are:

- Easy distribution of new software versions over the air (app stores).
- Easy error fix with frequent updates in the field.
- Everybody can participate and consume or produce content.
- High expectations of users regarding user interface functionality and innovations.
- Very fast response times.

The smartphone economy has had tremendous impact on nearly all business sectors. Smartphones were one of the most successful products of all time (Laudon et al. 2010). Apple's iPhone, for example, has generated more revenue and more profit than any other single product in mankind's history.

The pie chart in Fig. 7.8 shows the market share in 2016 of the leading platforms, iOS and Android. Figure 7.9 illustrates the share of smartphone manufacturers. The competition is so intense that even IT heavyweights like Microsoft, who used to dominate the desktop business, had to give in to Google and Apple on the mobile market (Dörner 2016).

The automotive industry, of course, was not used to such short release cycles, quick-and-dirty releases, and extreme software innovation cycles. This has led to the term “two-speed IT”:

- Fast-paced, possibly less reliable, go-to-market first apps that have to be developed and deployed as soon as possible
- Solid, well-tested, and reliable backbone systems that were created in the classical way

The possibilities of Car IT, connected services, infotainment, and apps have boosted this business. All premium manufacturers have reacted to this challenge and are investing into apps, company-specific app stores, and exciting new functionalities like:

- Connected parking
- Integration of car and smart home
- New mobility services
- Remote control of central functions
- Remote diagnostics

Automotive OEMs see a huge potential in agile development processes to meet these challenges. Therefore, the next section discusses this approach and gives a brief overview of one of the most widely used agile development processes—Scrum.

7.2 Agile Software Development

The rise of agile software development is best understood if one looks at the problems of large-scale software development (Ludewig and Licher [2013](#); Grechenig et al. [2010](#); Sommerville [2015](#)):

- Changes are difficult to handle in later phases.
- Huge projects failed because the requirement phase was overengineered and the communication between analysts, stakeholders, domain experts, coders, and operating specialists did not work properly.
- Linear phases like in the classical waterfall model are too rigid and inflexible.

In February 2001, the “Agile Manifesto” focused on a set of core principles that one can summarize in the following way (Sommerville [2015](#); Grechenig et al. [2010](#); iX [2017](#)):

- Changes are welcome and part of the business.
- Focus on the product and not on the documentation.
- Frequent releases of workable software.
- Communication between all stakeholders is preferred over rigid processes.

Scrum is by far the most popular agile development process (iX [2017](#); Sommerville [2015](#)). A Scrum project has two key roles:

- The Scrum master
- The product owner

The Scrum master orchestrates the whole process. Daily Scrum meetings ensure that everybody is informed about the status and the objectives of the project.

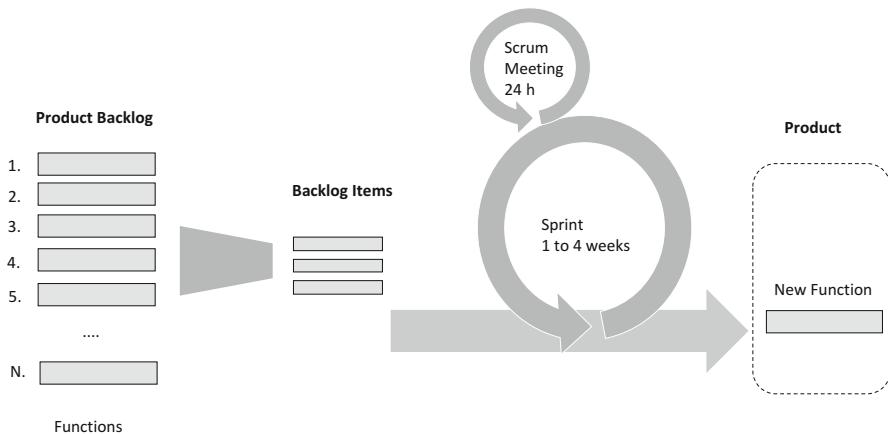


Fig. 7.3 Agile software development—example Scrum (Grechenig et al. 2010)

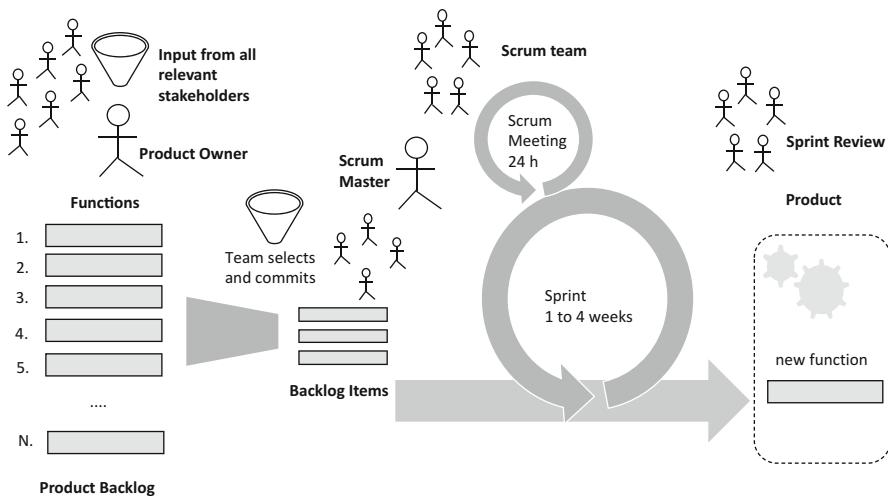


Fig. 7.4 Different roles in a Scrum project (Sommerville 2015; iX 2017)

The product owner is a key interface to the stakeholders. He or she will make sure that the main use cases get into the product backlog. Figures 7.3 and 7.4 summarize the Scrum development process.

The effort estimation uses a couple of interesting techniques that encourage communication and create a common understanding of the complexity and risks involved. Although, counting lines of code is not adequate to measure the complexity of software this simple metric gives an indication of the rising content of software in IT and embedded systems as shown in Fig. 7.5 (URL18 2017). Architectural, algorithmic, behavioral (e.g., real-time), nonfunctional (e.g., scalability), security,

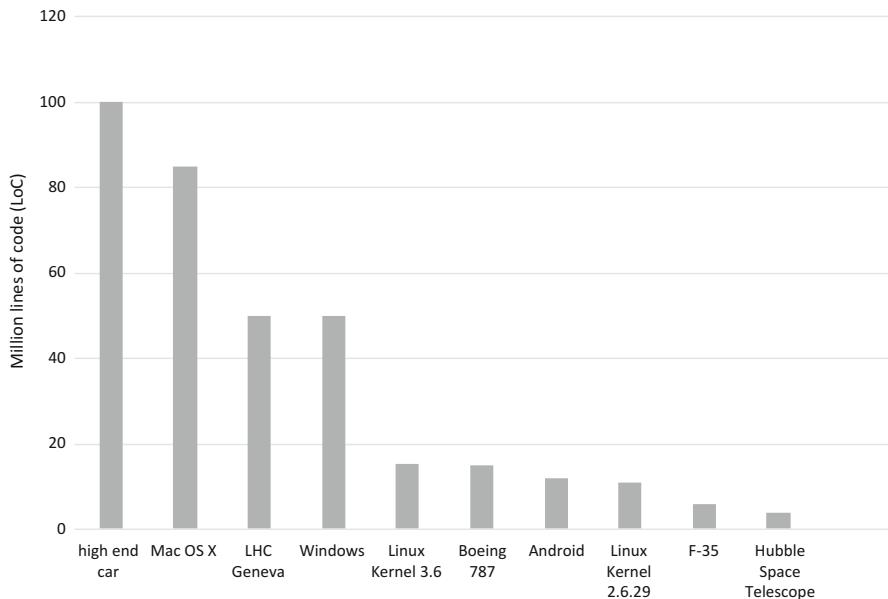


Fig. 7.5 Rising software content measured in Lines of Code (LoC) in different IT and embedded systems (URL2 2014; URL18 2017)

functional safety, life cycle, and operating environmental aspects become more and more important; the following papers discuss these topics more in detail (Grechenig et al. 2010; Ludewig and Licher 2013; Hoffmann 2013).

In agile software development, the effort estimation often relies only on a partial and incomplete understanding of the requirements. Still, a lot can be done to understand and grasp the complexity of the software product. Again, the team plays an important role in developing a common understanding of the inherent complexity.

Popular methods of agile effort and complexity estimation like:

- Bucket Estimation
- Planning Poker

can be found in a recent special issue of the iX magazine (iX 2017). Sommerville (2015) gives a detailed overview of agile software development.

A key questions that often arises in agile development are how to plan the releases, how to estimate efforts and costs, and how to manage the suppliers if the software development is partially or completely outsourced as shown in Fig. 7.6.

A good understanding of key nonfunctional requirements is important.

There is no black and white approach and often models are tailored. Experience plays a big role. A good architecture which can incorporate change and a thorough understanding of key nonfunctional requirements are important.

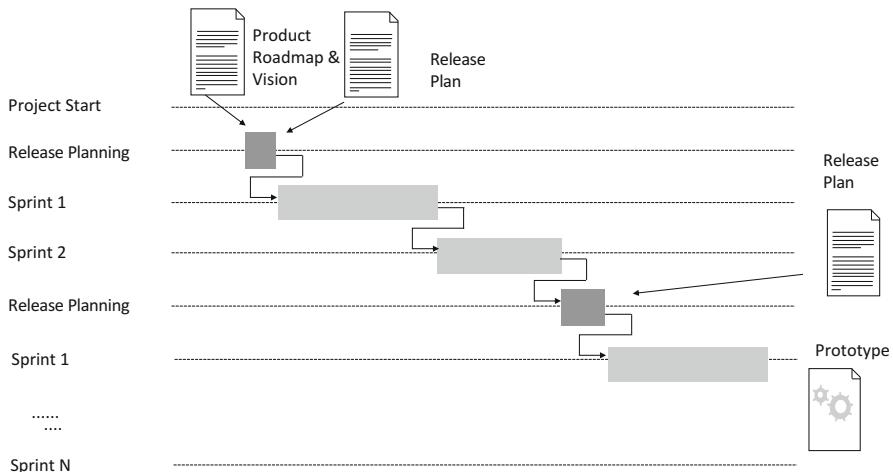


Fig. 7.6 Release planning in Scrum (iX 13/2017)

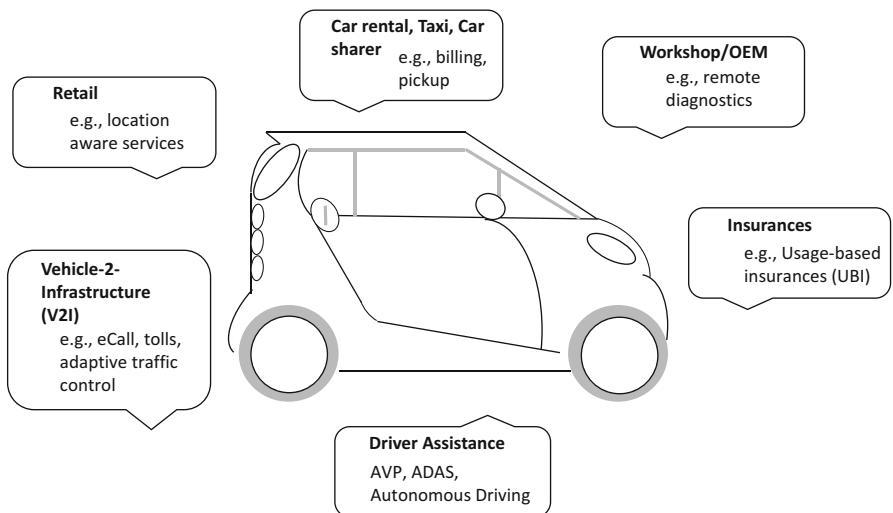


Fig. 7.7 The connected car—use cases

7.2.1 Challenges and Two-Speed IT

As the cycles for app development are much shorter than for standard automotive IT systems or embedded software in cars, the term “two-speed IT” came up. Automotive OEMs hope to catch up with the fast pace of app development while not compromising reliability, safety, and stability of the IT landscape (Herchet et al. 2015). This is not trivial to achieve and needs a thorough planning. Therefore, agile methodologies play a key role here (Ludewig and Licher 2013; Müller et al. 2015; Hülshorst et al. 2015), as illustrated in Fig. 7.7 for the connected car use case.

7.3 The Smartphone and App Market

The worldwide smartphone market in 2016 accounted for over a billion phones shipped. Several trends are emerging:

- Chinese manufacturers like Huawei attack in western markets with inexpensive derivatives of their flagship smartphones like the Huawei P10 Mate.
- New low-cost brands enter the market globally.
- Samsung is still very strong but had a setback with the Samsung Galaxy 7 battery problems. The company had to discontinue the product after an unsuccessful call back where the battery problem could not be fixed. Airlines warned passengers and ultimately did not allow the Samsung Galaxy phone onboard anymore ([Hecking 2016](#)).

New budget brands are growing rapidly in India and China. Android dominates the market with more than 80% share as shown in Fig. 7.8. Samsung reasserted its global leadership with the success of its new flagship devices but the recall and brand damage costed many billions of dollars ([Hecking 2016](#)).

Android share has risen, with strong growth in unit shipments by other players such as Huawei, OPPO, and others.

The market share of iOS has declined by more than 20% in the second quarter of 2016. With the iPhone 7 that was launched in September 2016, IDC expected strong

Fig. 7.8 Smartphone operating system distribution ([URL1 2016](#))

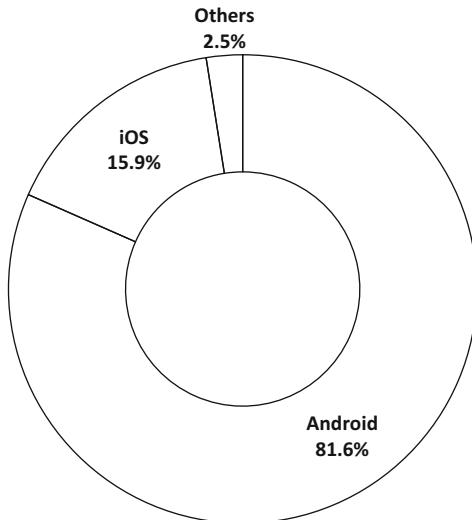
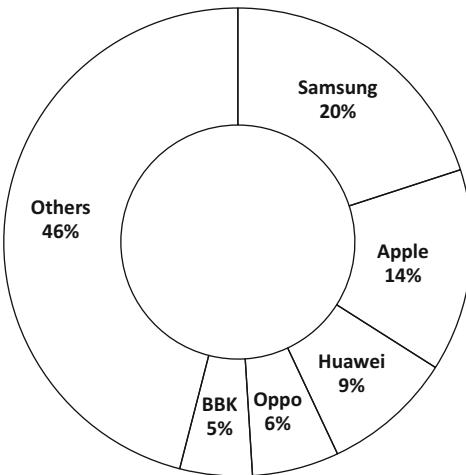


Fig. 7.9 Major smartphone manufacturers (market share by vendors in 2016 (URL4 2017))



sales in Q4 2016 (URL4 2017). The 2017 version of the iPhone introduced several new technologies as it marked the 10th anniversary of the launch of the first iPhone. However, the retail price was the highest so far.

WindowsPhone experienced a major setback. With Microsoft's focus on business users, the decline in the consumer market is expected to continue (Dörner 2016). The decline of Nokia, the leader in mobile phones and smartphone, just 10 years ago, has shown how competitive this market is (Busse 2016). Figure 7.9 shows the market share of major smartphone manufacturers in 2016.

7.4 iOS

7.4.1 The History of iOS

iOS is an adaptation of macOS for Apple mobile devices, like the iPhone and the iPad (URL1 2017, URL3 2017). It is not available for any non-Apple devices. This was Apple's mantra since the founding days and was re-inforced by Steve Job's return to Apple in 1996 (Lashinsky 2012). This is in contrast to other OEMs who concentrated on the hardware and licensed the operating system from Microsoft. For Apple it was always crucial to control the look and feel of the platform. Today there is another derivative for the iWatch which is called watchOS. iOS was developed under the lead of Scott Forstall (Lashinsky 2012). Apple's macOS operating system, the current version of it is macOS Mojave, has an interesting history (Linzmayer

2004; Singh 2007). The core functionality is based on NeXTSTEP, the operating system of NeXT Computers. NeXT was acquired by Apple in an attempt to develop a modern operating system as their own internal project called Copland was not able to compete with the graphic user interface (GUI) and performance of Microsoft Windows anymore (Linzmayer 2004). NeXTSTEP is a UNIX system that traces its roots back to the micro-kernel-based Mach operating system (Tanenbaum and Austin 2012). On a side note: Mach was also the basis for the popular Amiga computer that offered many modern OS features, a rich GUI, and powerful multimedia feature right back in middle of the 1980s.

Major versions of iOS are released annually. The current version, iOS 12, was released in June 2018. It runs on iPhone 5 and later, iPad 4th generation and later, iPad Pro, iPad Mini 2 and later, and the 6th-generation iPod Touch.

7.4.2 The iOS Platform

The iOS platform offers a rich set of features and a powerful application programming interface (API) for mobile app development and is fully integrated with Apple's iCloud ecosystem:

Typically, the iPhone performs very well in benchmarks. This is due to a powerful custom processor called AX, which is based on a multicore ARM architecture combined with a fast, embedded GPU. The current version, A12 Bionic, is the core processor of the iPhone XS and XR. Apple designed its own GPU and included a special processor for machine learning. Apple's AX processors are very fast, even outperforming classical desktop CPUs. Here, the seamless integration of the iOS software and custom hardware pays off. Drivers, core functions, and the platform are optimized for the few variants of Apple's own product family, while Android devices have to deal with a multitude of different hardware configurations.

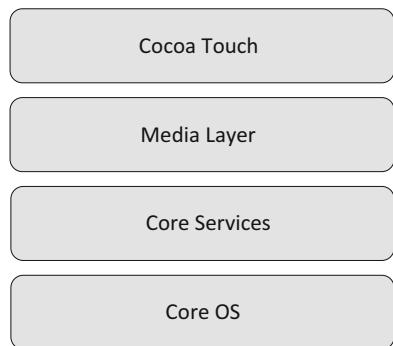
There are several core platform APIs that can be used for app development:

- Foundation Kit Framework
- GameKit framework for games
- iAd Framework
- Map Kit Framework for map-based applications
- UIKit Framework (based on Application Kit)

7.4.3 The iOS Architecture

At the highest level, iOS acts as an intermediary between the underlying hardware and the apps (URL1 2017). Apps do not talk to the underlying hardware directly. Instead, they communicate with the hardware through a set of well-defined system interfaces. These interfaces make it easy to write apps that work consistently on devices having different hardware capabilities (URL1 2017).

Fig. 7.10 Layered architecture of iOS



The iOS programming model is based on a set of layers as shown in Fig. 7.10. Lower layers contain fundamental services and technologies. Higher-level layers build upon the lower layers and provide more sophisticated services and technologies (URL2 2017). This is a fundamental method in computer science for complexity reduction and abstraction (Tanenbaum and Austin 2012; Ludewig and Licherter 2013; Stokes 2007; Schäfer 2010).

The layers comprise a set of functions, services, components and frameworks as shown in Fig. 7.11 (URL8 2017; Stevenson 2010):

- *Cocoa Touch Layer*: Cocoa is the application framework of macOS. Cocoa Touch is the corresponding layer in iOS. It contains key frameworks for building iOS apps. These frameworks define the appearance of the app (URL8 2017). They also provide the basic app infrastructure and support for key technologies such as multitasking, touch-based input, push notifications, and many high-level system services.
- *Media Layer*: The Media layer contains the graphics, audio, and video technologies that one can use to implement multimedia features.
- *Core Services Layer*: The Core Services layer contains fundamental system services. Key among these services are the core foundation and foundation frameworks, which define the basic types that all apps use. This layer also contains individual technologies to support features such as location, iCloud, social media, and networking.
- *Core OS Layer*: Contains the low-level features that most other technologies are built upon. Typically, an application developer is not directly confronted with this layer but the services are used implicitly by higher layers. However, in case one needs to communicate with an external hardware accessory, there is no way around to directly use the frameworks in this layer.
- *Core Graphics*: This is a low-level, C-based framework (URL8 2017). The layer handles high-quality vector graphics, path-based drawing, transformations, images, data management, and more. The easiest and most efficient way to create graphics in iOS is to use pre-rendered images with the standard views and

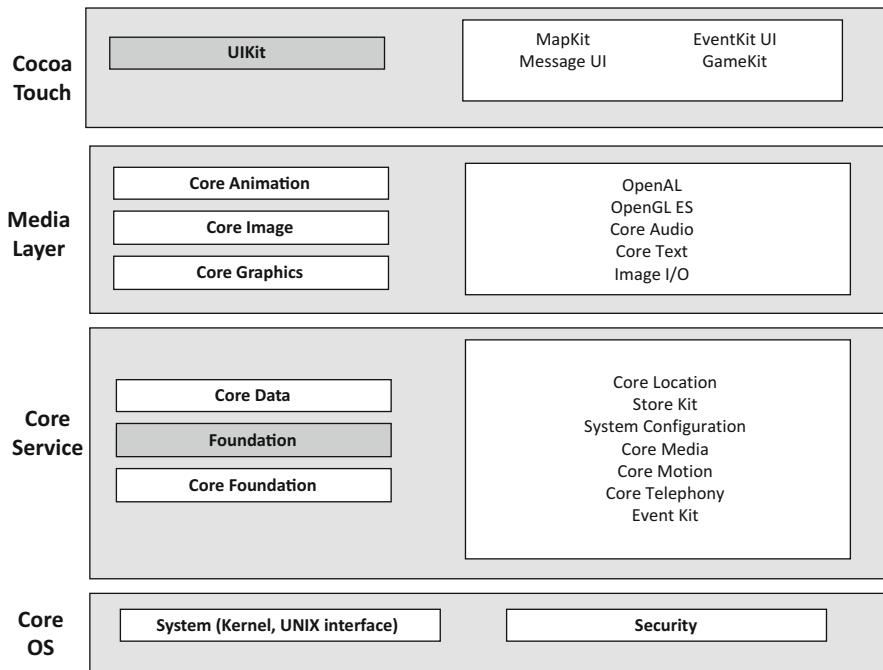


Fig. 7.11 iOS APIs ([URL12 2017](#))

controls of the **UIKit** framework, the higher-level framework which also provides classes for custom drawing including paths, colors, patterns, gradients, images, text, and transformations.

- **Core Animation:** Its interfaces are part of the Quartz Core framework and provide the functionality for higher-level animations and visual effects. **UIKit** provides animations that are built on top of the core animation technology. If one requires advanced animations that go beyond the capabilities of **UIKit**, one can use core animation directly. One can create a hierarchy of layer objects that can be manipulated, rotated, scaled, and transformed. There is no need for using low-level graphics APIs such as **OpenGL ES**.
- **Games:** iOS's **GameKit** framework supports the development of games. It contains the following kits ([URL7 2017](#); [URL8 2017](#)):
- **Sprite Kit:** This framework provides graphics support for animating arbitrary textured images or sprites. It also contains support for simulating of physical laws and is a good option for games and other apps that need complex animation chains.
- **OpenGL ES:** A low-level framework that supports hardware-accelerated 2D and 3D drawing giving high frame rates for full screen, game-style apps.
- **Game Controller:** This framework makes it simple to search controllers connected to a Mac or iOS device. Apple has designed specifications for

hardware controllers to ensure that all controllers have consistent sets of control elements that both players and game designers can rely on.

- *Data:* The Core Data framework manages an app's data model. It uses the built-in SQLite technology to save and manage data.

7.5 Xcode

Apple provides a powerful IDE for developers free of cost. It is called Xcode and is bundled with the Mac computer family (URL6 2016; URL8 2017). It allows multi-language and multi-target platform development for native macOS, iOS, tvOS, and watchOS applications. The supported languages are Apple's Objective-C and Swift with the Cocoa framework, but it is also possible to develop C and C++ applications (Fig. 7.12).

The Xcode editor shown in Fig. 7.12 offers all features of a professional editor with advanced code completion, code folding, syntax highlighting, and context-sensitive information within the code (Balzert 2009; Ludewig and Licher 2013). Apart from the editor, the Xcode programming environment provides a full set of tools that are needed for modern software development (URL6 2017):

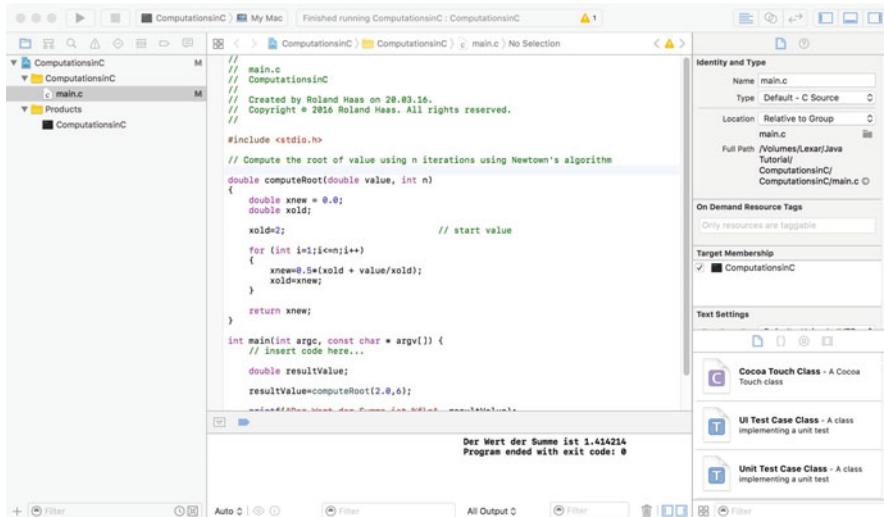


Fig. 7.12 The Xcode IDE

- *Editors:* The editor pane can be split in two. It can show the header counterpart, the superclass, callers, and callees. The version editor displays a running timeline of commits, helps to determine issues, and graphically goes back in time to compare source files. Subversion and Git source control are fully supported.
- *Interface Builder (IB):* The IB allows to design and test the user interface without writing a line of code, prototype in minutes, then graphically connecting the interface to the source within the Xcode editor.
- *iOS Simulator:* With the help of the simulator, one can build, install, run, and debug Cocoa Touch apps in a Mac-based iOS environment before deployment on actual physical devices.
- *Integrated Build System:* The build system can handle complex builds, scaling to maximize the power of multicore Macs, and will automatically sign, provision, and install iPad and iPhone apps onto a device.
- *Compilers:* Xcode integrates open source compilers for C, C++, and Objective-C which are optimized by Apple to produce fast apps specifically tuned for the CPUs in iPhone, iPad, and Mac.
- *Graphical Debugger:* Allows to debug the app directly within the Xcode editor.
- *Continuous Integration:* Xcode Server, a feature of OS X Server, controls server-side bots that continuously build, analyze, test, and even archive Xcode projects. The Xcode IDE configures these bots, analyzes nightly build and test results, and can track down which check-in caused an error.
- *Other features:* Shortened API documentation is displayed while programming, including comment suggestions that one can write for the code. A brief overview is presented during code completion, with more links and references available within the utility area.
 - Unit test can be generated with the XCTest API.
 - Static analysis finds bugs in the code before the app is even run by letting the built-in static analyzer try out thousands of possible code paths in a few seconds. One gets a report of potential bugs that could have remained hidden or are nearly impossible to replicate.
 - Data recording is possible to analyze what type of data to collect, and simply click the big red button as data is collected and stored for further analysis.
- *Visual Comparison:* As data is recorded and displayed over time, it is easy to see relationships, both between different types of collected data and the same data collected over multiple runs.
- *Drill Down:* Allows inspecting data spikes on the graph to see what code is executing at the time and then easily jumps back into Xcode to fix the problem.
- *Instrument Library:* One can choose any of the bundled instruments in the library from low-level CPU, network, or file activity, to advanced graphics and user-event instruments.
- *Zombie Detection:* Hard-to-find errors and crashes can be trapped within instruments when an app tries to access memory that is no longer available.
- *Source View:* Drills down through data points, sorts to find the most CPU-consuming methods, and displays the code directly within the instruments UI to pinpoint the problem.
- *System Trace:* Taking up very few resources, instruments records information about all the processes on your system, revealing performance bottlenecks caused as processes interact.

7.6 Android

Android powers hundreds of millions of mobile devices around the world. It's the largest installed base of any mobile platform and growing fast. Every day more than 1 million new Android devices are activated worldwide. Android applications are usually developed in Java using the Android Software Development Kit. Once developed, apps can be packaged easily and distributed through an app store such as Google Play, SlideME, Opera Mobile Store, Mobango, F-droid, or the Amazon Appstore. Android is a powerful operating system competing neck-to-neck with Apple's iOS. The core features are ([URL5 2017](#); [URL9 2017](#); [URL10 2017](#)):

- *Powerful User Interface*: Android OS basic screen provides a powerful and intuitive user interface.
- *Connectivity*: GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, NFC, and WiMAX.
- *Storage*: SQLite, a lightweight relational database, used for data storage purposes.
- *Media support*: H.263, H.264, MPEG-4 SP, AMR, AMR-WB, AAC, HE-AAC, AAC 5.1, MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF, and BMP.
- *Messaging*: SMS and MMS.
- *Web Browser*: Based on the open source WebKit layout engine, coupled with Chrome's V8 JavaScript engine supporting HTML5 and CSS3.
- *Multi-touch*: Android has native support for multi-touch which was initially made available in handsets such as the HTC Hero.
- *Multi-tasking*: One can switch from one task to another, and at the same time various applications can run simultaneously.
- *Resizable Widgets*: Widgets are resizable, so users can expand them to show more content or shrink them to save space.
- *Multi-language*: Supports single direction and bidirectional text.

Until 2016, Google also supported a plug-in for Eclipse that allowed the development of Android applications in the well-known and popular Eclipse environment. This was discontinued and replaced by Google's own development environment, Android Studio, which is available on all leading platforms.

The principle of developing and packaging Android applications is similar to iOS. The developer can use a powerful technology stack shown in Fig. 7.13 and a rich set of APIs which abstract from the underlying smartphone hardware. The design of the user interface is done visually using a GUI builder which is part of the IDE, in this case Android Studio. All configurations are stored as XML files. The IDE also contains powerful mechanisms for testing, simulation, and deployment.

The programming language of choice for Android is Java, with a special set of APIs and frameworks that are appropriate for the smartphone embedded platforms and its hardware constraints and features ([Meier 2012](#)).

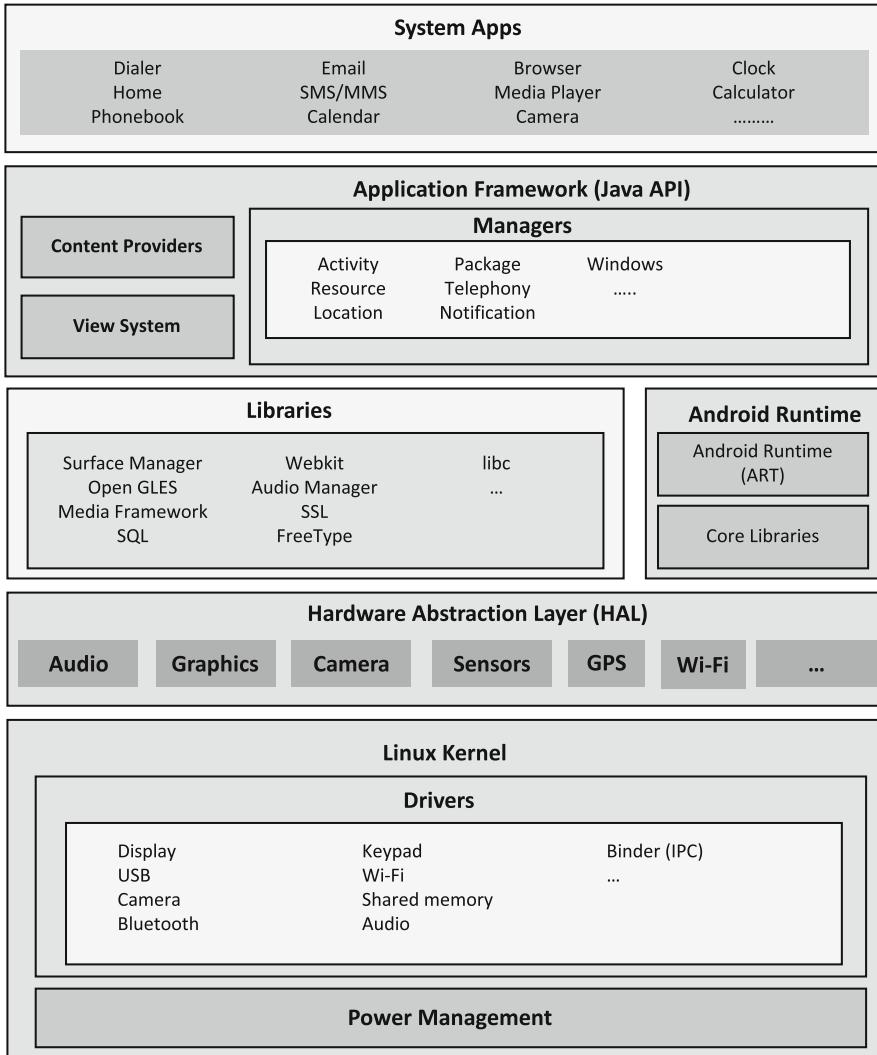


Fig. 7.13 The Android technology stack (URL11 2017)

7.7 iOS and Android in the Car

Car manufacturers are embracing the smartphone technologies by integrating Apple's and Google's hardware and software into the vehicle's infotainment system. Apple's technology is called "CarPlay," Google's "Android Car."

Both systems are similar and use the infotainment screen as a display while leaving the central control with the smartphone. The integration with the car's infotainment system is achieved by mirroring the smartphone's screen to the infotainment screen. The connection between smartphone and the car's infotainment system is established by Bluetooth. As the control remains with the user's smartphone, the user's inputs on the infotainment dashboard are sent back to the phone and processed on the phone, and the screen output is then synchronized with the car screen (Fig. 7.14).

Developers can choose from several approaches to integrate Android in a vehicle. Some vehicle manufacturers use Android as the core operating system for the infotainment system (Fig. 7.15).

Fig. 7.14 Apple CarPlay in a Seat car



Fig. 7.15 iOS Maps on CarPlay in a Seat car



Others use Android as a kind of guest operating system in a “container”. Virtualization technologies like Linux Container (LXC), enable the allocation of resources from the Linux host to the Android guest. This includes memory for apps, access rights, services, and interaction with other domains. The container should create a safe environment, so that the user can download only reliable apps.

Another technique to integrate Android in an infotainment and vehicle information (IVI)-system is to use a hardware or software virtualization layer. In this scenario, each operating system or domain runs on an assigned virtual device.

Communication can take place in a controlled manner between the domains. Boot-ups may be independent; this allows that safety-critical functions are available more quickly than the infotainment system or Android itself.

7.8 Objective-C, Swift, and Java App Development

7.8.1 Objective-C

Objective-C is a general-purpose, object-oriented programming language that adds Smalltalk-style messaging to the C programming language (Sadun and Wardwell 2014). It was the main programming language used by Apple for the OS X and iOS operating systems, and their respective application programming interfaces (APIs) Cocoa and Cocoa Touch prior to the introduction of Swift.

The programming language Objective-C was originally developed in the early 1980s. It was selected as the main language by NeXT for its NeXTSTEP operating system, from which OS X and iOS are derived.

The following code is an example of Objective-C implementing a class Person that holds the name of a person as a String and the age as an integer.

```
@interface Person : NSObject {
    @public
    NSString *name;
    @private
    int age;
}
@property(copy) NSString *name;
@property(readonly) int age;
-(id)initWithAge:(int)age;
@end
```

The next code snippet initializes the class and prints out the value on the console in two ways, first using the Smalltalk like message passing mechanism, and then based on the more common Java- and C#-like dot notation:

```
Person *aPerson = [[Person alloc] initWithAge: 53];

// NOTE: dot notation, uses synthesized setter,
// equivalent to [aPerson setName: @"Steve"];

aPerson.name = @"Steve";

NSLog(@"%@", "Access by message (%@), dot notation(%@),
property name(@) and direct instance variable access (@@)",
[aPerson name], aPerson.name, [aPerson valueForKey:@"name"],
aPerson->name);
```

The following code example is the implementation of a simple calculator model in Objective-C using Xcode and the Interface Builder (IB) as shown in Fig. 7.16.

We explain the implementation of the buttons 7, 8, 9, and 4. The buttons 6, 5, 3, 2, 1, and 0 are implemented the same way but not shown here in detail. The code also contains simple test functions to print out the keys on the console when pressed.

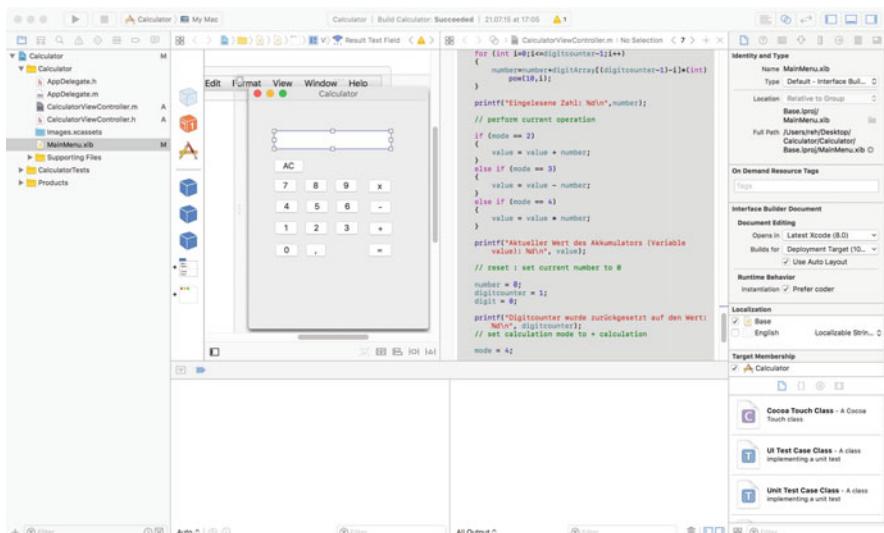


Fig. 7.16 Implementation of a simple calculator app in Objective-C with the help of Xcode's Interface Builder

```
//  
//  CalculatorViewController.m  
//  Calculator  
//  
//  Created by Roland Erik on 21. Nov.16.  
//  Copyright (c) 2016 Erik. All rights reserved.  
//  
  
#import "CalculatorViewController.h"  
@implementation CalculatorViewController : NSObject  
- (IBAction)buttonSevenPressed:(id)sender {  
    // Store digits in Array  
    digit=7;  
    digitArray[digitcounter-1]=7;  
    printf("7 was pressed.\n");  
    digitcounter++;  
    [numberString appendString: @"7"];  
    [_resultTextField setStringValue: numberString];  
}  
- (IBAction)buttonEightPressed:(id)sender {  
    // Store digits in Array  
    digit=8;  
    digitArray[digitcounter-1]=8;  
    printf("8 was pressed.\n");  
    digitcounter++;  
    [numberString appendString: @"8"];  
    [_resultTextField setStringValue: numberString];  
}  
- (IBAction)buttonNinePressed:(id)sender {  
    // Store digits in Array  
    digit=9;  
    digitArray[digitcounter-1]=9;  
    printf("9 was pressed.\n");  
    digitcounter++;  
    [numberString appendString: @"9"];  
    [_resultTextField setStringValue: numberString];  
}
```

The next code segment shows the implementation of simple arithmetic operators and the computation of the final result. The calculation is done keeping track of the computational status with the help of an automaton that switches states accordingly.

```
- (IBAction)buttonEqualsPressed:(id)sender {
    NSLog(@"The result is: %d", value);
    [_resultTextField setIntValue: value];
    mode = 2;
    value = 0;
    number = 0;
    digitcounter = 1;
    digit = 0;
}
- (IBAction)buttonPlusPressed:(id)sender {
    // Number generation
    printf("Current Status of Digitcounter: %d\n", digitcounter);
    number=0;
    digitcounter--; // Counter is by 1 to high
    for (int i=0;i<=digitcounter-1;i++)
    {
        number=number+digitArray[(digitcounter-1)-i]*(int)pow(10,i);
    }
    printf("Read Number: %d\n", number);
    // perform current operation
    if (mode == 2)
    {
        value = value + number;
    }
    else if (mode == 3)
    {
        value = value - number;
    }
    else if (mode == 4)
    {
        value = value * number;
    }
    printf("Current Value of Accumulator: %d\n", value);
    // reset : set current number to 0
    number = 0;
    digitcounter = 1;
    digit = 0;
    printf("Reset Digitcounter on Value: %d\n", digitcounter);
    // set calculation mode to + calculation
    mode = 2;
    [numberString setString: @" "];
}
- (IBAction)buttonMinusPressed:(id)sender {
    // Generate Number
    printf("Current Status of Digitcounter: %d\n",
          digitcounter);
```

```
number=0;
digitcounter--; // Counter is by 1 to high
for (int i=0;i<=digitcounter-1;i++)
{
    number=number+digitArray[(digitcounter-1)-i]*(int)pow(10,i);
}
printf("Read Number: %d\n",number);
// perform current operation
if (mode == 2)
{
    value = value + number;
}
else if (mode == 3)
{
    value = value - number;
}
else if (mode == 4)
{
    value = value * number;
}
printf("Current Value of Accumulator: %d\n", value);
// reset : set current number to 0
number = 0;
digitcounter = 1;
digit = 0;
printf("Reset Digitcounter on Value: %d\n", digitcounter);
// set calculation mode to + calculation
mode = 3;
[numberString setString: @" "];
}
- (IBAction)buttonMultiplyPressed:(id)sender {
    // Generate Number
    printf("Current Status of Digitcounter: %d\n",
          digitcounter);
    number=0;
    digitcounter--; // Counter is by 1 to high
    for (int i=0;i<=digitcounter-1;i++)
    {
        number=number+digitArray[(digitcounter-1)-i]*(int)pow(10,i);
    }
    printf("Read Number: %d\n",number);

    // perform current operation
    if (mode == 2)
```

```
    {
        value = value + number;
    }
    else if (mode == 3)
    {
        value = value - number;
    }
    else if (mode == 4)
    {
        value = value * number;
    }
    printf("Current Value of Accumulator: %d\n", value);
// reset : set current number to 0
number = 0;
digitcounter = 1;
digit = 0;
printf ("Reset Digitcounter on Value: %d\n", digitcounter);
// set calculation mode to + calculation
mode = 4;
[numberString setString: @" "];
}
- (IBAction)buttonACPressed:(id)sender {
    mode = 2;
    value = 0;
    number = 0;
    digitcounter = 1;
    digit = 0;
    numberString= [NSMutableString alloc] init];
}
@end
```

7.8.2 Swift

Swift is a general-purpose, multi-paradigm, compiled programming language developed by Apple Inc. for iOS, macOS, watchOS, tvOS, and Linux (Bleske 2016; URL7 2017).

Swift was introduced at Apple's 2014 Worldwide Developers Conference (WWDC). It underwent an upgrade to version 1.2 during 2014 and a more major upgrade to Swift 2 at WWDC 2015 (URL8 2017) and is currently available in version 4. Initially, a proprietary language, version 2.2 was published as open source software and made available under Apache License 2.0 on December 3, 2015, for Apple's platforms and Linux (URL8 2017).

Swift is designed to work with Apple's Cocoa and Cocoa Touch frameworks and the large body of Objective-C code already written for Apple products. Swift is intended to be more resilient to erroneous code, safer than Objective-C, and to integrate latest results of software technology research.

It is built with the LLVM compiler framework included in Xcode, on platforms other than Linux, and uses the Objective-C runtime library, which allows C, Objective-C, C++, and Swift code to run within one program (URL7 2017).

Swift supports the core concepts that made Objective-C flexible, notably dynamic dispatch, widespread late binding, extensible programming, and similar features. These features also have well-known performance and safety trade-offs, which Swift was designed to address. For safety, Swift introduced a system that helps address common programming errors like null pointers (URL8 2017). For performance issues, Apple has invested considerable effort in aggressive optimization that can flatten out method calls and accessors to eliminate this overhead. More fundamentally, Swift has added the concept of protocol extensibility, an extensibility system that can be applied to types, structs, and classes. Apple promotes this “protocol-oriented programming” as a real change in programming paradigms (Bleske 2016).

7.8.3 Java

Java is the primary development language for Android applications. The class framework is customized and the virtual machine optimized for high performance. User interfaces are generated from an XML description of the configuration and positioning of control elements. The design of the user interface is completely visual with the built-in GUI design tool from Android Studio.

Although, Microsoft has given up on competing with Apple and Google in the market for mobile operating systems, the company’s Xamarin framework is an interesting tool for cross platform development. With a C#-shared codebase, developers can use Xamarin to write native Android, iOS, and Windows apps with native user interfaces and share code across multiple platforms, including Windows and macOS (URL15 2017).

7.9 A Ride-Sharing Example

Carpooling and cab sharing are concepts which have become very popular (Chan and Shaheen 2012; URL1 2014; Rayle et al. 2014). These ideas are eco-friendly; they help to reduce cost and minimize traffic. Carpooling is the shared use of a car among people who know each other, particularly for commuting to work. Most of them may own a car themselves but choose to share the ride with others who are heading in the same direction. In the meantime, the own car can be used otherwise, e.g., for transporting family members.

Cab sharing is another similar idea, where one shares the ride with other people, who want to reach the same or a nearby destination, the exact details depending on whether the ride is short, for example, within the city, or long, for example, across states, which is typically planned well ahead. In the first case, the co-passengers may keep joining and stepping out, throughout the ride. In the latter case, the information about the location from where each person joins and steps out of the ride should be known well ahead.

The ride-sharing app *CoRide* offers shared rides among a group of app users (Reddy et al. 2016). It was developed in 2016 by a group of students (Abhijay V., Akshay Jindal, Anubhav Bhardwaj, Lijo Johny and Sanat Ramesh) of the International Institute of Information Technology Bangalore IIIT-B as a class project together with Clausthal University of Technology (TUC) in automotive electronics and Car IT keeping in mind the needs of student transport from and to the campus in crowded cities like Bangalore. The app is currently available for Android, but in the future also an iOS version is planned. The users request a ride with a simple click. The geo-location is logged and shared, and the position is shown on the map on the mobile phone. Also, drivers using the app who are willing to offer a ride are displayed. The drivers can see the request in their area or close vicinity and are able to respond with a ride offer. This transaction is based on a first-come first-serve principle. The system works cashless. Instead of cash, drivers get *incentive points* when they take a passenger on a ride. Passengers lose points if they use a ride which means that their point account is reduced.

Both passengers and drivers must be known to the system and have to authenticate themselves by logging in with a username and a password.

7.9.1 Core Use Cases

The CoRide Android application intends to make carpooling and cab sharing easier by allowing the user to find co-passengers and to share the ride.

There are three types of users (Reddy et al. 2016):

- *Customers*: These are people who use this app to get a shared ride. They are the primary beneficiaries of this application. The CoRide app provides three options for users: *Car Pooling*, *Long Ride*, and *City Cabs* represented by buttons on the app screen. The options are *Offer a Ride* and *Need a Ride*. *Offer a Ride* and *Need a Ride* buttons allow users to register his/her details with CoRide. When the *City Cabs* button is clicked a new screen will appear where a request for a shared cab ride can be made and the fare estimate can be viewed.
- *Drivers*: The CoRide app will notify registered City Cabs drivers about incoming requests. *Accept* and *Reject* options in the form of buttons will be shown. When a driver clicks on the *Accept* button, a new screen showing a map along with suggested route to the destination will be displayed. The suggested route should minimize trip duration and delays.
- *Administrator*: The manager or administrator oversees the operation and specifically addresses two issues—taking care of the accounting/financial management and dealing with complaints posted by customers. Hence, the CoRide app provides the administrator two options in the form of buttons—*Financial Report* and *Customer feedback*. When the *Financial Report* button is clicked, a new screen appears where reports can be generated by querying the Transaction database. When the *Customer feedback* button is clicked, a new screen appears where the administrator can read customer reviews about drivers and will also have an option to block drivers with negative feedback.

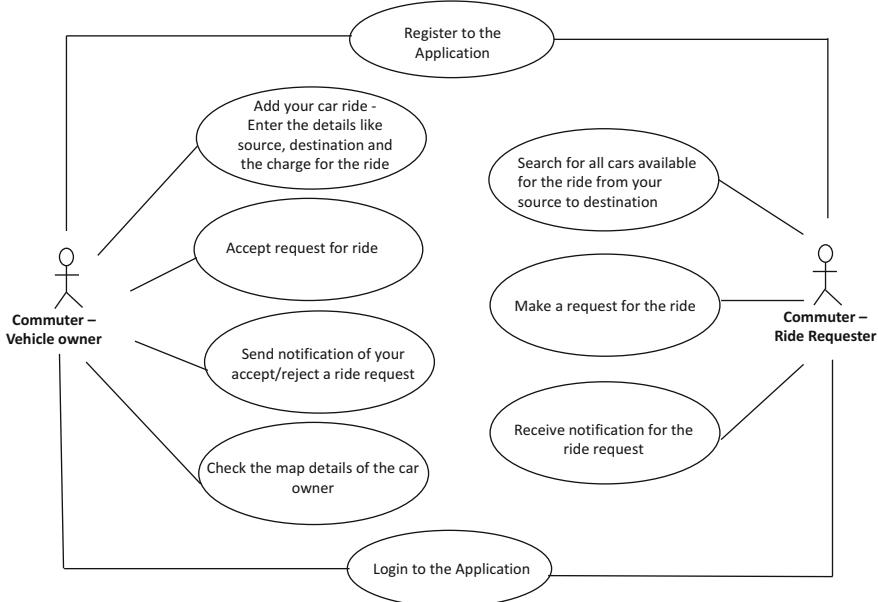


Fig. 7.17 Use case diagram for CoRide

The primary use cases for CoRide are shown in Fig. 7.17:

- *Registration:* General personal information including address, username and password choice.
- *Log-in:* Asking for username and password.
- *Requesting a ride:* Position is shown on the map, and cars in the area or close vicinity are shown as well as passenger information.
- *Accepting a passenger:* First come first serve principle. Driver's information is shown. Passenger can cancel a specific driver until he or she arrives.
- *Passenger pick-up:* Driver arrives and picks up passenger, destination will be specified, and ride transaction is initiated.
- *Ride:* Car's position is shown on the map. Distance of the ride is tracked.
- *Check-out ride:* Passenger leaves the car. Incentive points are booked on the driver's account and deducted from the passengers account.
- *Feedback (optional):* The passenger can rate the condition and cleanliness of the car and the driving skills of the car owner.
- *Reporting:* All users can check the status of their incentive points online.

7.9.2 OOA

The CoRide app is a self-contained application which operates on Android-based mobile phones. The app allows users to register themselves so that it acts as a platform where people who would like to share their rides and people in need of a ride come together. The app generates a bill for each user, reflecting the share of the total cost, although this does not involve any money transactions as the system works with incentive points. The CoRide app provides the facility to book a shared cab and addresses all kinds of ride-sharing needs.

The product functions of the CoRide app offer the following three main functionalities to the user:

- Facility to share a cab with others on a short-distance trip within the city
- A platform to find co-passengers for carpooling, especially for students and office employees who commute regularly between home, office, and campus.
- Facility to find co-passengers for sharing a ride on a long-distance trip

CoRide can be used by two categories of users:

- People who are looking for a ride
- People who are willing to share their vehicle for a ride

A person can belong to both types.

This section continues the development of use cases and requirements, discusses the main functionality, and looks at some aspects of the object-oriented analysis for CoRide.

Let us take a closer look at the core functionality of CoRide as described in the data flow diagram of Fig. 7.18:

- To start with, every user has to register with the app through his e-mail ID and by choosing a username and password.
- After successful registration, the user has to login to the app typing in the correct username and password.
- Users can share their vehicle for a particular ride with someone who wants to go to the same destination or to a destination on the same route. If the user is willing to share, the user has to go to the “Add Vehicles” tab and provide the details of the car, destination, seats available, and his preferred charge for a seat. After providing the details, the user has to click on the submit button.
- Now any user who is looking for a ride can click on the “View Vehicles Available” tab and provide the starting position (source in CoRide terminology) and destination. A list of vehicles will be shown for which this particular journey fits. By clicking on any car from the list a request will be sent to the person who is driving that particular car.
- The “Notifications Tab” informs the driver about an incoming request. The driver can accept or reject.

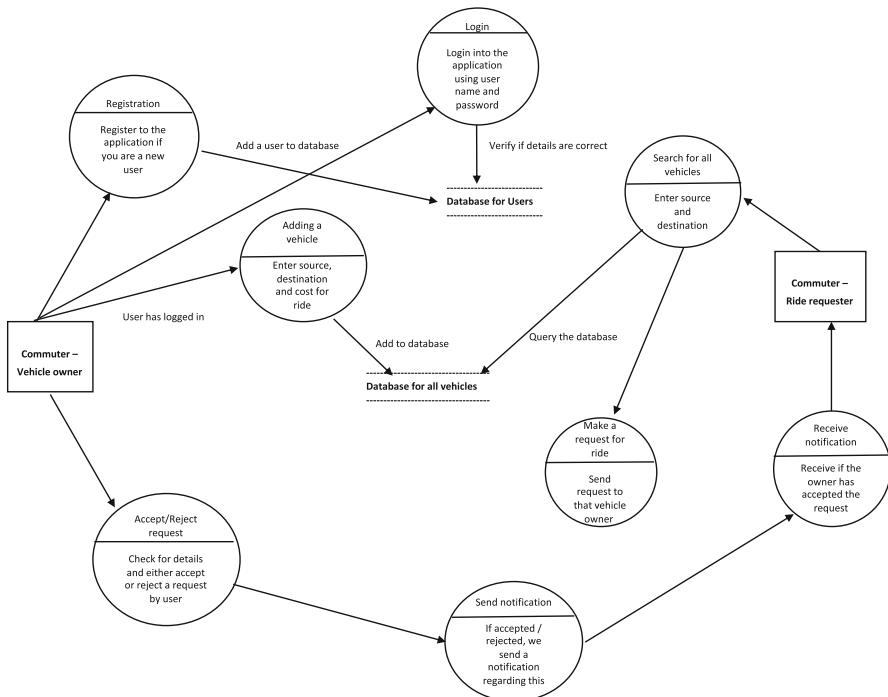


Fig. 7.18 Data flow diagram (DFD) for CoRide

- The Requester will be notified if his/her request is accepted, again through the “Notifications Tab.”
- Now the person who has accepted the request can go to “Accepted Users Tab” and request for the location of the user whose request he/she accepted.

Figure 7.19 shows a simplified OOA model of CoRide with the main classes and their associations (Jindal et al. 2016). Attributes and methods are hidden as the focus is on the domain model.

7.9.2.1 Nonfunctional Requirements

The CoRide app should provide real-time information about available rides and handle real-time transactions. Therefore, performance is a key issue. Time critical functions are database queries, route computations, client-server interactions and bandwidth of the mobile connections (see also URL13 2017). All these factors should be monitored carefully. During database queries, sending the details to drivers and customers and storing the customer details in the database are critical. The ride suggestions provided to the user should be prompt, too.

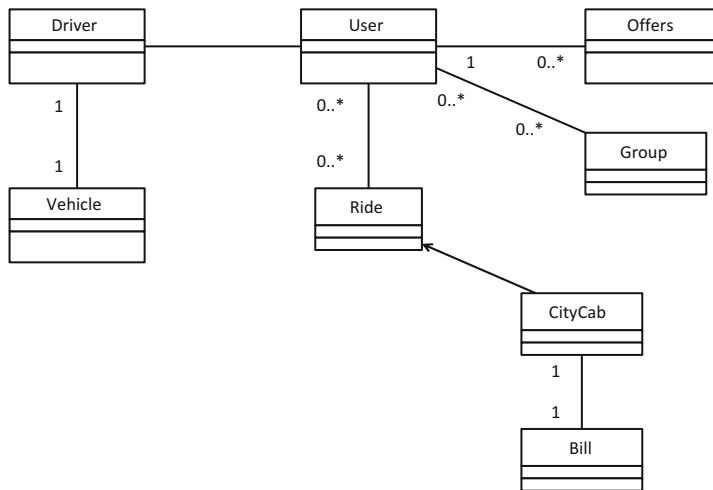


Fig. 7.19 Core classes and their associations in the CoRide app

The CoRide app is designed for the Android platform. As support for other platforms like iOS is planned for the future, portability is important and should be taken care of in the design phase of the app.

The CoRide app stores its data in various databases (DB). Hence, logical database requirements become relevant, w.r.t.:

- **Transactional Database**: The transactional database stores the details of all transactions in which the users are involved. It has the entire details about the drivers who offer the cab service. Currently MySQL is used as the transaction database.
- **Map Database**: All map-based information is handled by Google Maps or Mapbox (URL16 2017). This information is used for finding the routes in the map as well as locating the app users.

Figure 7.20 shows the sequence diagram of the interaction between customer, admin, and driver. Figure 7.21 illustrates the communication between carpoolsing users and CoRide server.

The core classes of the analysis model are driver, vehicle, user, city cab, group, ride, and bill. The parameters of these classes are listed below. Furthermore, it is discussed how these parameters can be mapped to database tables.

1. Driver:

- **driver_id**: Identity number of the driver who took the ride. This is the primary key of this table.
- **name**: Name of the driver who registers himself with CoRide app.
- **address**: Address of the driver.

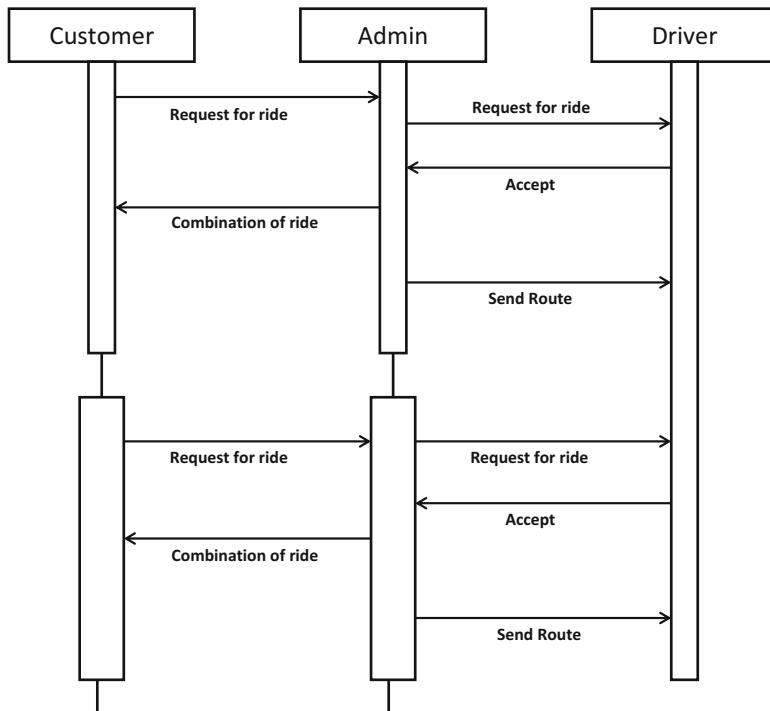


Fig. 7.20 Sequence diagram—interactions between customers, administrator, and drivers offering rides

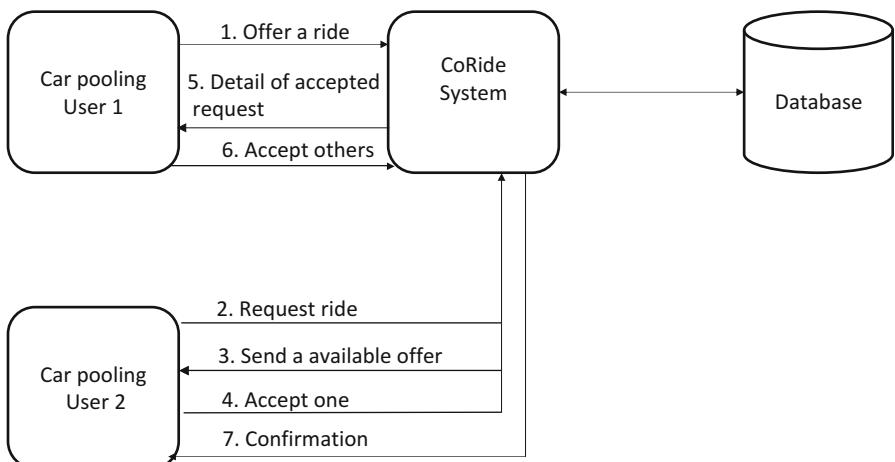


Fig. 7.21 Communication diagram of the CoRide app

- driving_license_number: Driving license number of the driver.
- PAN: personal account number of the driver.
- vehicle_id: The unique identity number generated by the DB for the vehicle the driver uses. It has a foreign key relation with “vehicle_id” field of “vehicle” Table.

2. Vehicle:

- vehicle_id: The unique identity number generated by the DB for the vehicle the driver uses. This is the primary key of this table.
- vehicle_number: The vehicle registration number of the vehicle.
- model: Model of the registered vehicle.
- capacity: Number of people who can travel in the vehicle.

3. User:

- user_id: The unique identification number generated by the DB for the user who registers with the CoRide app. This is the primary key of this table.
- name: Name of the user.
- address: Address of the user.
- gender: Gender of the user.
- phone_number: Phone number of the user.

4. City Cab:

- ride_id: Identity number given from the database to any city cab ride. It has a foreign key relation to the “ride_id” column in the “ride” table. This value would be unique for each and every ride. This field will be the primary key for this table.
- driver_id: Identity number of the driver who took the ride.
- distance: Total distance from source to destination in the particular ride.
- source: The starting point of the ride, i.e., the place from where the first person steps in.
- destination: The ending point of the ride or the place where the last person steps down.
- tolerance: The maximum delay (in percentage of total time) that can be tolerated by the rider.
- gender_preference: The preferred gender of co-passengers, which was given by the first incoming rider.

5. Group:

- group_id: The identity number given from the DB for a group of passengers who share a ride. This number will be unique for each shared ride. All the riders sharing the same ride (in any of the three possible options given) will have the same group_id. This field will be the primary key for this table.
- user_id: The unique identity number of a passenger in the group.
- ride_id: The unique number identifying the ride (refers the ride table).

6. Ride:

- ride_id: The unique identifying number given from the DB to each ride, regardless of the type which it belongs to. This field will be the primary key for this table.
- type: The type of the ride (carpooling, city cab or long-distance ride sharing).
- time-stamp: The time-stamp recorded when the ride begins.

7. Bill:

- bill_id: The unique identifying number given from the DB to each bill generated by the app. This field will be the primary key for this table.
- user_id: The user_id which uniquely identifies the user who receives the bill.
- ride_id: The ride_id identifies the particular ride.
- driver_id: The driver_id identifies the driver who was driving the cab.
- amount: The amount in the bill which the customer should pay.
- distance: The distance traveled by the customer to whom the bill is issued.
- timestamp: The time at which the bill was issued.

7.9.3 Design

The design and implementation of the CoRide app faces some design constraints since it involves interaction with several APIs. The server must be able to communicate with the Android app designed for customers and drivers, and it should be able to extract the location of the concerned stakeholder with the help of GPS. Also, the server must be able to interact with the map API (Google Maps or Mapbox), the map database, and the transaction database to implement the functionalities. The transaction database will be deployed using the services of a parse cloud.

The design of CoRide is based on the following assumptions and dependencies:

- The location of the app user provided by the GPS is accurate.
- The distances, location, and other map-related attributes are accurate enough.

All hardware interfacing will be handled by Android OS. By using the standard programming tools for Android OS (Android SDK), an efficient hardware abstraction is possible; hence, one does not need to go into the details of a specific implementation of the GPS interface.

CoRide uses the Google Maps API to identify a specific location and present this on a map. A MySQL database is used for storing all data.

CoRide will take location information and combine that with decisions based on user set preferences to query the maps server, select suitable results, and then present them to the user.

The backend required for CoRide will be deployed using cloud services.

Figure 7.22 illustrates the architecture of the CoRide system.

7.9.3.1 Client Server Communication

CoRide uses a REST API for client-server communication. REST stands for *representational state transfer*, which is essentially an “architecture for networked applications.” In other words, it’s a set of standards that describe how computers should communicate with each other and with applications across a network. REST defines certain specific operations that applications should be able to do in order to satisfy all of the CRUD (create, read, update, delete) requirements. HTTP is the

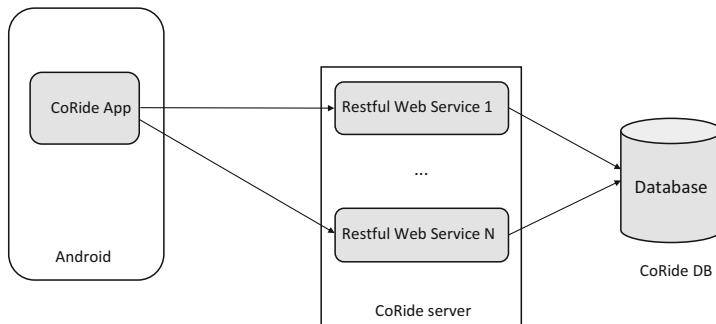


Fig. 7.22 Architecture of the CoRide application

protocol most frequently used to implement the REST architecture, supplying operations like PUT, POST, GET, DELETE, and HEAD.

Unlike SOAP-based web services, there is no “official” standard for RESTful web APIs. This is because REST is an architectural style, while SOAP is a protocol. Even though REST is not a standard per se, most RESTful implementations make use of standards such as HTTP, URI, JSON, and XML.

Building RESTful web services is part experience and part intuition. As RESTful web services don’t follow a prescribed standard except for HTTP; it’s important to build a RESTful API in accordance with industry best practices to ease development and increase client adoption.

In the CoRide project the REST API is used to send information entered by the user about origin and destination to the server. The server in turn directs this information to the selected driver.

Figure 7.23 shows a screenshot of the Android Studio integrated development environment (IDE) with the CoRide project opened in the project browser (left side) and a snippet of the SQL code used.

7.9.4 The Ridematching Algorithm

The CoRide app uses its own algorithm to decide how to match user requests for a shared cab ride to available drivers. While doing so, the algorithms takes into consideration the maximum time the user is willing to wait (tolerance). The CoRide app identifies all drivers who are in close vicinity of a particular user, who requests a shared ride, and also ensures that the capacity of the cab is sufficient (should not be already full) and the route requested by the new passenger perfectly matches with the route of passengers who are already traveling in the cab.

The screenshot shows the Android Studio interface with the project 'CoRideProject' open. The left sidebar displays the project structure under '1. Project' and '2. Structure'. The right pane shows a large block of SQL code for managing tables like 'Location', 'bill', and 'carpool_transaction'. The code includes various SQL statements such as 'CREATE TABLE', 'ALTER TABLE', 'INSERT INTO', and 'DROP TABLE'. The SQL code is color-coded for syntax highlighting.

```

coride.sql - CoRideProject - [~/Desktop/CoRideProject]
Project CoRideProject ~/Desktop/CoRideProject
  +-- CoRideProject
    +-- .idea
    +-- CoRide-app
    +-- app
      +-- src
        +-- androidTest
        +-- main
          +-- Java
            +-- com
              +-- example
                +-- devlesh
                  +-- Coride
            +-- res
              +-- AndroidManifest.xml
        +-- .gitignore
        +-- build.gradle
        +-- google-services.json
        +-- proguard-rules.pro
      +-- gradle
        +-- gradle.properties
        +-- gradlew
        +-- gradlew.bat
        +-- settings.gradle
    +-- coride.sql
    +-- Coride.war
    +-- Dataflow diagram for SaveWise.png
    +-- DataFlowCoride.png
    +-- QuestionsToCoRideTeam.rt
    +-- README.md
    +-- REST_Server.war
    +-- Ride_sharing_app.pdf
  +-- Terminal TODO
  +-- Event Log
Frameworks detected: Android framework is detected in the project Configure (today 15:53)

```

Fig. 7.23 The CoRide app in Android Studio, displaying the project structure on the left side and the SQL code in the right window

The CoRide user provides key input like his/her name, origin, destination, time of travel, and other necessary information. First of all, requests with exactly the same origin and destination are selected. Then those requests are selected and sent to the drivers which have a starting position (source in CoRide terminology) and destination that is on the way or within user-specified tolerance time of already accepted requests. The driver accepts or rejects the request. When a driver rejects a request, the request will be forwarded to other available drivers.

The Google Maps API is used to find all available drivers nearby (within a radius of 1 km of the passenger's source location). When a driver accepts a request, based on the source and destination of a passenger, a map will be provided to the driver. When a new request comes from a nearby location and the cab is not full, the algorithm calculates the distance between current location of the cab and the new user's source location, and then it also calculates the new distance if the request is accepted. Now the algorithm will compare the existing distance with the new distance. If the difference is below a threshold given by the passengers in the cab, then the new map will be generated for the driver; else this new request will be forwarded to another driver.

Let u_1 and u_2 be two users/passengers of the CoRide app. Let s_1, s_2 and d_1, d_2 be the source and destination of users u_1 and u_2 , respectively. Let c be the maximum capacity of the vehicle and e be the exact current location of the cab. Let $distance(x,y)$ be the distance between locations x and y . The algorithm works as follows:

- When driver accepts a request of u_1 , a map is generated from s_1 to d_1 .
- Calculate $distance(s_1, d_1)$ and let it be the current distance. When a new request comes from u_2 , calculate $distance(e, s_2) + distance(s_2, d_2) + distance(d_2, d_1)$ and let it be the new distance. If the number of passengers already in the cab is less than c and the estimated time for the calculated new distance is less than the time deviation preferred by existing users, i.e., $(estimated_time(new\ distance)) - estimated_time(actual\ distance)) \leq time\ deviation$, then the new request will be sent to the driver along with the newly generated map including the new source and destination.
- Repeat as often as requests come in.

7.9.5 Using Google Maps

An API is a set of methods and tools that can be used for building software applications. The *Google Maps API* lets one customize maps and the information on maps (URL17 2017). Google Maps APIs are available for different platforms: Web, Android, and iOS. These native platform APIs are integrated into the CoRide suite of web services. With the Google Maps Android API, one can add maps based on Google Maps data to any application. The API automatically handles access to Google Maps servers, data downloading, map display, and response to map gestures. One can also use API calls to add markers, polygons, and overlays to a basic map and to change the user's view of a particular map area. These objects provide additional information for map locations and allow a client to interact with the map.

The API supports the following graphical elements:

- Icons anchored to specific positions on the map (markers)
- Sets of line segments (polylines)
- Enclosed segments (polygons)
- Bitmap graphics anchored to specific positions on the map (ground overlays)
- Sets of images which are displayed on top of the base map tiles (tile overlays)

If one wants to use the Google Maps Android API in an application, one must include the Google Play Services attribution text as part of a “Legal Notices” section. It is recommended to include legal notices as an independent menu item or as part of an “About” menu item. The Google Maps Android API includes built-in support for accessibility. The accessibility features are automatically enabled for any application using the API.

7.9.5.1 The Google Maps Directions API

The Google Maps Directions API is a service that calculates directions between locations using an HTTP request for several modes of transportation, including transit, driving, walking, or cycling. Directions may specify origins, destinations, and waypoints either as text strings (e.g., “Silk Board” or “Koramangala”) or as latitude/longitude coordinates. The Directions API can return multipart directions using a series of waypoints.

Directions Requests

A Google Maps Directions API can process information as:

- JavaScript Object Notation (JSON)
- XML

The API can be accessed over HTTP:

http://maps.googleapis.com/maps/api/directions/output?parameters

HTTPS is recommended for applications that include sensitive user data, such as a user's location. Google Maps Directions API URLs are restricted to approximately 2000 characters, after URL encoding. As some Google Maps Directions API URLs may involve many locations along a path, it is important to be aware of this limit when constructing a URL.

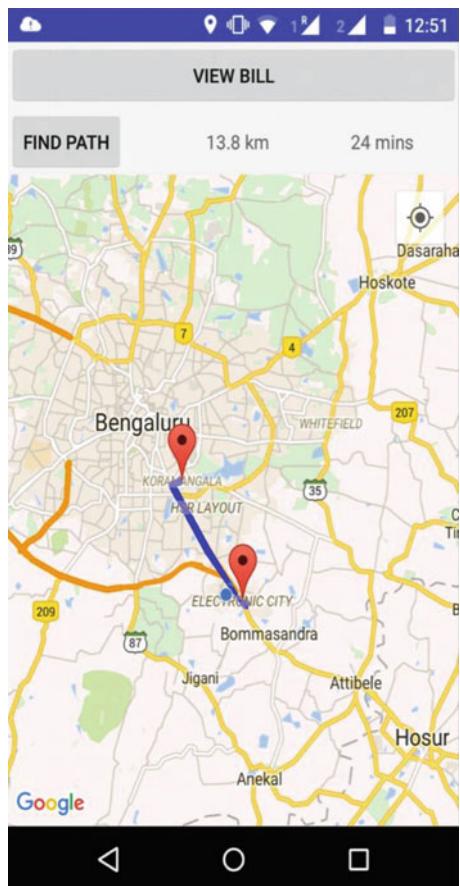
The HTTP request includes the following parameters:

- *origin*—The address from which one wants to calculate the directions. If an address is passed, the Directions service will geocode the string and convert it to a latitude/longitude coordinate to calculate directions. This coordinate may be different from that returned by the Google Maps Geocoding API if, for example, a building entrance is given rather than the center of the building.
- *destination*—The address, textual latitude/longitude value, or place ID of the destination. If one transmits an address, the Directions service will geocode the string and convert it to a latitude/longitude coordinate to calculate the directions. This coordinate may be different from that returned by the Google Maps Geocoding API.
- *key*—The application's API key. This key identifies the application for purposes of quota management.
- *waypoints*—Specifies an array of waypoints. Waypoints alter a route by routing it through the specified location(s). A waypoint is specified as an address which will be geocoded.

When calculating routes using the Google Maps Directions API, one may also specify waypoints for driving, walking, or bicycling directions. Waypoints are not available for transit directions but can be used to calculate routes through additional locations, in which case the returned route includes stopovers at each of the given waypoints.

Waypoints can be provided as locations separated by the pipe character (!), in the form of an address. To calculate directions the Google Maps Directions service will geocode the string and convert it to a latitude/longitude coordinate. Note that these coordinates may be different from that returned by the Google Maps Geocoding API, depending on the particular position taken, e.g., the entrance or the center of a building. Figure 7.24 is a screenshot of the app that shows the route on Google Maps.

Fig. 7.24 The CoRide app shows distance and estimated time of arrival on Google Maps



7.9.6 A Code Walk Through

In this section, we take a look at the source code. The complete system contains many thousand lines of code. The objective here is not an in-depth discussion but to give an overview of some key concepts and their implementation on the Android platform.

After loading and starting the CoRide app, the following home screen will appear (see Fig. 7.25). When the user is already registered, he or she can directly login pressing the LOGIN button. If not, a driver should click the DRIVER REGISTRATION button for filling in the required details, and a regular user, i.e., passenger, should click the USER REGISTRATION button to complete his or her registration.

We will now discuss the implementation of this home screen functionality in Android. The program itself is called an *Activity* and the appearance on the screen is called the *View*.

Fig. 7.25 CoRide app home screen



The View displays the title of the Activity CoRide app and shows three buttons that can be clicked (Login, User Registration, and Driver Registration).

The complete layout of the user interface elements, also called widgets or controls, can be generated by the built-in GUI builder which creates an XML representation. The builder, like Apple's Xcode Interface Builder, allows the visual configuration of the user interface without a need for direct coding (what you see is what you get, WYSIWYG). The software developer can switch between the XML description and the visual design back and forth. The XML representation is stored in a resource file which contains the XML description of all app components, e.g., the layout (for the View), values, dimensions, and strings.

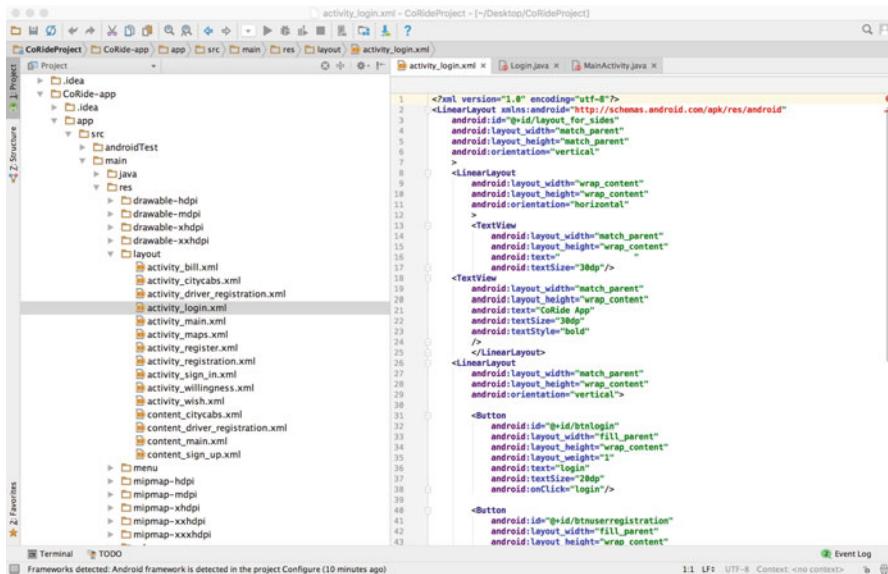


Fig. 7.26 Login View in Android Studio. Resource browser on the left side, XML file with the layout on the right side

The listing below shows the XML code of the View which was generated by the GUI builder. Figure 7.26 is a screenshot of Android Studio which shows the project browser and the XML code for the layout.

```

<LinearLayout xmlns:android=
    "http://schemas.android.com/apk/res/android"
    android:id="@+id/layout_for_sides"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal" >
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text=""
            android:textSize="30dp" />
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="CoRide App"
            android:textSize="30dp" />
    
```

```
        android:textSize="30dp"
        android:textStyle="bold" />
    </LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">
    <Button
        android:id="@+id/btnlogin"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="login"
        android:textSize="20dp"
        android:onClick="login"/>
    <Button
        android:id="@+id/btnuserregistration"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="user registration"
        android:textSize="20dp"
        android:onClick="registration"/>
    <Button
        android:id="@+id/btn_calc_result"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="driver registration"
        android:textSize="20dp"
        android:onClick="driverregistration" />
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="fill_parent"
    android:orientation="horizontal"
    android:background="@drawable/zimride" >
</LinearLayout>
</LinearLayout>
```

The onscreen position of GUI elements, for example, controls or widgets, is managed by so-called layout managers. Here, a linear layout is chosen which groups all controls in a single horizontal or vertical line.

The definition `android:orientation = "vertical"` defines a vertical layout.

The View consists of a title text (the CoRide app title) and three buttons. This user interface can be generated with a simple *TextView* layout enclosed between the XML element <TextView ... /TextView> and a *Button* layout (XML <Button ... /Button>).

The layout width is set with the *match_parent* construct, which effectively means that the widgets will be aligned with the parent view. The *weight* attribute defines the relative size of different widgets in the layout. In the case of the buttons, it is set to 1, and this means that all will appear in the same size. The layout has a unique identifier

android:id="@+id/layout_for_sides" with which it can be identified and referenced in the code. The buttons have a *text* attribute, e.g., android:text = "driver registration". This refers to the text that appears on the button and gives a hint of its functionality. The buttons also have identifiers and define a command/message event, e.g., android:onClick = "driverregistration" that is generated if the user clicks on them.

If maximum portability and multi-language features are important, the *TextView* layout should be modified. In the above code, the *textstring* is hardcoded by the XML construct android:text = "CoRide App". If we want to deploy the app in different countries and would like to change the login title screens later on, one could use a string resource instead:

android:text="@string/app_title". Now this string would refer to a string in the resource file where it would be set to a particular value

<string name="app_title">CoRide App</string>. The string itself is stored in the resource folder *values* as an XML element and therefore could be changed at any time to multiple target languages providing easy portability as there is only one location where the changes have to be done. This clearly shows the power of a declarative user interface description.

Android uses this mechanism for many platform-dependent parameters like measurement dimensions, colors, and user interface strings.

Android supports the model view controller (MVC) pattern which is based on a clear demarcation between:

- The model, i.e., the classes representing users, drivers etc. (see Fig. 7.20 from the OOA section)
- The View, the user interface, i.e., the View generated by the XML layout file
- The controller, a Java class that receives events from the View (the user interface) and interacts with the model, modifying parameters and calling methods of the objects (see Fig. 7.27)

The code below shows the corresponding controller for the Login View. It is a Java class *Login* which is a subclass of *AppCompatActivity* which provides the basic functionality in Android to process user interface events and coordinate subsequent processing stages.

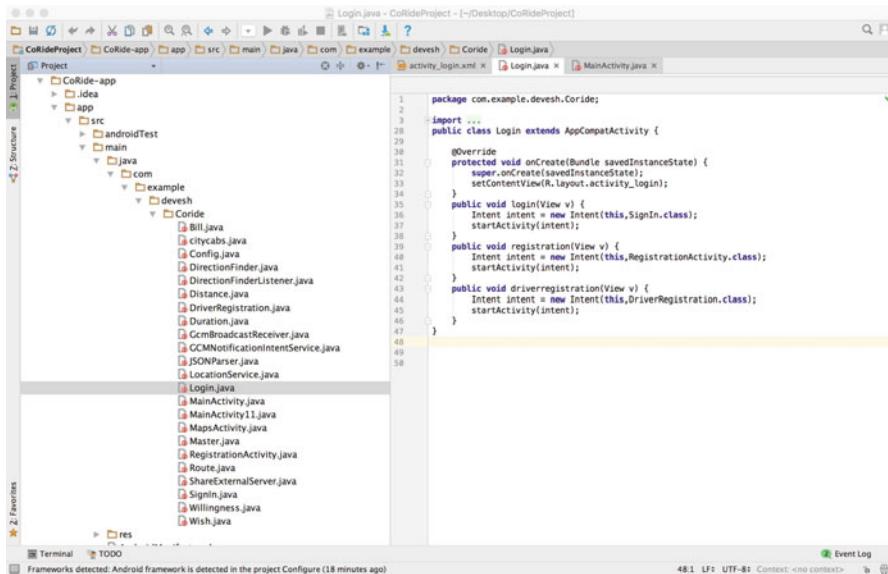


Fig. 7.27 The Login class shown in Android Studio which processes the information from the login-activity view

```
package com.example.devesh.Coride;

import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.pm.PackageInfo;
import android.content.pm.PackageManager;
import android.os.AsyncTask;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.text.TextUtils;
import android.util.Log;
import android.view.View;
import android.widget.AutoCompleteTextView;
import android.widget.EditText;
import android.widget.Toast;

import com.google.android.gms.gcm.GoogleCloudMessaging;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
```

```
import java.net.URL;
import android.widget.Button;

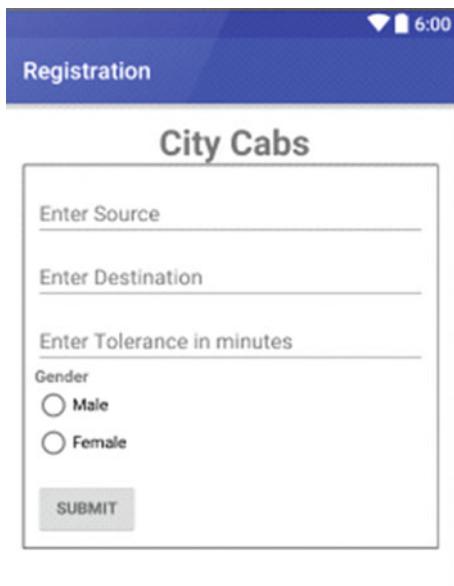
public class Login extends AppCompatActivity
{
    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
    }
    public void login(View v)
    {
        Intent intent = new Intent(this,SignIn.class);
        startActivity(intent);
    }
    public void registration(View v)
    {
        Intent intent = new Intent(this,
                RegistrationActivity.class);
        startActivity(intent);
    }
    public void driverregistration(View v)
    {
        Intent intent = new Intent(this,DriverRegistration.class);
        startActivity(intent);
    }
}
```

The application package is defined as `com.example.devesh.CoRide`. It contains all classes of the CoRide app and is mapped to a corresponding folder structure (`/com/example/devesh/CoRide`). The first lines of the code imports various Java packages that are being used.

The code starts with extending the `AppCompatActivity` class which is the master class for the activities. The `onCreate()` method is called first and creates the Activity. The layout for the view is specified in `setContentView (R.layout.activity_login)`. This links the View with the Login class such that the user interface events can be processed. When a button is pressed, the corresponding command method in the Login class is called, e.g., the button `DRIVERREGISTRATION` sends the command `driverregistration` which evokes the method:

```
public void registration(View v)
{
    Intent intent = new Intent(this,
            RegistrationActivity.class);
    startActivity(intent);
}
```

Fig. 7.28 The ride-sharing call a CityCab's view



The method in turn creates an *Intent* class that handles the driver registration process. The methods for login and registration are evoked in a similar way and create the corresponding Intent classes for further processing.

Figure 7.28 shows the View for calling a city cab shared ride. There are several editable text fields (the text can be changed by the user) for starting point (source in CoRide terminology), destination, and the acceptable waiting time (tolerance in CoRide terminology). A radio button group allows to specify the gender.

The listing below shows the XML code which was generated by the GUI builder. As the layout of the widgets is straight forward, a linear layout is sufficient to position the user interface elements.

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/
    android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center_horizontal"
    android:orientation="vertical"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.devesh.Coride.RegistrationActivity">
```

```
<!-- Login progress -->
<ProgressBar
    android:id="@+id/login_progress"
    style="?android:attr/progressBarStyleLarge"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:visibility="gone" />
<LinearLayout
    android:id="@+id/email_login_form"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:weightSum="1">
    <android.support.design.widget.TextInputLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <EditText
            android:id="@+id/source"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Enter Source"
            android:imeActionId="@+id/login"
            android:imeActionLabel=
"@string/action_sign_in_short"

            android:imeOptions="actionUnspecified"
            android:inputType="textPersonName"
            android:maxLines="1"
            android:singleLine="true" />
    </android.support.design.widget.TextInputLayout>
    <android.support.design.widget.TextInputLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <EditText
            android:id="@+id/destination"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Enter Destination"
            android:imeActionId="@+id/login"
            android:imeActionLabel=
"@string/action_sign_in_short"

            android:imeOptions="actionUnspecified"
```

```
        android:inputType="textPersonName"
        android:maxLines="1"
        android:singleLine="true" />
    </android.support.design.widget.TextInputLayout>
    <android.support.design.widget.TextInputLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <EditText
            android:id="@+id/tolerance"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Enter Tolerance"
            android:imeActionId="@+id/login"

            android:imeActionLabel=
            "@string/action_sign_in_short"

            android:imeOptions="actionUnspecified"
            android:inputType="textPostalAddress"
            android:maxLines="1"
            android:singleLine="true" />
    <RadioGroup
        android:id="@+id/radioGroup1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true" >
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Gender"
            android:textStyle="bold"
            />
        <RadioButton
            android:id="@+id/radio0"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/male" />
        <RadioButton
            android:id="@+id/radio1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/female" />
    </RadioGroup>
</android.support.design.widget.TextInputLayout>
<Button
```

```
        android:id="@+id/email_sign_in_button"
        style="?android:textAppearanceSmall"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:text="Submit"
        android:textStyle="bold"
        android:onClick="sendData"/>
    
```

```
</LinearLayout>
</LinearLayout>
```

The input controls for the editable text are being defined by the XML element `<EditText>`. One such textfield is being created for every parameter that can be keyed in. Again, the element is identified by an *id* that is used to refer to it. The radio buttons are aggregated by a `<RadioGroup>` element and specified one by one with the construct `<RadioButton>`.

The next listing shows the corresponding controller that processes the user interface inputs and interacts with the objects of the domain model.

```
package com.example.devesh.Coride;
import android.Manifest;
import android.content.Context;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.location.Location;
import android.location.LocationManager;
import android.os.AsyncTask;
import android.os.Bundle;
import android.support.v4.app.ActivityCompat;
import android.support.v4.content.ContextCompat;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.View;
import android.widget.Toast;

import java.util.ArrayList;

import android.widget.RadioButton;
import android.widget.RadioGroup;

import android.util.Pair;
import android.widget.EditText;

import java.io.BufferedReader;
import java.io.IOException;
```

```
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;

/**
 * A login screen that offers login via email/password.
 */
public class citycabs extends AppCompatActivity {
    Master master = new Master();
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_citycabs);
    }
    public void sendData(View view) throws MalformedURLException {
        Log.e("TEST LOG", "ho gya, aa gya loop me");
        Toast.makeText(this,"coride",Toast.LENGTH_SHORT).show();
        Master master=new Master();
        String url_string =master.url +"requestride";
        EditText source;
        EditText destination;
        EditText tolerance;
        RadioGroup rg;
        LocationManager locationManager;
        String ssource;
        String sdestination;
        String stolerance;
        Double latitude=0.00,longitude=0.00;

        int count = 0;

        rg = (RadioGroup) findViewById(R.id.radioGroup1);
        source =(EditText)findViewById(R.id.source);
        destination=(EditText)findViewById(R.id.destination);
        tolerance = (EditText)findViewById(R.id.tolerance);
        ssource= source.getText().toString();
        master.s1=ssource;

        sdestination= destination.getText().toString();
        master.d1=sdestination;
        stolerance= tolerance.getText().toString();

        String gender=
            ((RadioButton)findViewById(
                rg.getCheckedRadioButtonId()))).getText().toString();
```

```
if (ContextCompat.checkSelfPermission(
        this, Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED)
{
    Log.e("shishir", "I am here dude");
    ActivityCompat.requestPermissions(this, new String[] {Manifest.permission.ACCESS_FINE_LOCATION}, 0);
}
locationManager =
        (LocationManager) getSystemService(
                Context.LOCATION_SERVICE);
location = locationManager.getLastKnownLocation(
        locationManager.NETWORK_PROVIDER);
if(location==null)
{
    Toast.makeText(
            getApplicationContext(),
            "Location Disabled Enable it",
            Toast.LENGTH_SHORT).show();
}
else
{
    latitude = location.getLatitude();
    longitude = location.getLongitude();
}
Master obj1=new Ma
Master obj1=new Master();
if(count == 0)
{
    String json =
        "[{"source:\\""+ssource+
        "\",{"destination:\\""+sdestination+
        "\", {"tolerance:\\""+stolerance+
        "\", {"gender:\\""+gender+
        "\", {"latitude:\\""+latitude+
        "\", {"longitude:\\""+longitude+
        "\", {"mobile:\\""+obj1.mobile+"\\"}}]}";
SendDataTask snd = new SendDataTask(json);
//Go to sendData class from here
URL url = new URL(url_string);
snd.execute(url);
Intent intent = new Intent(this,MapsActivity.class);
startActivity(intent);
}
}
```

```
protected Void doInBackground(URL... urls) {
    try {
        String a ="ff";
        URL url = urls[0];

        HttpURLConnection conn =
            (HttpURLConnection)url.openConnection();

        conn.setDoOutput(true);
        conn.setRequestMethod("POST");
        conn.setRequestProperty("Content-Type",
            "application/json");

        OutputStream os = conn.getOutputStream();

        System.out.println("input : "+json);
        os.write(json.getBytes());//Sendin the json object
        os.flush();

        BufferedReader br =
            new BufferedReader(
                new InputStreamReader(
                    (conn.getInputStream())));
    }

    String output;
    System.out.println("Output from Server .... \n");

    while ((output = br.readLine()) != null) {
        System.out.println(output);
    }
    conn.disconnect();
}
catch (IOException i){
    i.printStackTrace();
}
return null;
}
```

The code defines the class `citycabs` as a publicly accessible class `citycabs` extends `AppCompatActivity`. Like the `Login` class, it is defined as a subclass of `AppCompatActivity` and links this java code with the activity layout file in the resource folder `setContentView(R.layout.activity_citycabs)`.

Let us concentrate on how the value of the edit text destination is read and processed. The class defines a variable (using the full qualification):

```
android.widget.EditText destination;
```

and reads the values of the widgets from the view by identifying them with their identity code:

```
destination = (EditText) findViewById(R.id.destination);
```

The `findViewById()` method retrieves a reference and converts it back to an `EditText` class. The current value is being extracted with the `getText()` method and converted into a *string* by calling `toString()`.

```
sdestination= destination.getText().toString();
```

The source string is saved in a master class which provides a simple container for managing trip requests.

The parameters are aggregated and form a JSON object which is needed for the communication with the REST-based server.

The following code section constructs such a JSON string:

```
String json =
    "[{"source:"+ssource+""}, {"destination:"+ sdestination+
    "\", {"tolerance:"+stolerance+""}, {"gender:"+gender+
    "\"}, {"latitude:"+latitude+""}, {"longitude:"+longitude+
    "\"}, {"mobile:"+obj1.mobile+""}]";
```

JSON strings can be parsed by a separate parser class, the JSON Parser, which is provided in the CoRide app.

7.10 Summary and Recommended Readings

Balzert (2009; 2011) gives a detailed overview of many aspects of software engineering. Sommerville (2015) is a classic introduction to software engineering with an emphasis on agile methodologies. Schäfer (2010) focuses on software architecture and discusses the special role of software architects in large and complex software projects. Grechenig et al. (2010) discuss different software engineering methodologies from a practical point of view. Various case studies from different application fields show the trade-offs between different technologies and methods. Masak (2010) gives a detailed overview of architecture decisions in large-scale software systems. Kernighan and Pike (2006) focus on the coding aspect of software

engineering. Booch et al. (2004) is the classical textbook on object-oriented software engineering and the UML.

Mangiapane and Büchler (2015) and Laudon et al. (2010) give a good introduction to the processes, technologies, and regulations of modern IT management. Werle (2015) shows how the digital transformation turns the car into a “smartphone on wheels.” Steinacker (2016) reflects upon the digital transformation and discusses the rising importance of software that controls more and more aspects of human life. Haas and Schreiner (2002) give an overview of Java enterprise technologies and enterprise software architectures.

Software is increasingly becoming a critical competence for car manufacturers. The first-tier suppliers have tens of thousands of software specialists. Randak (2016) looks at the skill sets necessary in the future and discusses the rising role of the first-tier suppliers.

Müller et al. (2015) discuss a case study of agile development methodologies for HMI.

Kroker (2015) shows how production processes in the automotive industry are being transformed by the app business. Backend systems and cloud integration are important topics for automotive OEMs. Kroker (2016) shows how Frankfurt becomes the center for many of the European cloud activities.

The connected car is at the center of many innovations. Jung and Kalmar (2015) show a concept how this data can be shared in a secure way that respects the privacy of the owner/driver. This can be achieved by anonymizing the vehicle identification numbers and blurring critical information. Kacher (2016) discusses some critical aspects of the rapid digital transformation.

For a general discussion on automotive and Car IT software development, see Hülshorst et al. (2015), Herchet et al. (2015), Drabek and White (2013), and Burkert (2013).

Johanning and Mildner (2015) give a good overview of the different HW and SW aspects of car IT. A general discussion of connected cars is given in Vembo (2016). As the car becomes more and more connected, the integration with IoT platforms is important. Balani (2015) gives a good overview of the leading IoT platforms (IBM Watson IoT, GE Predix, etc.) and discusses their key features.

The rise of Apple and the decline of Nokia, the leading smartphone manufacturer of the last decade, has become legendary. Lewis (1997) has many interesting examples of how other industries have thrived and vanished.

For a good overview and a general discussion of mobile computing, refer to Roth (2005).

For an in-depth discussion of app development on iOS, refer to Sadun and Wardwell (2014). A good introduction to Android app development is Meier (2012).

A general overview of the ARM processor and instruction set architecture, which is the basis for most modern smartphones, is given in Sloss et al. (2004).

Tanenbaum and Austin (2012) discuss different mobile processor architectures. Silberschatz et al. (2012) is a good introduction to hardware/software interfaces and virtualization technologies.

7.11 Exercises

What is meant by the term *mission and main tasks of the central IT organization in a large company*?

Describe the characteristics of mission and main tasks of the central IT organization in a large company.

What is meant by the term *chief information officer (CIO)*?

Describe the responsibilities the CIO has.

What is meant by the term *CDO*?

Describe why many organizations create the role of a CDO, and what are similarities and what are the differences to the role of a CIO?

What is meant by the term *IT landscaping*?

Describe the main characteristics of IT landscaping.

What is meant by the term *IT management organization*?

Describe the main domains of IT management for a mid-sized company.

What is meant by the term *performance with respect to an IT organization*?

Describe how to measure the performance of an IT organization.

What is meant by the term *shared services*?

Describe the main tasks of shared services.

What is meant by the term *revenue on IT in automotive companies*?

Comment on whether 2.2% of IT spending as a percent of revenue is too much or not even enough for automotive companies.

What is meant by the term *ERP*?

Describe why large, standardized ERP packages like SAP are popular?

What is meant by the term *mobile app economy*?

Describe what influence does the mobile app economy have on the classical IT management.

What is meant by the term *Car IT*?

Describe the main tasks of Car IT.

What is meant by the term *two-speed IT*?

Describe why companies talk about two-speed IT.

What is meant by the term *COTS software*?

Describe the benefits and rawbacks of COTS software.

What is meant by the term *system integration*?

Describe the challenges of system integration.

What is meant by the term *software architecture*?

Describe the goal of enterprise software architecture.

What is meant by the term *digital transformation*?

Describe what role IT plays in the digital transformation of organizations.

What is meant by the term *agile software*?

Describe the effort in an agile software project.

What is meant by the term *Agile Manifesto*?

Describe the objectives of the Agile Manifesto.

What is meant by the term *Scrum*?

Describe the main tasks of Scrum.

What is meant by the term *Scrum master*?

Describe the main tasks of a Scrum master.

What is meant by the term *product owner*?

Describe the main tasks of a product owner.

What is meant by the term *V-model*?

Describe how Scrum differs from the classical V-model development process.

What is meant by the term *extreme programming*?

Describe the main tasks of extreme programming.

What is meant by the term *software complexity*?

Describe the main parameters that play a major role in measuring software complexity.

What is meant by the term *automotive mobile app*?

Describe the development and application scenario of a mobile app for Android.

What is meant by the term *heterogeneity of a target platform*?

Describe how to handle the heterogeneity of a target platform.

What is meant by the term *PAI stack*?

Describe the main functionality.

What is meant by the term *PAI stack*?

Describe the main functionality w.r.t. the joint development by Daimler and IBM.

What is meant by the term *SOA*?

Describe the main functionality.

What is meant by the term *iPhone*?

Describe why the iPhone became so popular.

What is meant by the term *Objective-C*?

Describe the main functionality.

What is meant by the term *Xcode*?

Describe the main functionality.

What is meant by the term *Swift*?

Describe the key features of Apple's programming language Swift.

What is meant by the term *Java*?

Describe the differences between Java and Swift.

What is meant by the term *iOS*?

Describe the differences between iOS and Android.

What is meant by the term *Symbian*?

Describe the differences in comparison to iOS and Android.

What is meant by the term *Windows Phone*?

Describe the differences in comparison to iOS and Android.

What are the *core features of Android Studio*?

Describe the main functionality.

What is meant by the term *OBD*?

What value add could the driver get from access and visualization of OBD data?

What is meant by the term *Xamarin framework*?

Describe the main functionality.

What is meant by the term *mobile target operating system*?

Describe how cross development platforms for various mobile target operating systems work.

What is meant by the term *software quality*?

Describe the main features of software quality how to ensure the quality for software.

What is meant by the term *CMM*?

Describe the main features.

What is meant by the term *CMMI*?

Describe the main features.

What is meant by the term *Daimler's Journey to Excellence (J2E) Program*?

Describe the main features.

What is meant by the term *big data*?

Describe the role big data play for connected car services.

What is meant by the term *data streams*?

Describe how to handle the data streams coming from connected cars.

What is meant by the term *connected car*?

Describe some scenarios for connected cars in 2020 and 2025.

Describe the revenue sources for connected car data.

Give examples of *connected car data*?

Describe the similarities and the differences between smartphone data and connected car data.

What is meant by the term *location-aware use cases for connected cars*?

Describe some location-aware use cases for connected cars.

References and Further Readings

(Balani 2015) Balani, N.: Enterprise IoT – A Definite Handbook, Self published, Kindle Edition, 2016

(Balzert 2009) Balzert, H.: Textbook of Software Engineering: Basic Concepts and Requirements Engineering (in German). Spektrum Publ., 2009

(Balzert 2011) Balzert, H.: Textbook of Software Engineering: Design, Implementation, and Operation (in German). Spektrum Publ., 2011

(Bleske 2016) Bleske, C.: iOS Apps with Swift - The Easy Entry into the Development for iPhone, iPad and Co - including AppleWatch (in German). dpunkt Publ., 2016

(Booch et al. 2004) Booch, G, Rumbaugh, J, Jacobsen, I.: The UML User Guide, Addison-Wesley Publ., 2004

(Burkert 2013) Burkert, A.: Perspectives of Software-based Connectivity (in German). ATZ elektronik 01/2013

(Busse 2016) Busse, C.: With the first Smartphone began a tragic story (in German). August 12th 2016. Sueddeutsche online. Available from: <http://www.sueddeutsche.de/digital/nokia-communicator-mit-dem-ersten-smartphone-begann-eine-tragische-geschichte-1.3115519>

(Chan and Shaheen 2012) Chan, N. D., Shaheen, S. A.: Ridesharing in North America: Past, Present, and Future. In: Transport Reviews Vol. 32 No.1, pp. 93–112, 2012

(Dörner 2016) Dörner, S.: Microsoft's Long Good By to the Consumer (in German). Welt online. May 20th 2016. Available from: <https://www.welt.de/wirtschaft/webwelt/article155499927/Sobverabschiedet-sich-Microsoft-vom-Verbraucher>

(Drabek and White 2013) Drabek, C., White, G.: Better software models with a policy catalog (in German), ATZ elektronik, 03/2013

- (Grechenig et al. 2010) Grechenig, T., Bernhart, M., Breiteneder, R., Kappel, K.: Software Engineering - Case Studies from Real Development Projects (in German). Pearson Publ., 2010
- (Haas and Schreiner 2002) R. Haas und U. Schreiner (2002). Java technologies for enterprise applications, J2EE (in German). Carl Hanser Publ., 2002
- (Hecking 2016) Hecking, M.: Why Samsung with Note 7 had to pull the tear rope - Samsung's 4.5 billion Euro Firewall (in German). Manager Magazin online. October 11th 2016. Available from: <http://www.manager-magazin.de/thema/samsung/archiv-2016285.html>
- (Herchet et al. 2015) Herchet, H., Bien, T., Pollner, M.: Car-IT - The Revolution in Software Development (in German). ATZ elektronik, 06/2015
- (Hoffmann 2013) Hoffmann, D.: Software-Quality (in German). Springer Publ., 2013
- (Hülshorst et al. 2015) Hülshorst, T., Richenhagen, J., Richert, F., Nase, A.: New Dimensions in Automotive Software development (in German), ATZ elektronik, 2015
- (iX 2017) iX Special – Agil better to develop software (in German). iX Spezial 13/2017, Heise Publ., 2017
- (Johanning and Mildner 2015) Johanning, V., Mildner, R.: Car IT compact - Driving connected and autonomously (in German). Springer-Vieweg Publ., 2015
- (Jung and Kalmar 2015) Jung, C., Kalmar, R.: Re-interpret Data Security - the Data Gold and Business Models (in German). ATZ elektronik, 04/2015
- (Jindal et al. 2016) Jindal, A., Bhardwaj, A., Johny, L., Ramesh, S., Abhijay, V.: Ride Sharing – Software Requirements Specification, class paper, *Car IT and Cybersecurity class*, IIIT-B, 2016
- (Kaplan and Norton 1996) Kaplan, R. S., Norton, D. P.: The Balanced Scorecard - Translating Strategy into Action, Harvard Business Review Press, 1996
- (Kacher 2016) Kacher, G.: We are expecting a terribly beautiful auto world (in German). Sueddeutsche online. August 16th 2016. Available from: <http://www.sueddeutsche.de/auto/automobile-zukunft-uns-erwartet-eine-schrecklich-schoene-autowelt-1.3114841>
- (Kernighan and Pike 2006) Kernighan, B. W., Pike, R.: The Practice of Programming, Pearson Publ., 2006
- (Kroker 2015) Kroker, M.: App into the Factory: Now the Automakers Board the App Business (in German). Wirtschaftswoche online. July 1st 2015. Available from: <https://www.wiwo.de/unternehmen/it/app-in-die-fabrik-jetzt-enterprise-it-konzerne-das-app-geschaef/11932894.html>
- (Kroker 2016) Kroker, M.; How Frankfurt is Developing into the Cloud Center of Europe (in German). Wirtschaftswoche online. August 12th 2016. <https://www.wiwo.de/unternehmen/it/cloud-computing-frankfurt-erfüllt-wichtige-infrastruktur-voraussetzungen/14004492-2.html>
- (Lashinsky 2012) Lashinsky, A.: Inside Apple – How America's Most Admired – and Secretive – Company Really Works, Wiley-VCH Publ., 2013
- (Laudon et al. 2010) Laudon, K., Laudon, J., Dass, R.: Management Information Systems, Pearson Publ., 2010
- (Linzmayer 2004) Linzmayer, O. W.: Apple Confidential 2.0 – The definite History of the World's Most Colorful Company, No Starch Press, 2004
- (Lewis 1997) Lewis, T.: The friction free economy, Marketing strategies in a wired world: Strategies for success in a wired world. Harper Business Publ. 1997
- (Ludewig and Licher 2013) Ludewig, J., Licher, H.: Software Engineering - Basics, People, Processes, Techniques (in German). Dpunkt Publ., 2013
- (Mangiapane and Büchler 2015) Mangiapane, M., Büchler, R. P.: Modernes IT Management (in German). Springer Vieweg Publ., 2015
- (Masak 2010) Masak, D.: The Architecure Review (in German). Springer Publ., 2010
- (Meier 2012) Meier, R.: Professional Android 4 Application Development, Wrox Publ, 2012
- (Müller et al. 2015) Müller, G., Quathamer, G., Opel, C., Lauder, M.: Agile Methods in Software Development for HMI and Graphics (in German). autotechreview. 4(12), December 2015. Available from: https://autotechreview.com/media/attachments/32_35_atr_dec15.pdf
- (Randak 2016) Randak, S.: BMW, Daimler and VW cornered by Apple? Tesla? The danger for German automakers is lurking somewhere else (in German). Manager Magazin online.

- December 2nd 2016. Available from: <http://www.manager-magazin.de/unternehmen/artikel/autobauer-in-gefahr-zulieferer-haben-bessere-entwicklungskompetenz-a-1124068.html>
- (Rayle et al. 2014) Rayle, L., Shaheen, S., Chan, N., Dai, D., Cervero, R.: App-Based, On-Demand Ride Services: Comparing Taxi and Ridesourcing Trips and User Car - Characteristics in San Francisco. University of California Transportation Center Working Paper, August 2014. Available from: https://www.its.dot.gov/itspac/dec2014/ridesourcingwhitepaper_nov2014.pdf
- (Reddy et al. 2016) Reddy, B. S., Reddy V. V. A., Reddy T. H.: Ride sharing App – Application Manual, class paper, Car IT and Cybersecurity class, IIIT-B, 2016
- (Roth 2005) Roth, J.: Mobile Computing – Foundations, Technique, Concepts (in German). Dpunkt Publ., 2005
- (Sadun and Wardwell 2014) Sadun, E., Wardwell, R.: The Core iOS Developer's Cookbook – Essentials and Advanced recipes for iOS Programmers, Addison-Wesley Publ., 2014
- (Schäfer 2010) Schäfer, W.: Software Development - Introduction for the Most Demanding (in German). Addison-Wesley Publ., 2010
- (Silberschatz et al. 2012) Silberschatz, A., Galvin, P., Gagne, G.: Applied Operating System Concepts, Wiley Publ., 2012
- (Singh 2007) Singh, A.: Mac OS X Internals – A Systems Approach, Pearson Publ., 2007
- (Sloss et al. 2004) Sloss, A. N., Symes, D., Wright, C.: ARM System Developer's Guide - A Designing and Optimizing System Software, Elsevier Publ., 2004
- (Sommerville 2015) Sommerville, I.: Software Engineering, Addison-Wesley Publ., 10th edition, 2015
- (Steinacker 2016) Steinacker, L.: Code capital – The software code becomes a crucial factor (in German). Wirtschaftswoche online. September 11th 2016. Available from: <https://www.wiwo.de/my/technologie/digitale-welt/code-kapital-der-software-code-wird-zur-entscheidenden-groesse/14483036.html>
- (Stevenson 2010) Stevenson, S.: Cocoa and Objective-C Up and Running, O'Reilly, 2010
- (Stokes 2007) Stokes, J.: Inside the Machine - An Illustrated Introduction to Microprocessors and Computer Architecture, No Starch Press, 2007
- (Tanenbaum and Bos 2014) Tanenbaum, A., Bos, H.: Modern Operating Systems. Pearson Publ., 2014
- (Tanenbaum and Austin 2012) Tanenbaum, A., Austin, T.: Structured Computer Organization, Pearson Publ., 6th edition, 2012
- (Vembo 2016) Vembo, D.: Connected Cars – Architecture, Challenges and Way Forward. Whitepaper Sasken Communication Technologies Pvt. Ltd. 2016. Available from: https://www.sasken.com/sites/default/files/files/white_paper/Sasken-Whitepaper-Connected%20Cars%20Challenges.pdf
- (Weber 2012) Weber, R.: Technology of Enterprise Software (in German). Springer-Vieweg Publ., 2012
- (Werle 2015) Werle, K.: World in digital change – the game changer – BMW smartphone on wheels (in German). Manager Magazin online. November 23rd 2015. Available from: <http://www.manager-magazin.de/unternehmen/artikel/game-changer-bmw-sieger-in-wettbewerb-von-bain-und-mm-a-1063812.html>

Links

2014

- (URL1 2014) <http://www.kpmg-institutes.com/content/dam/kpmg-im/automotive/me-my-car-my-life.pdf>
- (URL2 2014) <https://www.linkedin.com/pulse/20140626152045-3625632-car-software-100m-lines-of-code-and-counting/>

2016

- (URL1 2016) <http://www.businessinsider.in/Theres-no-hope-of-anyone-catching-up-to-Android-and-iOS/articleshow/53815473.cms>
(URL2 2016) <https://www.smartface.io/>

2017

- (URL1 2017) <https://en.wikipedia.org/wiki/IOS>
(URL2 2017) <https://www.ralfebert.de/ios/ueberblick-ios-xcode/>
(URL3 2017) https://en.wikipedia.org/wiki/IOS_version_history
(URL4 2017) <https://www.idc.com/promo/smartphone-market-share/os>
(URL5 2017) [https://en.wikipedia.org/wiki/Android_\(operating_system\)](https://en.wikipedia.org/wiki/Android_(operating_system))
(URL6 2017) <https://developer.apple.com/xcode/features/>
(URL7 2017) <https://intellipaat.com/tutorial/ios-tutorial/ios-technologies/>
(URL8 2017) <https://developer.apple.com/>
(URL9 2017) <https://developer.android.com/training/index.html>
(URL10 2017) <http://www.vogella.com/tutorials/android.html>
(URL11 2017) <https://developer.android.com/guide/platform/>
(URL12 2017) <http://www.androidauthority.com/>
(URL13 2017) <https://yalantis.com/blog/uber-underlying-technologies-and-how-it-actually-works/>
(URL14 2017) <https://developer.omnis.net/blog/infographic-global-smartphone-sales-market-share-by-vendor-and-os>
(URL15 2017) <https://en.wikipedia.org/wiki/Xamarin>
(URL16 2017) <https://en.wikipedia.org/wiki/Mapbox>
(URL17 2017) https://en.wikipedia.org/wiki/Google_Maps
(URL18 2017) https://de.wikipedia.org/wiki/Lines_of_Code (in German)



Carsharing

8

This chapter discusses carsharing. Section 8.1 introduces the carsharing concept and the different variants of it as well as carsharing services offered so far. The focus in Sect. 8.2 is on the carsharing concept by Daimler, its car2go business model. In Sect. 8.3 the use cases in regard to the different phases of carsharing are analyzed, and the resulting requirements are discussed. Section 8.4 describes significant modifications to the hardware and software infrastructure of a smart car for using it in the carsharing business model. The focus of Sect. 8.4 is on connectivity which is realized through a GSM module that is embedded in the telematics unit. In Sect. 8.5 the impact of electric vehicles in carsharing applications is discussed. It also shows the block diagram of a standard electric vehicle. Section 8.6 refers to the carsharing activities by other OEMs and their brands. Since the whole use case of carsharing relies on the constant connectivity between the car and the backend system, the proper security of the vehicles is a major concern which can be realized by intrusion detection and prevention to avert vulnerabilities through cyberattacks. In this regard Sect. 8.7 introduces cyberattack surfaces and discusses the mitigation of cyberattacks (see also Chap. 6). Section 8.8 finally wraps up within a conclusion, while Sect. 8.9 contains a comprehensive set of questions on the carsharing business model, finally followed by references and suggestions for further reading.

8.1 The Carsharing Concept

Carsharing is a model of car rental where people rent cars for short periods of time, often by hour or even by minutes. This is an attractive option for customers who make only occasional use of a vehicle, as well as others who would like occasional access to a vehicle of a different type than they use everyday (Dämon 2013; Knieps 2016; Meyer and Shaheen 2017). The organization renting the cars may be a commercial business, or the users may be organized as a company, public agency, cooperative, or as an ad hoc grouping.

Carsharing services are available in over a thousand cities in many countries. Services include Autolib, City Car Club, Greenwheels, Stadtmobil, Zipcar, and others. Traditional car rental companies have introduced their own carsharing services, including Hertz on Demand, Enterprise CarShare by Enterprise Rent-A-Car, Avis on Location by Avis, U Car Share by U-Haul, and others. In addition, car manufacturers have also introduced their own carsharing services, including Daimler's car2go, BMW's DriveNow, and VW's Quicar, which was discontinued and relaunched under the new Moia mobility brand (Burt 2016; URL4 2016).

Carsharing is becoming more and more popular; the number of users grew rapidly from 2006 to 2014. In 2014 there were nearly 5 million carsharing users globally (URL6 2017). Of these, 800,000 were carsharing members in the USA. As of 2017, Zipcar, the largest in the USA, had more than a million members and offers >12,000 vehicles in 500 cities and towns throughout North America and Europe (URL7 2017).

In Germany, there are roughly 150 carsharing service providers, with a total of 1.7 million customers (URL4 2017). The biggest is Daimler's car2go, closely followed by BMW's DriveNow and Deutsche Bahn's Flinkster (URL4 2017). The city of Berlin alone has more than 10 carsharing service providers (URL4 2017).

There are three main variants of carsharing (URL12 2017; Reindl et al. 2016):

- *Traditional/station based:* This is the oldest variant. Cars should be reserved and dropped off at a particular location (station). The cost is calculated on an hourly base.
- *Flexible/free-floating carsharing:* There is a fixed operating regime/area where cars can be picked up and dropped off wherever one finds them in the city. A reservation is not necessary; if one comes across a car which is free, one can rent it on the spot. The cars will be distributed throughout the city based on the users drive pattern. Costs are typically calculated by the minute. The benefit is the high degree of flexibility; on the flip side, longer trips can be quite costly.
- *Private carsharing:* Here the individual car owners rent out their cars to others. The cars can be found on Internet portals like BlaBla Car (URL10 2017). This model is particularly interesting in rural areas. However, the insurance is an issue.

The benefits of carsharing are appealing to more and more especially younger users, and can be summarized as follows (URL4 2017):

- Access to new cars
- Efficient utilization and overall total cost of ownership (TCO) advantages compared to owning a car
- No cost for
 - Fuel
 - Maintenance, repair, and taxes
 - Purchasing a car
- No need for parking space

8.2 Example car2go

car2go is a subsidiary of Daimler AG which provides carsharing services in cities across Europe, North America and China offering primarily 2-seater cars from its smart subsidiary shown in Fig. 8.1, charging by the minute, with hourly and daily rates available, allowing for one-way rentals and offering customers pre-paid on-street parking (URL5 2017; URL3 2015). The service forgoes the typical centralized rental office, and cars are user accessed where parked, via a downloaded proprietary smartphone app or chip card which communicates with a RFID reader mounted behind the windscreen (Fig. 8.2). Daimler pioneered the car2go service in Ulm, Germany, in October 2008 developed, by one of its internal business innovation units' test-marketing the service exclusively with Daimler employees (URL3 2016).

As of now there are globally more than 2.4 million customers, >1.3 million in Europe and >670,000 users in Germany (URL3 2017). car2go currently operates in eight countries. There are 26 locations in Europe and the USA. In 2016, a new location in China was opened as a first step into the Chinese market. car2go operates a fleet of nearly 14,000 cars with more than 1300 out of these being pure electric cars (URL3 2017).

Berlin has the largest fleet closely followed by Vancouver with more than 1000 cars each (URL3 2017).

The car2go business model is similar in all markets, though rates vary by location. The company charges per minute rates, with discounted fixed rates for hourly and daily usage also available as illustrated in Fig. 8.1. The rates are all-inclusive and cover rental, gas, insurance, parking in authorized areas, and maintenance. In addition, there may be a low fixed annual fee. In most markets,



Fig. 8.1 car2go concept (Haas and Möller 2017)



Fig. 8.2 RFID reader behind the windshield

car2go vehicles can either park in specifically designated parking spots, or in standard parking areas, with a special permit from the local municipality. Access to the car is granted by two-factor authentication. Firstly, the driver has to use the smartphone app or put the membership card with the embedded RFID tag close to the windscreens RFID reader shown in Fig. 8.2. Then, a four-digit PIN code has to be keyed in which has been specified before upon which the glove compartment opens and the car key is released. The initial car2go prototype was developed by the Daimler subsidiary TSS. The requirements were documented with IBM's requirement engineering tool DOORS (URL16 2017) which allowed for an efficient tracking of changes, establishing a link between the requirements and the corresponding test cases. To develop the software system the team led by TSS followed Daimler's standard development processes, using object-oriented methodologies, a sophisticated tool chain and agile concepts wherever appropriate.

8.3 Use Cases and Requirement Analysis for Carsharing

This section describes the use cases of a generic carsharing service and gives an overview of the main components to operate such a service, like:

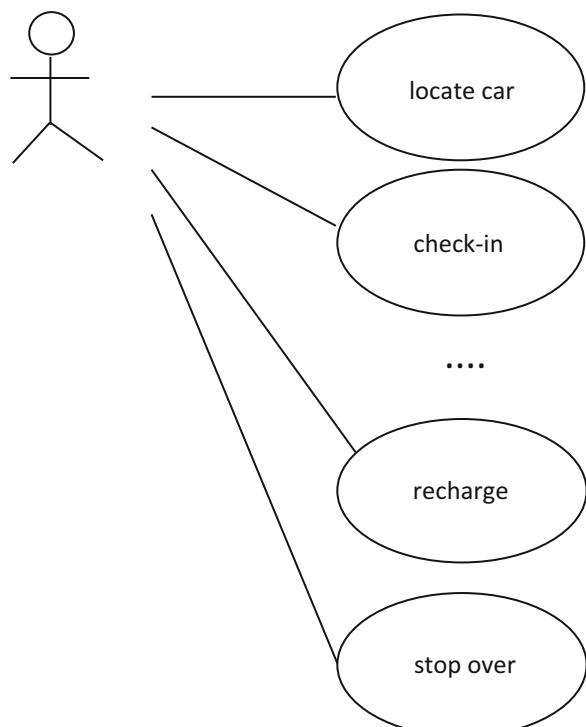
- In-car software running on a special head unit
- User App
- In-car frontend and infotainment system with a specific user interface
- Backend systems
- Portal for registration, billing information, user settings, and so forth

From a bird's eye perspective, the system consists of various physical and conceptual entities interacting with each other:

- *Shared Car*: Vehicle which is being shared by the users subscribing to the service.
- *Head unit*: Special head unit which implements the in-vehicle carsharing functions and communicates with the backend servers.
- *Customer*: Person who wants to use the service.
- *Customer Database*: Database of all the customers who are enrolled and registered for the service and are provided a membership card.
- *Wireless Carrier*: Agent who is responsible for all the communication that happens between the car and the backoffice, be it voice or data, including GPS.
- *RFID Reader*: Mounted, for example, against windscreen to read the RFID tag from the membership card. User will be authenticated, board computer starts up, door unlocks.
- *Smartphone*: Running the app to locate, reserve and open a car, communicate with the backoffice and handling various administrative functions
- *Backoffice*: Backbone of the shared car service, performing all the functions like booking, reserving a vehicle, billing, and managing the accounts of the members. It is also responsible for all the administrative tasks and handles customer complaints taking appropriate measures to deal with the issues.

Use cases can be modeled in UML as shown in Fig. 8.3 and specified in natural language like:

Fig. 8.3 Key use cases of a carsharing service



Check-in

- Door is openend via smartphone app or via holding the membership card close to the windscreen-mounted RFID reader.
- The driver is asked to key in a PIN code which was specified in the web portal.
- The vehicle board computer checks the PIN code with the server and acknowledges the PIN number.
- If the PIN code is correct, the driver can start the engine with the key he finds in the glove compartment.
- If the PIN code is wrong, the user has four trials before the system will block. In this case, the driver has to leave the car and check with the call center.
- As an option, voice/speech/fingerprint/face recognition can be used for biometric identification of the driver.

Driving

- The GSM module sends GPS position information, battery and fuel status, and other data to the base station at regular intervals.
- Driver uses the navigation module of the board computer to get guidance.
- Driver uses the infotainment system in the car for his needs during driving.

Stop-over

- User can stop on the way and park the car.
- Location and stopover status will be sent back to the base station.
- User shuts off systems, including board computer.
- After coming back to the car, the user again types in the PIN for authentication. Car door opens normally (with the key), and the car can be started. The board computer sends start information to the server.
- Optional: If the car is parked in a wrong place, the navigation system checks and warns the user.

Check-out

- The user can drop the car at any public parking place or a special reserved parking slot for shared vehicles. The navigation system guides the driver to specific shared car parking slots.
- The check-out procedure begins with a health and status assessment of the car similar to the check-in phase.

- In addition to the feedback on the car's condition while in the check-in phase, the driver should report problems that occurred during the driving phase.
- Location information is sent to the server. The driver puts the car key into the drawer and shuts the doors. The board computer shuts down systems and locks the doors. The head unit shuts down the systems and locks the doors.
In case a problem occurs or the car is in a poor condition but it is still possible to drive it, the driver can drop the car at a shared car contract garage receiving a voucher for future shared car rentals.

Health-check

- Driver has to check the condition of the vehicle outside and inside.
- A simple user-interface guides him through the checkup.
- After completing the check-up questions, the car can be started and used.
- If the condition of the car is poor, the user can make a free call to the call center and will be guided to the next available car.
- Optional: Camera is used to check interior conditions.
- Driver can make a statement about the problem which will be recorded by built-in voice recording functions.

Refueling

- User is notified once the fuel drops down to a fixed threshold and is guided to the nearest fuel station.
- User can refuel the car in filling stations with a voucher.
- The fuel tank information is sent to the server and optionally cross verified with voucher currency.
- Optional: Threshold is decided dynamically based on nearest fuel station and current fuel level.

Emergency

- If an accident happens, the car will send emergency data to the server w.r.t. location and vehicle data.
- If the GSM module is still working, the driver can make an emergency call.
- In case of a medical emergency, the driver can also make a call with the built-in GSM module. Location information is automatically sent to the server.
- Optional: Emergency button to log position and automatically call for help. Car will be picked up by service team or can be dropped off at a garage by the user.

8.4 Hardware/Software Modifications for Carsharing

The use cases described in the previous subsection require significant modifications to the E/E systems, especially to the head unit. Connectivity plays a fundamental role as many functions require a reliable communication and the car needs to be tracked carefully within the area of operation. Figure 8.4 shows the high-level block diagram of the system architecture of a typical shared car service. The major modification to the car is a special head unit which includes a GPS that constantly will send the geo location data to the central backoffice. The connectivity is realized through a GSM module that is embedded in the telematics unit. The system will send authentication data to the backend in the check-in phase, send geo location and status data during the driving phase, and communicate with the backend system in the drop-off phase. Should the customer choose a stopover, this will also be communicated. The backend system consists of modules for monitoring and tracking the car in an area of operation, billing, and handling customer data like address, PIN code and so forth.

Special care has to be taken in situations where the GSM signal is weak or the GPS signal is not available. In this case, the exact position of the car cannot be detected, and the driver has to find another location for the drop-off.

The in-car software functions of the shared car service require a powerful embedded platform. Microsoft offers a proven and stable solution, called Windows Embedded Automotive, as shown in Fig. 8.5. Many modern microcontrollers from vendors like Qualcomm/NXP/Freescale, Infineon, Intel, Samsung, Renesas offer the necessary performance to run the core system. Figure 8.6 shows the i.MX (NXP/FreeScale) microcontroller architecture, which has a lot of multimedia support and I/O interfaces on-board and is certified for Windows Embedded Automotive.

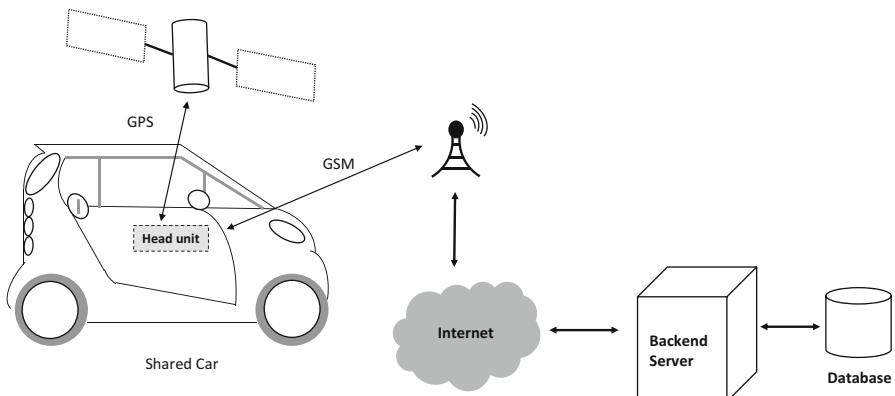


Fig. 8.4 Top-level architecture of a shared car service

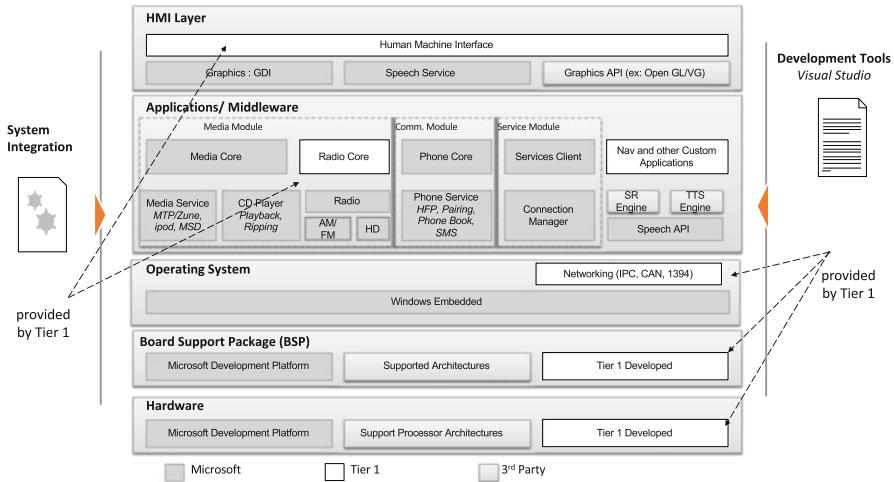


Fig. 8.5 Windows Embedded Automotive Stack (source: Microsoft, modified)

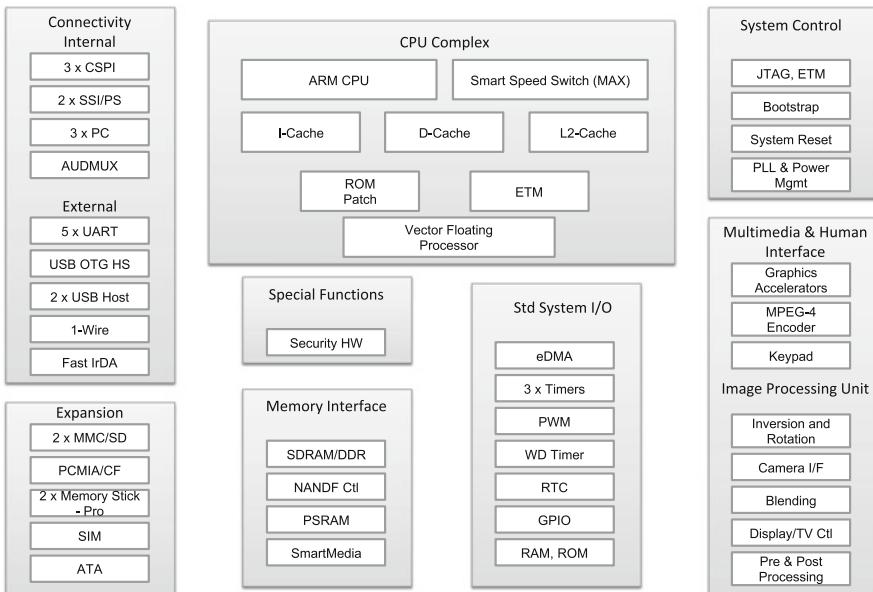


Fig. 8.6 i.MX microcontroller architecture (source: NXP, (URL15 2017))

8.5 Electric Vehicles and Carsharing

Electric cars still phase various hurdles for a wide acceptance (Reimann 2014; Hinderer et al. 2016; Loogen 2015). The main problems are the high price, primarily driven by the battery cost, the lack of a widespread charging infrastructure, the

shorter ranges compared to internal combustion engine (ICE)-based cars, and the longer time for re-charging compared to the short time needed for refueling. On the other hand, electric cars are emission free, although the total carbon footprint is a different story, they typically have lesser wear and tear, they enjoy tax benefits and subsidies, and they are fun to drive. Still, it will take time to overcome the main issues, and the penetration of electric cars will be single digit percentages over the next years to come.

Car manufacturers, nevertheless, see carsharing as an attractive mechanism to popularize electric cars giving drivers a first taste of the benefits of this technology (URL1 2017). As shared cars operate in smaller regions, the range is not such a big issue, and if cities provide the charging infrastructure, both sides can benefit, for example, by emission-free traffic in inner city areas.

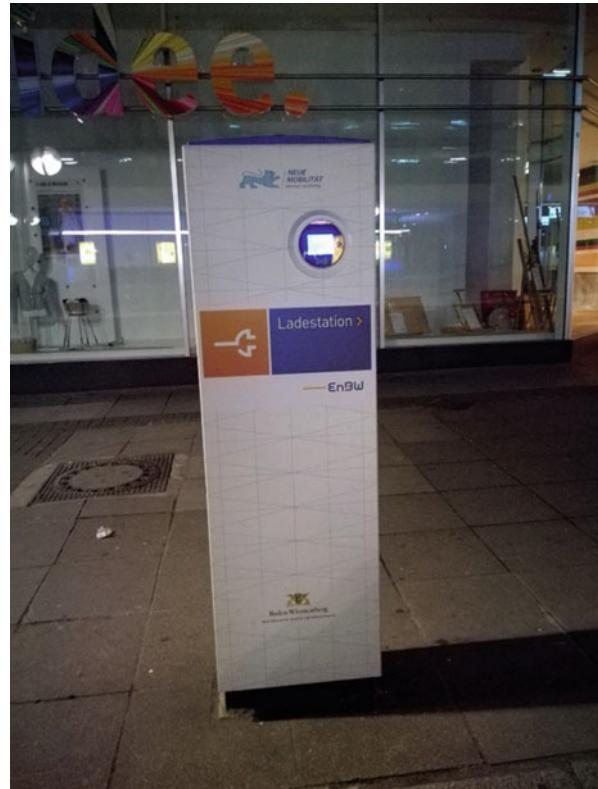
Daimler operates a fleet of electric Smarts in Stuttgart as shown in Fig. 8.7. Other cities with predominately electric fleets are Amsterdam and Madrid (URL3 2017). Recently also the electric version of the B class was added to the fleet of shared cars in Stuttgart (URL2 2017).

The city of Stuttgart provides a network of charging station as shown in Fig. 8.8, together with EnBW. The charging station offers quick charging functionality. The billing is done transparently, the car2go driver just has to put the plug into the socket. All communication between the car and electric utility company is being handled automatically based on a specific machine-to-machine (M2M) protocol. If one puts the electric version of the car2go smart for charging, one gets some credit points that can be redeemed against the next rental cycle. This is similar to the incentive scheme for refueling an internal combustion engine powertrain (ICE)-based car2go car.



Fig. 8.7 An eletric car2go smart in Stuttgart

Fig. 8.8 Battery charging station in Stuttgart



In Fig. 8.9 the block diagram of an electric car is shown (Hinderer et al. 2016; Stellet et al. 2014). One typically differentiates between different combinations of internal combustion engine and electric motors like, serial and parallel hybrids, plug-in hybrid, and full electric cars. The Smart cars, which are being operated in Stuttgart, are fully electric with no ICE-backup. The range is typically between 100 and 140 km depending on the style of driving, ambient temperature, and various other factors. The largest block is the battery pack. It is a lithium-ion technology with a capacity of 14 kWh. The battery pack is controlled by a sophisticated battery monitoring system that constantly checks the state of charge and health of the battery and regulates the current which is fed into the driveline. The electric motor typically is a brushless, asynchronous machine that is controlled by power electronics.

Electric cars contain a sizeable amount of software that supervises and controls the different components of the electric powertrain (Almeida et al. 2017; Kampker et al. 2013; Hinderer et al. 2016). Figure 8.10 shows the battery management system which controls charging, monitors the battery health and carefully balances the charge of the different Li-ion cells (URL13 2017; URL14 2017). An efficient

power management is essential as this directly affects the range. The Smart e-car also uses sophisticated recuperation technologies to feedback electric energy to the battery.

Thermal loads have to be handled carefully, and the overall management of this is another area where sophisticated control technologies are being deployed.

DC-DC converters step down the high voltage used in the drivetrain to the voltage levels of the other electric components as shown in Fig. 8.9.

It is interesting to compare the cost structure of a conventional and an electric car which is shown in Figs. 8.11 and 8.12 (Hinderer et al. 2016; Kampker et al. 2013). The battery dominates the cost structure, while the drivetrain becomes much simpler with typically lesser wear and tear. This has dramatic consequences for the value chain in a predominantly ICE-based industry (Sorge 2016a, b).

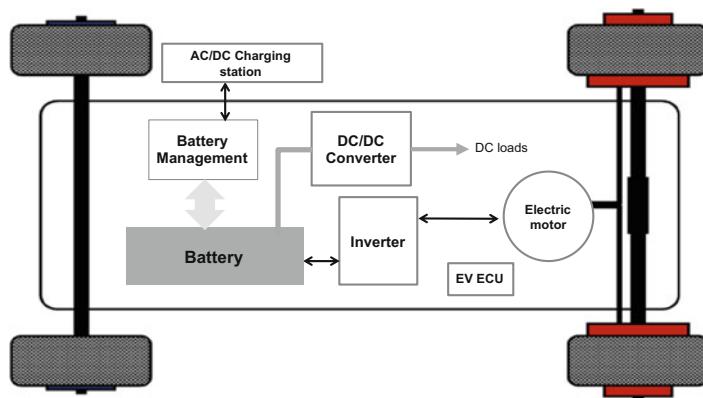


Fig. 8.9 Architecture of an electric drivetrain

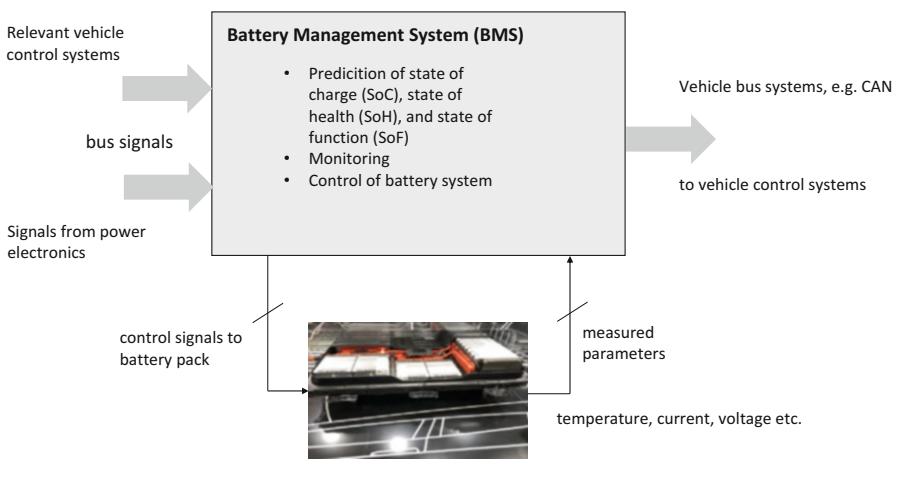


Fig. 8.10 Architecture of a battery management system (see URL13 2017)

Fig. 8.11 Typical cost structure of a conventional vehicle

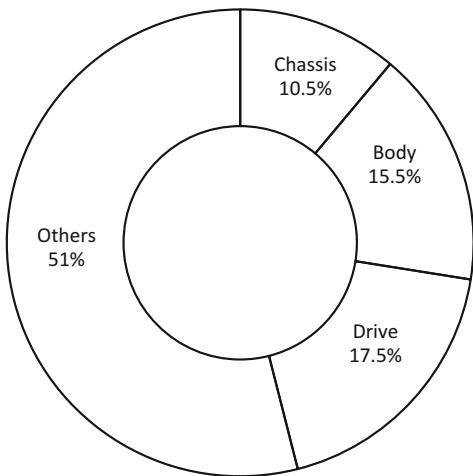
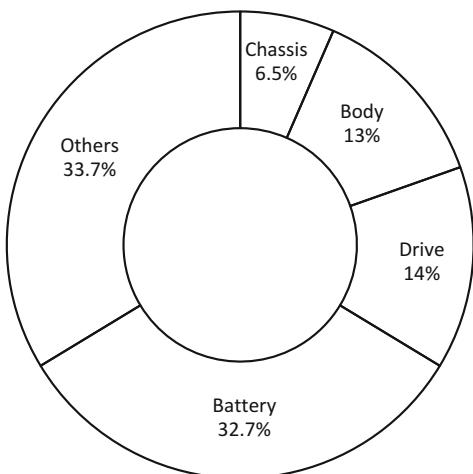


Fig. 8.12 Typical cost structure of an electric vehicle



8.6 Carsharing Activities by Other OEMs

BMW has launched a similar concept like Daimler under the brand name DriveNow, as shown in Fig. 8.13. The fleet consists of Mini cars with similar modifications to the car than for car2go.

More than 850 electric BMW i3 are available for experiencing electric mobility. DriveNow customers already have used these vehicles for 1.4 million electric car trips (URL1 2017).

Thirty thousand DriveNow customers even use electric cars exclusively (URL1 2017). Berlin tops the list of cities with electric carsharing (URL1 2017).

BMW's carsharing solution is based on the Android platform. The use cases and features are very similar to the offerings of Daimler. Daimler and Bosch will launch an automated valet parking feature for car2go in 2018 (URL3 2015; URL17 2017). Clearly, there is a tough competition in the market.

VW has started several initiatives (Burt 2016; URL4 2016). The group invested into Gett (URL11 2017) and is launching a carsharing service soon (Burt 2016). Also, Chinese players are moving into the market (see Fig. 8.14).



Fig. 8.13 BMW's carsharing business is called DriveNow



Fig. 8.14 Chinese carsharing companies are going global

8.7 Cyber Attack Surfaces and Mitigation of Cyber Attacks

The whole concept of carsharing relies on the constant connectivity between car and backend system and the proper authentication and authorization of the car driver. Connectivity ensures the steady tracking of the car's position and state of health, while the user has to be authenticated to avoid theft and to organize the billing. Connectivity, however, creates potential attack surfaces that could be exploited by cyber criminals (Greenberg 2013; Currie 2015; Vembo 2016; Serio and Wollschläger 2015; Schöttle 2015a; URL4 2015; URL5 2015; URL1 2017). In the case of an electric car, the communication between car and charging infrastructure is another attack vector that has to be taken care of (Haas and Möller 2017). Also, sensitive data about the health of the battery has to be monitored.

Cybercrime is a serious threat and gets a lot of attention recently (McMillan 2011; Berke 2015; Miller and Valasek 2015; URL1 2016; URL2 2016; URL9 2017).

Let us first look at the authentication mechanism between car and driver (Wolf et al. 2016). The driver basically has two options to authenticate and open the car. In the first scenario, a mobile phone is used to open the door. This requires a communication from phone to backend and backend to car. The second option uses the RFID tag embedded in a membership card as shown in Fig. 8.2. The tag communicates with the card reader installed behind the window screen and opens the door if the tag is authenticated. Opening the door is based on a signal which is sent to the door ECU. This signal could be compromised by manipulation of the wiring harness.

In both cases, using the smartphone or the member card with RFID tag, a second authentication mechanism (two-factor authentication) requires the driver to prompt in a 4-digit number. The correct PIN will release the car key and the car can be started like any other Smart car.

There are multiple other attack scenarios (Haas and Möller 2017; Stockburger 2016). Some of these are:

- Attack on the GSM communication between head unit and backend server.
- Attack on the smart card for filling the tank.
- DDoS attacks on the backend servers: DDoS is a distributed denial-of-service attack in which multiple compromised computer systems attack a target, such as a server, website, or other network resource and cause a DoS for users of the targeted resource. The flood of incoming messages, connection requests, or malformed packets to the target system forces it to slow down or even crash and shut down, thereby denying service to legitimate users or systems.
- Hacking user accounts to get access to personal data.
- Over-the-air update (OTA) attacks.
- Spoofing of GPS signals to redirect the car.

Also, with the introduction of a new valet parking functionality (URL3 2015; URL2 105), the potential attack surfaces have become larger. Valet parking will be discussed in detail in Chap. 10. This feature is particularly interesting for shared cars as finding a parking space is often difficult, especially, if also a charging infrastructure is needed. To make things easier, Bosch and Moovel, the parent company of car2go, collaborate to offer an automatic valet parking function in 2018 (URL3 2015). This will allow a car2go driver to just hop out of the car and let the car park itself (Rees 2016).

Tesla has a very open policy to address security risks (Pickhard et al. 2015; Schöttle 2015b; Brisbourne 2014). Vulnerabilities will be published openly and dealt with transparently and rapidly (Zetter 2015). That way, zero-day exploits will not be open for long, and the joint effort of customers and hackers can help to prevent attacks.

8.8 Conclusion

Carsharing is getting more and more popular. All of the large automotive OEMs have special offerings for carsharing. Daimler was one of the pioneers with the car2go concept which first went into trial phase in Ulm in 2011.

Carsharing differs from the standard rental model in various ways. The user typically does not have to reserve the car upfront, the booking can be done ad hoc and the tariffs are flexible in terms of mileage and duration. Rental car companies generally only offer full day rates and do not allow for hourly or minute-based usage. Also, shared cars can typically be dropped off at any public parking space within the area of operation. These areas are restricted to a city or

metropolitan region, long haul distances are not covered. The goal is to make it as easy as possible to move from one point to another within a limited geography.

Carsharing requires some key modifications to the car. A telematics box is required which constantly sends GPS data to the backend. Also, an authentication device is needed which allows access to the car by opening the door for authorized users only.

Some OEMs also use carsharing as a means to familiarize their customers with electro mobility. car2go, for example, operates a fleet of electric Smarts in Stuttgart. The city offers a network of charging stations.

Electric cars expose a broad attack surface, as a sophisticated M2M communication is required between the car and infrastructure, particularly the charging stations. We looked at the key issues and referred to further literature on the cybersecurity of electric cars.

car2go and DriveNow will merge shortly, after long discussions how best to integrate their fleets ([URL5 2016](#)). In 2015, Flinkster and car2go were in serious discussions to combine their business ([URL6 2015](#)). However, IT issues turned out to be a serious hurdle.

In carsharing the need for constant connectivity and the remote access features are potential attack vectors for cyber criminals. We briefly discussed some of the issues. Strong authentication mechanisms and intrusion detection systems can help to prevent cyberattacks and to increase overall system security ([URL4 2015](#)).

8.9 Exercises

What is meant by the term *new mobility trends*?

Describe some key aspects of the new mobility trends.

What is meant by the term *carsharing*?

Describe some key aspects of carsharing.

How does carsharing differ from ridesharing?

Describe some key aspects of the differences.

What is meant by the term *car2go*?

Describe some key aspects of car2go.

What is meant by *carsharing initiatives by other OEMs*?

Name a few other carsharing initiatives by other OEMs.

How does *carsharing differ from car rental*?

Describe some key aspects of the differences.

How does a *user authenticate himself/herself to the car*?

Describe some key aspects methods.

How will one *know if the driver has a driver's license*?

Describe some key aspects.

How is the *refueling being handled for shared cars*?

Describe some key aspects.

Why is it important to *drop off the car in an operating area*?

Describe some key aspects.

What happens if there is *no GPS connection available when car is dropped off?*

Describe some key aspects.

What happens if *one has an accident?*

Describe some key aspects.

What are the *cost for carsharing?*

Describe some key aspects.

How will *maintenance and repair be handled?*

Describe some key aspects.

How do the *carsharing offerings of OEMs differ?*

Describe some key differences.

What are the *main components of an electric car?*

Describe the key ones.

What are the main drivers for the *cost structure of an electric car?*

Describe the cost structure of an electric car.

How is the *charging phase being organized?*

Describe the key aspects.

What are *typical ranges of electric cars?*

Describe the key ones.

How does the *driving experience differ between electric and combustion engine cars?*

Describe some key differences.

What are the *success factors for carsharing in cities?*

Describe the key aspects.

What *data is being exchanged by the car and the backend system?*

Describe the key aspects.

What *connectivity solutions are typical being used in carsharing?*

Describe the key aspects.

Describe the *attack surfaces of a shared car?*

Describe the key aspects.

What are *critical attack vectors?*

Describe the key aspects.

What *authentication mechanisms are being used?*

Describe the key aspects.

What *other concepts could you think of?*

Describe the other concepts.

What means of *cybersecurity would you suggest to fight of cyberattacks?*

Describe your suggestion in detail.

Why is *carsharing not so popular in India?*

Describe the key reasons.

What *market potential for carsharing do you see in US cities?*

Describe the key aspects.

What ideas do you have to *reduce maintenance and repair costs?*

Describe your ideas in detail.

How do you insure that a damage is being reported?

Describe reporting in detail.

How can you *be sure who caused the damage?*

Describe the key aspects.

What role does *the availability of parking space play in the success of a carsharing concept?*

Describe the key aspects.

What areas are particular *important for parking?*

Describe the key aspects.

What *benefits do Daimler and BMW see in merging their activities?*

Describe the key aspects.

Why has it been difficult to *integrate car2go and Flinkster?*

Describe the key aspects.

What benefits do you see in a *cooperation by car2go and Flinkster have?*

Describe the key aspects.

References and Further Reading

- (Almeida et al. 2017) Almeida, F., Silva, P., Leite, J.: Proposal of a Carsharing System to Improve Urban Mobility. In: Theoretical and Empirical Researches in Urban Management, Vol. 12, Issue 3, pp. 32–44, 2017
- (Berke 2015) Berke, J.: When Cyberattacks lead to Bankruptcy (in German). Wirtschaftswoche online. November 25th 2015. Available from: <https://www.wiwo.de/unternehmen/it/hackerangriffe-auf-unternehmen-wenn-cyberattacken-in-den-bankrott-fuehren/12632916.html>
- (Burt 2016) Burt, M.: Volkswagen unveils Moia, its new mobility services brand. Autocar online. December 5th 2016. Available from: <https://www.autocar.co.uk/car-news/industry/volkswagen-unveils-moia-its-new-mobility-services-brand>
- (Brisbourne 2014) Brisbourne, A.: Tesla's Over-the-Air Fix: Best Example Yet of the Internet of Things? Wired online. February 2014. Available from: <http://www.wired.com/insights/2014/02/teslas-air-fix-best-example-yet-internet-things/>
- (Currie 2015) Currie, R.: Developments in Car Hacking. SANS Institute. December 5th 2015. Available from: <https://www.sans.org/reading-room/whitepapers/internet/developments-car-hacking-36607>
- (Dämon 2013) Dämon, K.: Corporate Carsharing – Companies want to get away from the company car. Wirtschaftswoche online. April 19th 2013. Available from: https://www.wiwo.de/unternehmen/auto/corporate-carsharing-unternehmen-wollen-weg-vom-dienstwagen/v_detail_tab_print/8081522.html
- (Greenberg 2013) Greenberg, A.: Hackers reveal nasty new car attacks-with me behind the wheel. Forbes online. July 24th 2013. Available from: <https://www.forbes.com/sites/andygreenberg/2013/07/24/hackers-reveal-nasty-new-car-attacks-with-me-behind-the-wheel-video/#64771b28228c>
- (Haas and Möller 2017) Haas, R. E., Möller, D. P. F.: Automotive Connectivity, Cyber Attack scenarios and Automotive Cyber Security. In Proceedings IEEE/EIT Conference, Lincoln, NE, 2017
- (Hinderer et al. 2016) Hinderer, H., Pflugfelder, T., Kehler, F. (Eds): Electromobility – Opportunities for suppliers and manufacturers (in German). Springer Automotive Media, 2016
- (Kampker et al. 2013) Kampker, A., Vallee, D., Schnettler, A.: Electromobility – The Basis of a Future Technology (in German). Springer/Vieweg Publ., 2013
- (Knieps 2016) Knieps, S.: Humans will always want to drive themselves - Daimler Board member Entemann about the future of the community car and how the autonomous car will change the business model of car2go (in German)

- (Kuchler and Strzelczyk 2013) Kuchler, G., Strzelczyk, M.: Tire Pressure Monitoring – Safety Aspects and Value-added Functions (in German). ATZ elektronik, 02/2013
- (La Vinh and Cavalli 2014) La Vinh, H., Cavalli, A. R.: Security attacks and solutions in vehicular ad hoc networks: a survey. International Journal on AdHoc Networking Systems (IJANS), Vol. 4, No. 16, pp. 1–20, 2014
- (Lampart and Bähren 2015) Lampart, O., Bähren, H.: Hardware- and Software-Security Solutions for Infotainment Systems (in German), ATZ elektronik, 03/2015
- (Loogen 2015) Loogen, F.: Moment of Electromobility (in German). ATZ elektronik, 03/2015
- (McMillan 2011) McMillan, R.: With Hacking, Music Can Take Control of Your Car. PCWorld online. March 11th 2011. Available from: https://www.pcworld.idg.com.au/article/379477/hacking_music_can_take_control_your_car/
- (Meyer and Shaheen 2017) Meyer, G., Shaheen, S.: Disrupting Mobility – Impacts of Sharing Economy and Innovative Transportation on Cities. Springer Publ. 2017
- (Miller and Valasek 2015) Miller, C., Valasek, C.: Remote exploitation of an unaltered passenger vehicle. August 10th 2015. Available from: <http://illmatics.com/Remote%20Car%20Hacking.pdf>
- (Pickhard et al. 2015) Pickhard, F., Emele, M., Burton, S., Wollinger, T.: New Thinking for Secure Connected Vehicles (in German). ATZ elektronik, special edition, 7/2015
- (Reimann 2014) Reimann, W.: Electrified into the future (in German). ATZ elektronik, 06/2014
- (Reindl et al. 2016) Reindl, S., Klümper, M., Maier, B.: Mobility Services in the Automotive Industry (in German). In Dietz, W., Reindl, S., Bracht, H.: Principles of the Automotive Management (in German), Springer Automotive Media, 2016
- (Rees 2016) Rees, J.: Mobility – Never have to park yourself (in German). Wiwo online. May 6th 2016. Available from: <https://www.wiwo.de/technologie/mobilitaet/mobilitaet-nie-mehr-selber-einparken-muessen/13529696.html>
- (Schlesiger 2016) Schlesiger, C.: Flixbus – The creepy triumphal march of the startup. Wirtschaftswoche online. October 17th 2016. Available from: <https://www.wiwo.de/unternehmen/dienstleister/flixbus-der-unheimliche-siegeszug-des-start-ups/14680498.html>
- (Schöttle 2015a) Schöttle, M.: Security for Software and IT (in German). ATZ elektronik, 03/2015
- (Schöttle 2015b) Schöttle, M.: Hacker instead of Cracker (in German), ATZ elektronik, 04/2015
- (Serio and Wollschläger 2015) Serio, G., Wollschläger, D.: Networked Automotive Defense Strategies in the Fight against Cyberattacks (in German). ATZ elektronik, 06/2015
- (Stockburger 2016) Stockburger, C.: IT security of cars: You have no choice but to trust the manufacturers (in German). Spiegel online. November 1st 2016. Available from: <http://www.spiegel.de/auto/aktuell/hacker-angriffe-man-hatkeine-andere-wahl-als-den-autoherstellern-zutrauen-a-1092224.html>
- (Sorge 2016a) Sorge, N.V.: Electric Cars threaten the creation of value in Germany, The Explosive Billionpoker around the Battery Manufacturers (in German). Manager Magazin online. October 26th 2016. Available from: <http://www.manager-magazin.de/unternehmen/autoindustrie/batteriefabriken-das-milliardenspiel-der-autokonzerne-a-1118054-druck.html>
- (Sorge 2016b) Sorge, N.-V.: Warren Buffett's Electric Car Chinese – BYD is attacking Daimler with its own factory in Europe (in German). Manager Magazin online. October 13th 2016. Available from: <http://www.manager-magazin.de/unternehmen/autoindustrie/byd-warren-buffetts-elektroauto-beteiligung-greift-an-a-1116320.html>
- (Stellet et al. 2014) Stellet, J., Gießler, M., Gauterin, F., Puente León, F.: Model-based Traction Control for Electric Vehicles (in German). ATZ elektronik, 02/2014
- (Vembo 2016) Vembo, D.: Connected Cars – Architecture, Challenges and Way Forward. Whitepaper Sasken Communication Technologies Pvt. Ltd. 2016. Available from: <https://www.sasken.com/insights/white-papers/connected-cars---architecture-challenges-and-way-forward-0>
- (Wolf et al. 2016) Wolf, A., Greiff, S., Obermaier, R.: Vehicle access systems of tomorrow (in German). ATZ elektronik, 03/2016

(Zetter 2015) Zetter, K.: Researchers Hacked A Model S, But Tesla's Already Released A Patch. Wired online. August 6th 2015. Available from: <https://www.wired.com/2015/08/researchers-hacked-model-s-teslas-already/>

Links

2015

- (URL1 2015) Intel Security White Paper Automotive Security Best Practice. Intel/Mcafee. June 2016. Available from: <https://www.mcafee.com/enterprise/en-us/assets/white-papers/wp-automotive-security.pdf>
- (URL2 2015) https://www.wiwo.de/unternehmen/auto/digitalisierung-der-autoindustrie-kuenftig-braucht-man-das-lenkrad-nicht-mehr/v_detail_tab_print/11602152.html
- (URL3 2015) <http://telematicsnews.info/2015/15/daimler-bosch-and-car2go-develop-automatic-parking/>
- (URL4 2015) <https://www.symantec.com/content/dam/symantec/docs/white-papers/building-comprehensive-security-into-cars-en.pdf>
- (URL5 2015) https://www.markey.senate.gov/imo/media/doc/2015-02-06_MarkeyReport-Tracking_Hacking_CarSecurity%202.pdf
- (URL6 2015) <http://www.carsharing-news.de/flinkster-und-car2go vereint/>

2016

- (URL1 2016) <http://www.faz.net/aktuell/wirtschaft/macht-im-internet/verbrechen-4-0-crime-as-a-service-14510568.html>
- (URL2 2016) <https://www.wiwo.de/unternehmen/industrie/cyber-angriffe-auf-die-industrie-jedes-zweite-unternehmen-hat-keinen-notfallplan-/13496740.html>
- (URL3 2016) <http://www.faz.net/aktuell/wirtschaft/daimler-baut-konzern-fuer-die-digitalisierung-um-14424858.html>
- (URL4 2016) https://www.wiwo.de/unternehmen/auto/volkswagen-neue-mobilitaetsdienste-erhalten-eigene-marke/v_detail_tab_print/14616692.html
- (URL5 2016) <http://www.bmwblog.com/2016/12/15/bmw-daimler-considering-merging-drivenow-with-car2go/>

2017

- (URL1 2017) <http://www.carsharing-news.de/drivenow-elektrofahrten-2017/>
- (URL2 2017) <http://www.carsharing-news.de/elektro-b-klasse-car2go-stuttgart/>
- (URL3 2017) https://www.car2go.com/media/data/germany/microsite-press/files/fact-sheet-car2go_mai-2017_de.pdf
- (URL4 2017) <http://www.carsharing-news.de/carsharing-anbieter/>
- (URL5 2017) <https://www.car2go.com/DE/de/>
- (URL6 2017) <https://de.statista.com/statistik/daten/studie/388022/umfrage/anzahl-der-weltweiten-carsharing-nutzer/>
- (URL7 2017) <http://www.zipcar.com/press/overview>
- (URL8 2017) <http://www.manager-magazin.de/unternehmen/it/google-roboterwagen-fahren-aus-angst-vor-hackern-ohne-internet-a-1129346.html>
- (URL9 2017) <https://argus-sec.com/car-hacking/>

- (URL10 2017) <http://blablacar.com>
- (URL11 2017) <http://gett.com>
- (URL12 2017) <https://de.wikipedia.org/wiki/Carsharing>
- (URL13 2017) https://www.iisb.fraunhofer.de/en/research_areas/energy_electronics/stationary_battery_systems/example_of_developments/battery_management_systems_bms.html
- (URL14 2017) https://www.researchgate.net/publication/283796319_Smart_Battery_Cell_Monitoring_with_Contactless_Data_Transmission/figures?lo=1
- (URL15 2017) <https://www.nxp.com/products/processors-and-microcontrollers/applications-processors/i.mx-applications-processors/i.mx-mature-processors/applications-processors-integrated-image-processing-unit-ipu-connectivity-arm11-core:i.MX31>
- (URL16 2017) <https://www.ibm.com/us-en/marketplace/rational-doors>
- (URL17 2017) <http://www.bosch-presse.de/pressportal/de/de/bosch-und-daimler-zeigen-fahrerloses-parken-im-realnen-verkehr-116096.html>



Car Hailing and Ridesharing

9

This chapter is about car hailing and ridesharing. Section 9.1 introduces car hailing and ridesharing as a promising approach for reducing own car usage in a city cutting down the need for parking spaces, reducing traffic jams, and for the city as a whole helping to reduce pollution. Section 9.2 discusses online transportation network companies offering cab services/car hailing and ride-hailing which provide cab services through their respective apps for smartphones (see also Chap. 7). Section 9.3 focuses on the metropolitan area of Bangalore as an example of ridehailing and ridesharing operations with regard to cab types and prices as well as services offered. Section 9.4 describes surge pricing mechanisms taking into account peak hours where cab aggregators charge two or three times more. In Sect. 9.5 the problem of safety and initiatives to prevent crime and increase the safety both for the customers as well as for the drivers. In this regard, Sect. 9.5.3 refers to crime incidents in ridesharing, while Sect. 9.5.4 discusses government policies for ridesharing companies and Sect. 9.5.5 presents important legal cases. Section 9.6 refers to fraud, cyberattacks, and cybersecurity in ridesharing (see also Chap. 6). Section 9.7 finally wraps up within a conclusion, while Sect. 9.8 contains a comprehensive set of questions on the carsharing business model, and the last Sect. 9.9 includes references and suggestions for further reading.

9.1 Introduction

In recent years, an innovative rideshare service offering known as real-time ridesharing, dynamic ridesharing, or technology-enabled ridesharing has emerged (Amey 2010). Traditionally, ridesharing arrangements between two or more unrelated individuals for commuting purposes have been relatively inflexible, long-term arrangements based on fixed departure time schedules and driving responsibilities. The complexity of work and social schedules and the increase in vehicle trip

complexity, such as trip chaining, has made this type of commuting arrangement less desirable (Amey 2010).

Real-time ridesharing, also known as instant ridesharing, dynamic ridesharing, ad hoc ridesharing, on-demand ridesharing, and dynamic carpooling, is a service that arranges one-time shared rides on very short notice. This type of carpooling generally makes use of three recent technological advances (URL1 2017):

- GPS navigation to determine a driver's route and to arrange the shared ride
- [Smartphone](#) for travelers to request a ride from wherever they happen to be
- Social networks to establish trust and accountability between drivers and passengers

These services are coordinated through a network service, which can instantaneously handle the driver payments and match rides using an optimization algorithm (URL1 2017). Thus, ridesharing is a promising approach for reducing car usage in a city and cutting down the needed parking spaces as the cars are in motion and need not wait for the driver to return. This reduces traffic jams and, for the city as a whole, has a positive effect on air pollution (Bay 2016). In recent years, a plethora of web- and smartphone-based solutions have emerged for facilitating ridesharing. Famous examples are Uber, Didi, Lyft, and Ola.

Hence, this chapter gives an overview of the respective business models, pricing, and services offered and takes a close look at safety concerns and cybersecurity. Safety concerns arise with the inherent problem of matching two unknown parties—drivers and customers—for a service which is based on trust. In the conventional taxi business, customers rely on the checks and balances of authorities, taxi operators, and the peer pressure in the industry to insure the quality and safety of the transportation service. In the fast-growing ride-hailing economy, one has to rely on automatic and often remote checks and the honest feedback of former customers.

The competition between ride-hailing companies is tough as there is a lot of money involved as illustrated in Fig. 9.1 (Eckl-Dorna 2016; Freitag et al. 2015; Schultz 2016). In this regard, the battle between the two Chinese cousins, one leading Uber China and the other Didi, has become legendary (Hirn 2016). But, also in India, the competition is intense where the global giant Uber faces the local competitor Ola (Kashyap 2016).

In a new, growing market like the ride-hailing business with enormous potential and market evaluation, it is no surprise that this economy suffers from unfair market practices and cybercrime because the business is vulnerable to cyberattacks as everything depends on the proper functioning of the telco network connection and the smartphone app. Some of the major issues are:

- Cyberattacks on smartphones
- Denial-of-service (DoS) attacks with bot nets
- Fake accounts
- Fraud
- Identity theft

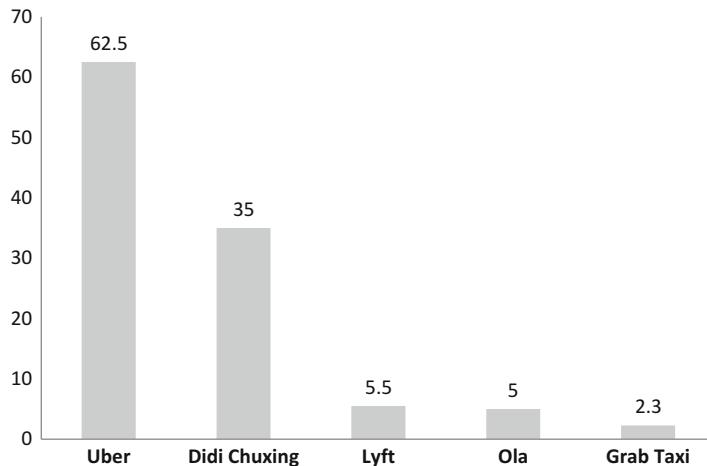


Fig. 9.1 Market evaluation of leading ride-hailing companies in bn USD (URL10 2017)

Therefore, this chapter also introduces some attack scenarios and discusses measures to counteract them, especially intrusion detection and big data analytics that can be deployed to detect anomalies in the stream of transactional data (Alheeti et al. 2015; Currie 2015). Ride-hailing companies like Uber are already planning the next steps, deploying autonomous cars (URL16 2016). Google is partnering with car manufacturers and taxi companies to help steer regulations and make driverless cars a reality (Spehr 2016; URL11 2016). This, of course, will need an even more stringent cybersecurity approach, a fact, Google is acutely aware of. This is one reason why Waymo cars take special precaution regarding Internet connectivity. Another clear indication how important cybersecurity has become for a proper functioning of these services is the hiring of Chris Valasek and Charlie Miller by Uber (Isaac and Perlroth 2015 and URL1 2015).

9.2 Ride-Hailing Companies and Taxi Aggregators

Uber, Didi, Lyft, Ola, and Gett are online transportation network companies offering cab services, car hailing, and ride-hailing (Fig. 9.2). The companies provide cab services through their respective apps for smartphones.

Uber Technologies Inc., commonly known as Uber, is an American multinational cab service. It was founded in March 2009 by Garrett Camp and Travis Kalanick in San Francisco, USA. Its services are currently available in over 60 countries and 444 cities (Vikas 2016; URL2 2017).

Figure 9.3 shows an Uber advertisement in the baggage claim section of Delhi airport. Uber cars have a special parking area at the airport and compete directly with radio taxis that also wait for customers.

Fig. 9.2 Logos of some of the leading ride-hailing and ridesharing companies

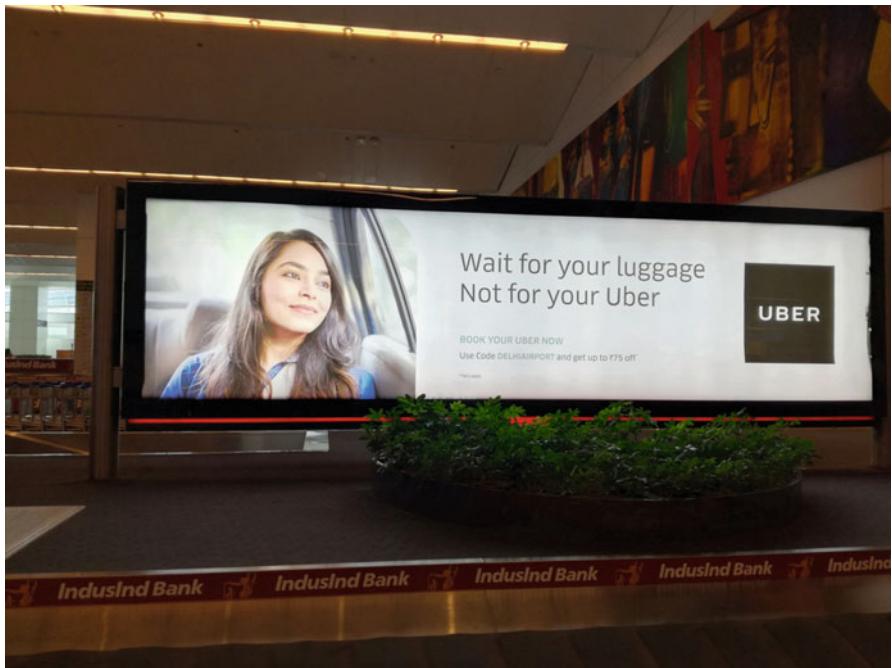


Fig. 9.3 Uber advertisement at Delhi airport

Didi Chuxing is the leading ride-hailing company in China. Didi was founded in 2012 by Cheng Wei, and later merged with its competitor Kuadi Dache, backed by the two largest Chinese Internet companies, Tencent and Alibaba. The business model is very similar to Uber. The growth since 2012 is mind-boggling.

In 2015, Uber completed its 1 billionth ride, which was still below the 1.4 billion rides completed by [Didi Chuxing](#) at that time. In October 2016, it was reported that 40 million riders used the service in a single month and that riders spent an average of approximately \$50 per month on the service (Kirsch 2016; URL3 2017).

Lyft is a [transportation network company](#) based in San Francisco, CA, which operates the Lyft car transportation [mobile app](#). Launched in June 2012, Lyft is

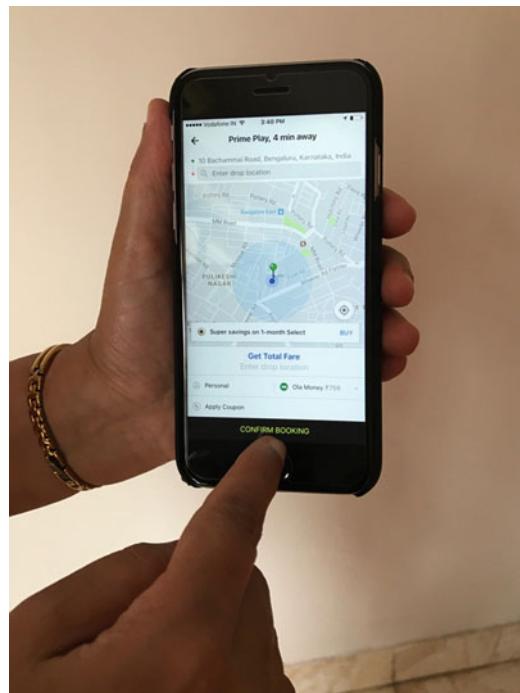
active in approximately 300 US cities, including New York, San Francisco, and Los Angeles, and provides 18.7 million rides a month. The company was valued at \$7.5 billion as of April 2017 and has raised a total of \$2.61 billion in funding (URL4 2017).

Ola Cabs, commonly known as Ola, is an Indian cab service. It started off as an online cab aggregator in Mumbai in December 2010 founded by Bhavish Aggarwal and Ankit Bhati. It is now based out of Bangalore (URL5 2016). It currently offers cab services across more than 100 cities in India with a network of more than 200,000 cars (URL6 2016). Ola has acquired its smaller rival TaxiForSure (TFS) in early 2015, thus giving itself a bigger leadership role in the market. Ola integrated TFS and its services in such a way that former TFS customers can now book TFS cabs and hatchbacks from the Ola app as shown in Fig. 9.4.

Uber, Didi, Lyft, and Ola follow a similar business model in terms of searching for a cab, booking a car/cab, and paying for it. They act as facilitator and aggregator between the customers and the cab drivers. The business model can be summarized as follows:

- Drivers (car owners) join the network and get access to driver's app; customers/riders hail a car with the ride-hailing app.
- The companies charge a commission on the fare (typically 20%) and the driver gets the rest.
- Dynamic pricing.
- The companies have no fleet of their own.

Fig. 9.4 GPS and smartphone enable real-time ride-hailing. Booking confirmation on Ola app



The last point is very important. The cabs are not owned by either Uber, Didi, Lyft, or Ola. The companies just act as middlemen between the customer and the driver. Thus, the investments on the part of the aggregator are minimal, especially in terms of assets and depreciation on these assets. The companies focus instead on marketing, market penetration, and growth as well as IT infrastructure. Also, a sizeable amount of the money is poured into research and development (R&D) topics like new means of transportation, business innovations, and autonomous driving (Abraham et al. 2016).

Ride-hailing services are offered in a four-step model (Sahithi et al. 2016):

- *Cap Request:* The customers can book different categories of cabs based on their size, charges per km/miles, services such as in-vehicle Wi-Fi, and others. The customer can choose a cab as per availability and convenience. For all nearby available cabs of different categories, the app shows the estimated time of arrival (ETA) (see Fig. 9.5). The customer just has to click on the desired cab type to book it.

Fig. 9.5 Uber shows all available cars near to one's location

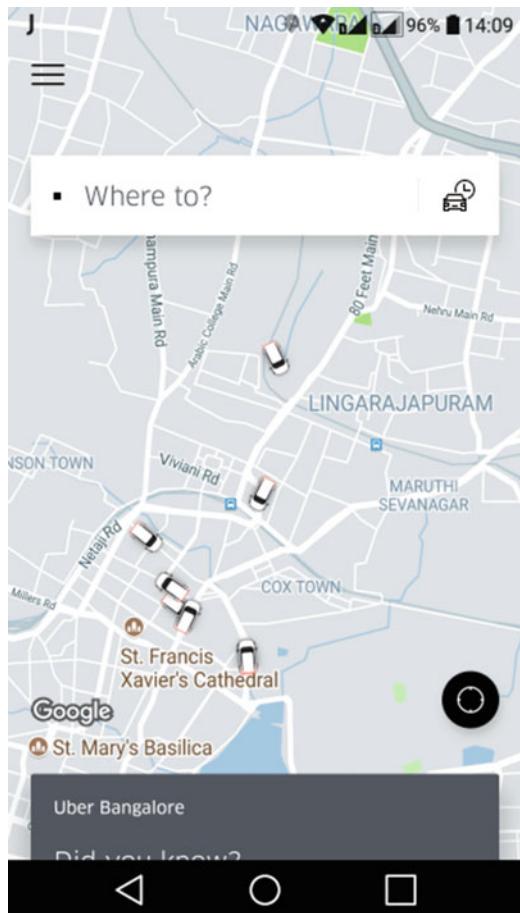
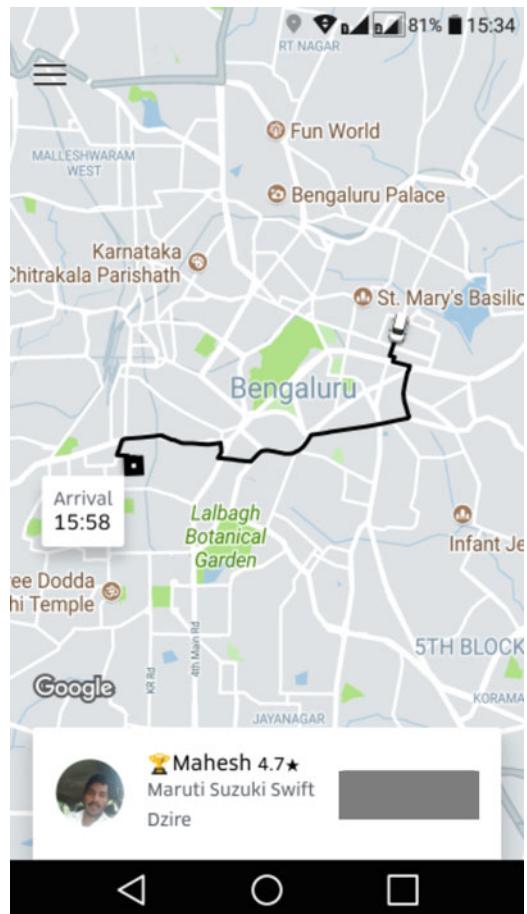


Fig. 9.6 A driver is booked and the estimated arrival time is shown



- *Accept/Reject the Ride:* On getting a cab request, the driver can accept or reject the ride. If the driver rejects the ride, the customer will be notified about it. If the driver accepts the ride, the customer will receive a confirmation notification, and information about the driver's name, phone number, and the vehicle's license number will be displayed on the app. Also, a text message will be sent to the customer (see Fig. 9.6).
- *Track the Ride:* After confirmation, the cab arrives at the customer's location and it can be tracked on the map right from confirmation of the ride until the drop-off. Once the customer gets into the cab and gives approval, the meter starts.
- *Payment:* After the ride is over, the customer will pay for the ride by credit card. Some companies also allow for cash payments or offer online wallet options, depending on the particular country, state or region, such as PayPal (US), Paytm (Uber India) or Ola Money for Ola in India.

MyTaxi is an example of a taxi app and taxi aggregation model. It was taken over by Daimler. The idea is to replace the central taxi call center by a smartphone app. A charge is taken for every ride. The main benefits compared to the telephone-based taxi ordering are:

- Customers do not have to explain their position as the data is transferred automatically
- Customers can track the position of the taxi when it comes to the pick-up location
- Taxi drivers can see the position of the client in a similar way
- The billing is done automatically against the customer's credit card. Many users prefer these cashless transactions
- The bills are electronic and sent automatically, which is convenient for tracking business travel expenses
- On the flip side, charges are higher than those of the central taxi.

The aggregator model was also adopted by the German startup Flixbus for bus rides. The company grew rapidly to become one of the biggest bus operators in Europe (Schlesiger 2016).

9.3 Example Bangalore

In this section we look at ride-hailing in Bangalore as a typical example of service offerings and prices in a metropolitan area. Bangalore has grown rapidly over the last 20 years and has reached a total population size of nearly 10 million inhabitants. Like many fast-growing Asian cities, the city suffers from traffic congestion, pollution, and long waiting times in traffic jams. The public transport is based on buses, taxis, and auto-rickshaws. There is a subway system in place since 2012, but it has to be extended significantly and, so far, is used only by a small fraction of the commuters. Each and everyday people have to commute to the outer satellite cities for work. This situation has attracted international ride-hailing companies like Uber and local competitors like Ola.

Competition for airport pickup in Indian metropolitan cities is especially tough.

While Uber has secured special parking slots at Delhi airport (see Fig. 9.3), Ola cabs have a parking area at Bangalore airport (see Fig. 9.7).

9.3.1 Cab Types and Prices

Ola offers three categories of cabs in Bangalore (Sahithi et al. 2016):

- Ola Micro with cars like Maruti Wagon R, Hyundai Eon, and others
- Ola Mini with cars like Maruti Ritz, Nissan Micra, and Tata Indica
- Ola Prime operating cars like Toyota Etios, Hyundai Accent, and Tata Indigo

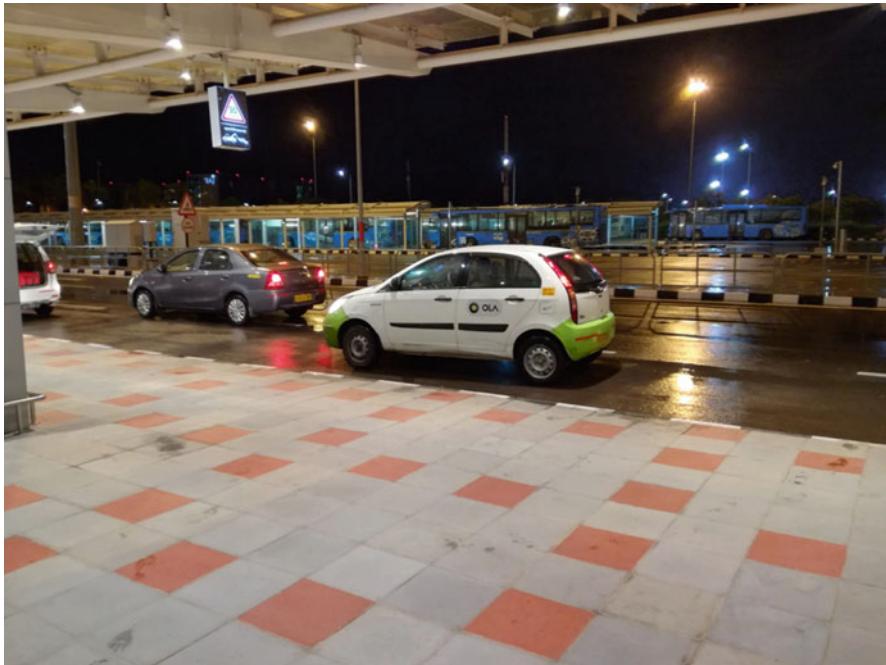


Fig. 9.7 Ola Cab at Bangalore airport

The Ola Share system provides the following:

- Auto
- Lux
- Outstation
- Rentals

All types can accommodate up to four passengers. Ola Prime is targeting business travel and provides Wi-Fi services (URL2 2016). The prices for these categories in 2016 can be found in Table 9.1.

UberPOOL is a ridesharing service with co-passengers who could be picked up or dropped off en route with a slight detour. Maximum seats for a booking are limited to two and are restricted to certain hours of the day and unavailable during night. In Bangalore, Uber provides three categories of cabs (Sahithi et al. 2016):

- UberGo: Cars like Toyota Etios Liva, and Maruti Ritz
- UberX: Cars like Toyota Etios, Hyundai Accent, and Tata Indigo
- UberXL: SUVs or minivans

UberGo and UberX can accommodate up to four passengers; UberXL can be used by up to six passengers. Sample prices for a particular month in 2016 are shown in Table 9.2.

Table 9.1 Ola price chart (URL3 2016)

Standard fare					
Category	Base fare	Distance fare	Wait time fare	Ride time fare	Cancellation fee
Micro	₹40	₹6 per km till 15 km	N/A	₹1.5 per min	₹50
		₹12 per km after 15 km			
Lux	₹300	₹25 per km till 15 km	N/A	₹3 per min	₹100
		₹30 per km after 15 km			
Prime Play	₹50	₹10 per km till 15 km	N/A	₹1 per min	₹75
		₹16 per km after 15 km			
Prime Sedan	₹50	₹10 per km till 15 km	N/A	₹1 per min	₹75
		₹16 per km after 15 km			
Mini	₹50	₹8 per km till 15 km	N/A	₹1 per min	₹50
		₹16 per km after 15 km			
Prime SUV	₹150 for first 4 km	₹17 per km	N/A	₹1 per min	₹100
Share	N/A	N/A	N/A	N/A	₹25
Outstation	N/A	N/A	N/A	N/A	₹150

Table 9.2 Uber price chart

Category	Base fare	Distance fare	Time fare
UberGo	Rs. 42	Rs. 7.35 per km	Rs. 1,58 per min
UberX	Rs. 52,5	Rs. 8.4 per km	Rs. 1.58 per min
UberXL	Rs. 84	Rs.17,85 per km	Rs. 2.1 per km

9.3.2 Services

Ola introduced the ridesharing option Ola Share for single customers to share cabs with other customers intending to travel on the same route (Sahithi et al. 2016). If customers are selective about their co-passengers, they can join groups based on their preferences such as location, workplace, gender, etc. The carpooling option is attractive as it is environment-friendly (URL7 2016) and results in lower fares than if a customer would ride alone. Carpooling is also comparatively more environment-friendly (URL7 2016). Ola also provides customers with a Ride later option where the customer can provide a date (within the next 1 month) and time for the expected pickup. While Ola does not guarantee a cab at the required time, the customer will

not have to search for a cab at that time as they are booked automatically, if available. Ola offers sedans with Wi-Fi availability for an extra charge.

Ola Auto is a popular option considering the fact that auto-rickshaws have always been the vehicle of choice for inexpensive travel in India. They cost less than a cab, and arrive at the door step almost within five minutes of their booking for just an additional Rs. 10 compared to the fare of a standard auto rickshaw. Ola relaunched its outstation taxi service in the Delhi-NCR and Jaipur regions for travels outside the cities (Sahithi et al. 2016; Vikas 2016).

The Prime Minister of India, Mr. Narendra Modi launched this environment-friendly service in 2016 as part of the “Stand-Up India” initiative. While initially only users in the Delhi-NCR region can use this service, Ola announced that it will also launch this feature in small towns and Tier III cities along with ridesharing options in order to bring down costs (URL10 2016).

Uber recently launched the motorcycle service called UberMOTO in Bengaluru and Gurgaon (Agarwal 2016a, b). The motorcycle has a driver, and the customer will be a pillion rider. With fares, as low as Rs. 3/km, this could be an interesting offer for price conscious travelers in the Indian market (URL8 2016).

Uber offers customers with free Wi-Fi in UberX and UberSUV cars. The company has tied up with Airtel to provide 4G connections to drivers who can create a hotspot for the customers. Currently, not all the drivers are trained for it yet.

As of May 2015, Uber seemed to be lagging behind Ola by less than half the installs. Just a year before, both Ola and Uber were neck to neck in terms of their Weekly Active Users (WAU), but Ola started to pick up gradually, and as of October 2015, Ola was leading the market (URL13 2016).

The following Table 9.3 shows the app installation rate over several months (URL11 2017). It is a neck-to-neck race. The drop in the rates for TaxiForSure is due to the takeover of the company by Ola in late 2016.

Truecaller analyzed data from over 130 million registered users on its platforms by calculating the number of calls to numbers which were saved as taxi cab services (URL14 2016). For the first months of 2016 until mid April, it recorded 24.9 billion such calls made to these numbers (URL12 2016).

Out of these calls, around 4.1% calls were made to Ola drivers and 1.6% calls to Uber drivers, which show that Uber is lagging behind Ola by a large margin (URL12 2016).

Ola and Uber have been fighting it out severely to gain monopoly over the \$12 billion Indian taxi market. While current trends show that Ola is leading the race, it must be noted that Uber started out late in the Indian market and was marred

Table 9.3 App installation rate

	Sep 2015 (%)	Feb 2016 (%)	June 2016 (%)	Oct 2016 (%)	Jan 2017 (%)
Ola Cabs	8	11	11	9	10
Uber	3	10	11	12	12
TaxiForSure	1	1	1	0	0

by an assault case on its driver in its initial Indian market days. Since then, Uber has made tremendous progress in the market, and India is now the second biggest market after the USA ([URL3 2016](#)).

9.4 Surge Prices

Surge pricing comes into picture when demand outstrips supply, i.e., if the demand for taxis is higher than the number of available drivers on the road ([Sahithi et al. 2016](#)). Ride-hailing companies claim that surge pricing encourages more drivers to offer rides when the demand is high ([Sivaram 2016](#)), hence, enabling the companies to provide consistent and reliable services.

Let us take a closer look at Uber's surge pricing policy. Under normal circumstances Uber's pricing is similar to metered taxis, although payment is handled exclusively through Uber and not with the driver personally. In some cities, if the car is traveling at a speed greater than 11 mph (18 km/h), the price is calculated on a distance basis; otherwise, the price is calculated on a time basis. Uber uses an automated algorithm to increase prices to "surge price" levels, responding rapidly to changes of supply and demand in the market, and to attract more drivers during times of increased rider demand. The company applied for a US patent on surge pricing in 2013.

The practice has often caused passengers to become upset and invited criticism when it has happened as a result of holidays, bad weather, or natural disasters.

During New Year's Eve 2011, prices were as high as seven times normal rates, and during the 2014 Sydney hostage crisis, Uber charged fares of up to four times the normal charges; while it defended the surge pricing at first, it later apologized and refunded the surcharges. In India, Uber will increase the prices by 20%, 40%, 120%, etc., as per the demand, and the additional fare is given as an incentive to the driver. Like in the USA, this price surge is driven by algorithms and it lasts only for short time, typically a few hours. Depending on the demand and supply gap, the cab rates keep fluctuating. Both Ola and Uber employ similar pricing mechanism.

The problem with the price surging is that in peak hours, the cab aggregators charge two or three times the regular cost, which makes it too costly for many passengers.

In 2016, the Karnataka government temporarily banned surge pricing by cab aggregators like Uber and Ola in Bangalore and set upper limits for the per-km fare.

9.5 Safety in Ridesharing

One of the key issues in the sharing economy is the fact that people have to interact face to face, who often meet for the very first time. This is the case if one lets out a shared room as it is when offering a shared ride. Of course, the platforms do some background check, but the incredible growth often has prevented tight controls.

Evaluation schemes help to warn, but for criminals it is possible to exploit such a system which ultimately is based on trust.

In ride-hailing and sharing this can be particularly dangerous. In this section we look at the problem of safety in more detail, we will discuss some known incidents, and we compare the measures companies have taken to prevent crime and increase the safety both for the customers as well as for the drivers.

The companies provide various safety features such as reducing information asymmetry between users and providers through reviews, ratings, and quality control (Ravindran et al. 2016). Ridesharing allows cash-free transactions and self-identified customers which substantially mitigate one of the worst risks associated with traditional taxis such as the risk of violent crime. Smartphone-based ridesharing technology has gained momentum but still needs to deal with several issues such as safety (such as traveling with strangers) and liability (accidents). Even with various safety measures put into practice, it is still not clear how effective the methods and policies are.

9.5.1 Problem Background

While the ridesharing platforms have made short-distance travel easier and more affordable, safety for both the drivers and the passengers, is still a concern (Ravindran et al. 2016). Some of the main concerns of safety are driver distraction, driving under the influence (DUI); property theft, including robbing homes; physical assault; kidnapping; sexual assault; and murder. Rideshare drivers can pose threats such as stalking the passengers and violating their privacy. Sometimes criminals pretend to be drivers picking up a customer, but then kidnap, harm, or rob the passengers who think they are simply taking a ride and are unaware of the danger. While most ridesharing services perform background checks before contracting their drivers, there is no guarantee that these checks will catch all dangerous drivers. Although, companies may claim that their investigations are adequate, in 2016 the major ridesharing services did not take the fingerprints of their drivers making it easy for individuals to hide their true identities. No background check can assure that the person driving the vehicle is the one who had actually signed up to work for the service. This is a concern for passengers in light of several reports that some rideshare drivers have committed crimes against their passengers. According to www.whosdrivingyou.org, a public awareness website aimed at publicizing crimes by ridesharing drivers, there had been many cases where the drivers are accused of violence or sexual assault against their passengers. The passengers who use ridesharing services are not only afraid of being attacked by drivers but also have to worry about drivers being intoxicated or being distracted. One source of distraction is the smartphone, because the driver is required to rely on his or her phone while driving to get the directions. Some companies have allowed their drivers to schedule new rides while they are transporting a passenger which requires the driver to access the app on his or her phone, look at the location of the request, and choose to accept or decline the new fare. Using a phone while driving is dangerous causing

thousands of accidents every year. Looking down at a phone for even a few seconds is incredibly risky, yet, the ridesharing app encourages drivers to take their eyes off the road. Bad and distracted drivers are not the only safety risk for ridesharing passengers and innocent third parties. Old, poorly maintained vehicles can also pose dangers. Personal vehicles are used very differently than commercial vehicles. Ridesharing drivers use the same vehicles for personal trips. There is a fundamental difference between ridesharing and taxis. Any negligence or irregular inspection of vehicle conditions might result in an accident. While the majority of ridesharing customers arrive at their destinations safely, accidents and injuries happen everyday. The drivers are also vulnerable because their job consists of giving rides to anonymous strangers. Some of the passengers might be aggressive, and the driver might be unaware of their intentions. Here, the driver is as vulnerable as the passenger and is exposed to similar threats that a passenger might be exposed to. Sometimes, even the competitors in the market might be behind an incident trying to pull the market in their favor by damaging their competitor's reputation.

9.5.2 Initiatives to Increase Safety

Ridesharing mitigates many of the safety risks associated with late-night and weekend transport for both riders and drivers (Ravindran et al. 2016). Many ridesharing platforms like Uber and Lyft pay special attention to both driver's and passenger's safety implementing various measures to deal with the most serious and most common risks such as rider violence, poor driver behavior, fare evasion, fare gouging, and mishandled complaints. The companies' ridesharing trips are de-anonymized. This helps to detect unethical or illegal behavior by removing the anonymity of both riders and drivers. Riders know the identity of the drivers, and drivers know the identity of the rider. By recording the personal details of both parties and by recording the route taken, reported incidents can be quickly investigated. All rides must be requested through the app. The entire trip is GPS tracked and riders can share their ETA and route in real-time with friends or family to ensure safety while traveling. Since the payment is being handled by the platform, there is no need to exchange cash. The app facilitates an automatic transaction at the end of the trip. This helps to reduce the threat of cash robbery, fare evasion, or credit card fraud since the calculation and payment of fees is beyond the control of either party. Furthermore, riders and partners should mutually rate one another at the end of each trip. The star rating system is an effective behavioral incentive that delivers good results. However, safety is a continuous improvement process and more innovative ideas are required to ensure better safety in the future.

Uber's toy experiment: Uber has conducted an interesting experiment in Charlotte, North Carolina. The drivers will keep Bop It toys in the back seats of their cars so that the loud, engaging toy will keep aggressive passengers calm (Ravindran et al. 2016). Bop It is a children's toy that consists of a stick or handle with buttons, knobs

and cranks, and a pre-recorded voice that tells users when a button should be bopped, a crank should have been twisted, and a knob should be pulled. This goes on increasingly fast until the player screws up. The effectiveness of this still remains to be proven, but it is believed that a rider who is engaged in something interesting is less likely to become aggressive.

Uber's panic and SOS button in India: After the rape of a female passenger by an Uber driver in India, the company was facing heavy resistance from people as well as government (Ravindran et al. 2016). Many state governments have even considered to ban the Uber ridesharing app in India. As a response, the company came up with new safety features such as panic button and SOS button. When the rider presses the panic button of the Uber app, the system will immediately alert the local authorities, including local police, sharing the user's name, current location, and other trip details available. Hence, it will be very easy to track the location. The app also has an "SOS" button to call the local police directly.

Uber's SafetyNet, India: Uber also came up with the SafetyNet feature in their app (Ravindran et al. 2016), where users can share their ride details along with their GPS location and other relevant information with up to five friends (emergency contacts). With SafetyNet, riders no longer need an SMS plan on their phone to send their status to friends and family. The feature is available to every rider regardless of their particular phone plan.

Ola's Masked Number Feature, India: This feature basically encrypts all calls between riders and drivers so that phone numbers of both parties are protected for added safety. This allows riders and drivers to communicate with each other for coordination on pickup, through anonymous numbers, without revealing their personal phone numbers, providing increased security before, during and after the ride.

Uber's Driver's Safety: Uber drivers benefit just as much as riders from the platform's safety features (Ravindran et al. 2016). The driver is able to see the rider's name ahead of the pickup as well as call the smartphone from which the request was made, ensuring the right person enters the vehicle. Unlike traditional taxi drivers who are required to carry cash, Uber drivers operate on a cashless system and are therefore far less attractive targets for robbers. Also, drivers no longer need to worry about riders running off without paying. A recent study provides evidence that moving from cash-based transactions to the use of electronic payments reduces crimes like robbery and assault. Drivers who partner with Uber also know that any rider can be immediately identified to authorities in the event of an incident, unlike an anonymous taxi passenger. Just as riders rate drivers, drivers can rate riders. As a result, riders are also held accountable for their conduct during the trip.

Ola's One Time Password (OTP): Ola requires the passengers to authenticate themselves by a one-time password sent through the smartphone. That way the driver can be sure that the right passenger, who had actually called for the ride, is

entering the vehicle. The OTP is a four-digit number similar to a PIN code and will only be valid for a short time. The mechanism also helps to avoid rides to be snatched away by other passengers waiting nearby.

Insurance: This is major concern with ridesharing as it falls between traditionally distinct domains of commercial and personal insurance. The full-time professional drivers are covered for any accidents which occur during driving by commercial insurance, whereas people using their personal cars are covered for their private driving. The people who work for ridesharing companies might be working on a part-time basis and don't fall into any of the categories mentioned above. The personal insurance excludes coverage of any insurance for accidents while carrying passengers. Initially, Uber and Lyft have denied any legal responsibilities for accidents that might occur using their services arguing that they are aggregator platforms which just connect drivers with passengers. After considerable pressure from government, the ridesharing platforms started purchasing insurances to cover the accidents involving their drivers. Now, the companies offer \$1 million worth of primary coverage for death, injury, and damages suffered by either drivers or passengers when the other party is at fault ([URL12 2017](#)).

9.5.3 Reported Crime Incidents in Ridesharing

As already discussed above, safety is a main concern among both drivers and customers when they choose a ridesharing platform like Uber or Ola (Ravindran et al. [2016](#)). In recent years, these companies have been facing many complaints as they continued to grow because the process of onboarding drivers is not totally secure which compromises passenger safety. The result is a series of incidents like physical and sexual assaults, kidnappings, felonies, impersonating, driving under influence and even deaths where passengers got harmed and also drivers were assaulted. Some of the past incidents that were reported are:

- On January 22, 2016 in Delhi, an Uber cab driver was arrested for molesting a woman journalist. According to her complaint, the driver was taking a wrong route and she asked him to stop the vehicle. After that, the driver chased her and behaved inappropriately. Uber authorities reacted saying that they would suspend the driver and take necessary steps.
- On December 18, 2014 in Delhi, an Uber cab driver was arrested for raping a passenger. After this incident Uber was asked to temporarily shut down its operations in the city. There were also cases where the drivers had been assaulted.
- On January 23, 2016, in the USA, an Indian-origin doctor verbally abused and physically attacked the driver.

- On January 28, 2016, in Los Angeles, an Uber cab driver was arrested on passenger assault and theft charges. The same person was previously convicted of three felonies which included grand theft and forgery.

Flaws in Uber's background checking system allowed the onboarding of drivers having criminal records which include murder, child abuse, kidnapping etc. (URL2 2015). Also, an information provided by Uber to customers turned out to be incorrect as some drivers and customers gave fake details when registering with the platform.

9.5.4 Government Policies for Ridesharing Companies

Most of the safety concerns are country specific. Below are some examples for government policies in India, China and US which regulate ridesharing in these countries.

In India, taxi aggregators, ridehailing, and ridesharing companies must comply with the following rules:

- The cars used for ridesharing must be fitted with emergency safety buttons, and the app must include an emergency call feature.
- Ridesharing companies must conduct a criminal background check on the drivers before recruiting them.
- Cars must have devices fitted for tracking their location.
- The aggregator company must operate 24/7 call centers.
- The vehicle should have a valid pollution under control (PUC) certificate.
- The aggregator company must provide an office address within the area of operation.
- The service provider must maintain a database of all the drivers.

China has imposed a similar set of rules:

- The cars must be registered as taxis and not as private vehicles.
- The aggregator companies must have insurance in place to cover the vehicle and the passengers.
- The company must have formal employment contracts with the drivers.
- The company must have servers in China and share information with the transport authorities.
- Government can restrict the number of cars allowed as well as the area in which they can be operated.

In the US, the city of Chicago has imposed the following set of rules:

- Drivers have to get either a restricted chauffeur's license or a taxi chauffeur's license.
- They are required to take a 6-day course and have to pass a test.

9.5.5 Legal Cases and Accusations

The battle between the two leading taxi services in India requires millions of their investors' money as they fight for market share in the Indian taxi market which was already worth \$12 billion in 2016.

- An interesting case is still in court in which Ola filed a lawsuit accusing Uber of flouting a court order to switch to clean-fuel cars in Delhi after Ola already had done this.
- In 2015 Uber sued Ola for \$7.5 million accusing it of creating fake user accounts with Uber and using them to make more than 405,000 false bookings. Ola has denied the allegations.
- Uber also accused Ola of wrongful interference which led to 23,000 of its drivers quitting.

9.6 Cyberattacks and Cybersecurity in Ridesharing

Cybersecurity risks in ridesharing are real and are taken very seriously by the providers. Uber for example hired the two specialists of the Jeep Hack, Miller and Valasek, to build up a cybersecurity threat intelligence and response division. Other ridesharing companies have similar units and are investing heavily into cybersecurity protection. Typical cybersecurity concerns (see also Chap. 6) are:

- Attacks on the billing system
- Bot algorithms that call a cab and cancel the ride
- Compromising the security of the driver's smartphone by injecting malware
- DDoS attacks on the infrastructure
- Fake accounts
- Hacked and compromised accounts

Also, there is fraud exploiting the incentive systems. Uber, for example, experimented with incentives for drivers in India to do short and frequent trips. This potentially could have increased the market share, but the concept was exploited. Drivers asked the passengers to stop over and call the cab again, thereby splitting up a long-distance ride into multiple smaller trips, ultimately causing losses. One way to detect this problem is to filter the transaction data screening for patterns that suggest a misuse of the incentive system. Drivers who try to squeeze out more incentives can be identified and—if the behavior repeats—will be fired. The same applies to users who cancel trips unnecessarily or try to persuade the driver to extend the tour. They can be shunned from the service.

As the competition, especially in emerging markets like China and India, is very tough, lots of DDoS attacks were reported. An indicator of the level of competition is

the \$1 billion loss which Uber reported in China in 2015. Uber finally decided to sell off their China operation to Didi.

9.7 Conclusion

Ridesharing and ride-hailing are promising ways to reduce urban traffic and to save costs. It is possible to save time as one can track the directions and request for a vehicle without waiting for long hours. However, it is important to seriously consider the risks of ridesharing. Even though the ridesharing companies claim that they have adequate safety measures in place, the effectiveness of enforcing the policies and maintaining strict standards of background checks to rule out fake identities is still a topic of debate.

Ride-hailing and ridesharing are still restricted or even illegal in many countries, but enforcing proper policies, strict cybersecurity, and proper insurance coverage can ensure that these services will be suitable for inexpensive mobility.

9.8 Exercises

What is meant by the term *ridesharing*?

Describe some key aspects of ridesharing.

What is meant by the term *ride-hailing*?

Describe the differences between ridesharing and ride-hailing.

What is meant by the term *carpooling*?

Describe some key aspects of carpooling.

What is meant by the term *carsharing*?

Describe the differences between carpooling and carsharing.

What is meant by the term *dynamic pricing*?

Describe some key aspects of dynamic pricing.

What is meant by *unfair competition between ride-hailing companies*?

Describe the unfair competition between ride-hailing companies.

What are the *specific needs of ride-hailing in emerging economies*?

Describe some key aspects.

What is meant by the term *new mobility trends*?

Describe some key aspects of new mobility trends.

Why are *automotive OEMs interested in investing into ride-hailing companies*?

Describe the key reasons and give an example.

What are the *major safety concerns in ride-hailing and ridesharing*?

Describe the key aspects and give an example.

What are *critical attack vectors on a ride-hailing service*?

Describe the key aspects and give an example.

What is meant by the term *authentication mechanisms*?

Describe the key aspects and give an example.

What is meant by the term *cheating*?

Describe how to make sure that the customers do not cheat and how drivers are being prevented from cheating.

What is meant by the term *cybersecurity*?

Describe the key aspects and give an example.

What is meant by the term *incentive mechanism*?

Describe what problems can arise with the incentive mechanism and how can incentives lead to cheating.

Why has *Ola* been so successful?

Describe Ola and Uber and compare their similarities and differences.

Why is *Uber* investing so much in autonomous driving?

Describe the key aspects and give an example for autonomous driving.

Why is *ride-hailing* an example of a shared economy?

Describe the defining characteristics.

References and Further Reading

- (Abraham et al. 2016) Abraham, B., Brugger, D., Strehlke, S., Runge, W.: Autonomous driving only a trojan horse of digital companies? (in German). ATZ elektronik, 01/2016
- (Agarwal 2016a) Agarwal, S.: Bike service UberMOTO to debut in Bengaluru today; fares as low as Rs 3/km. Economic Times India online. March 3rd 2016. Available from: http://economictimes.indiatimes.com/articleshow/51231622.cms?utm_source=contentofinterest&utm_medium=text&utm_campaign=cppst
- (Agarwal 2016b) Agarwal, M.: Uber Claims to Surpass Ola in Number of Rides; Globally Positions India as Number 2 Market after US. Inc42 online. October 5th 2016. Available from: <https://inc42.com/buzz/uber-surpass-ola/>
- (Alan and Brem 2013) Alan, T., Brem, A.: A conceptualized investment model of crowdfunding. In: Venture Capital, Vol. 15, No. 4, pp. 335–359, 2013
- (Alheeti et al. 2015) Alheeti, K. M. A., Gruebler, A., McDonald-Maier, K. D.: An Intrusion Detection System against Black Hole Attacks on the Communication Network of Self-Driving Cars. In: 6th International Conference on Emerging Security Technologies (EST), pp. 86–91, 2015
- (Amey 2010) Amey, A. M.: Real-Time Ridesharing – Exploring the Opportunities and Challenges of Designing a Technology-based Rideshare Trial for the MIT Community. Master Thesis Massachusetts Institute of Technology (MIT), MA, USA, 2010
- (Argus 2017) Argus Cybersecurity – Protecting Cars, Trucks and Commercial Vehicles from Hacking – an Overview. Argus Cyber Security. Available from: <https://argus-sec.com/car-hacking/>
- (Badger 2014) Badger, E.: Taxi Medallions have been the Best Investment in America for Years – Now Uber may be changing that. In: Wonkblog, Washington Post, 2014
- (Barro 2014) Barro, J.: Under pressure from Uber, taxi medallion prices are plummeting. In: The New York Times, 2014
- (Bay 2016) Bay, L.: Carsharing and Ridesharing in Germany – They Growing up so Fast. Wirtschaftswoche online. March 26th 2016. Available from: <https://www.wiwo.de/unternehmen/industrie/carsharing-und-ridesharing-in-deutschland-sie-werden-so-schnell-gross/13368794.html>
- (Berger 2015) Berger, R.: Total Cost of Car Ownership over its Lifetime. The Dough Roller. April 22nd 2015. Available from: <http://www.doughroller.net/smart-spending/true-cost-of-a-car-over-its-lifetime/>

- (Cahn and Shaheen 2012) Chan, N. D., Shaheen, S. A.: Ridesharing in North America: Past, Present, and Future. *Transport Reviews*, Vol. 32, No. 1, pp. 93–112, 2012
- (Currie 2015) Currie, R.: Developments in Car Hacking. <https://www.sans.org/reading-room/whitepapers/internet/developments-car-hacking-36607>
- (Eckl-Dorna 2016) Eckl-Dorna, W.: VW puts millions in car ridehailing – This is how gett VW's new app investment works. Manager Magazin online. May 25th 2016. Available from: <http://www.manager-magazin.de/unternehmen/autoindustrie/taxi-vermittlungsdienst-gett-so-funktioniert-vws-neues-investment-a-1094141.html>
- (Fallstrand and Lindstrom 2015) Fallstrand, D., Lindstrom, V.: Automotive IDPS: Applicability analysis of intrusion detection and prevention in automotive systems. Available: <http://publications.lib.chalmers.se/records/fulltext/219075/219075.pdf>
- (Freitag et al. 2015) Freitag, M., Maier, A., Palan, D.: Apple, Google, Facebook, Uber American Hybris, Manager Magazin online. April 28th 2015. Available from: <http://www.manager-magazin.de/magazin/artikel/apple-google-facebook-uber-groessenwahn-im-silicon-valley-a-1030869.html>
- (Haykin 2009) Haykin, S.: Neural Networks and Learning Machines. 3rd edition. Pearson Education, 2009
- (Hirn 2016) Hirn, W.: Car Rental Services in China – Didi vs. Uber – The billion Dollar battles of the Chinese cousins (in German). Manager Magazin online. July 27th 2016. Available from: <http://www.managermagazin.de/finanzen/artikel/a-1105011.html>
- (Kirsch 2016) Kirsch, S.: Mobile Payment – How Chinese establish Mobile Payment in Germany (in German). Wirtschaftswoche online. August 2nd 2016. Available from: https://www.wiwo.de/finanzen/geldanlage/mobiles-bezahlen-wie-chinesen-mobiles-bezahlen-in-deutschland-establieren/v_detail_tab_print/13937126.html
- (Koch 2017) Koch, L.: Greyball – Uber used secret software (in German). Zeit online. March 4th 2017. Available from: <http://www.zeit.de/digital/2017-03/greyball-uber-software-kontrollen-polizisten-umgehung>
- (IET 2015) Automotive CyberSecurity: An IET/KTN Thought Leadership Review of riskperspectives for connected vehicles. IET. Available from: <http://www.theiet.org/sectors/transport/documents/automotive-cs.cfm>
- (Isaac and Perlroth 2015) Isaac, M. and Perlroth, N.: Uber Hires Two Engineers Who Showed Cars Could be Hacked. The New York Times online. August 28th 2015. Available from: <https://www.nytimes.com/2015/08/29/technology/uberhires-two-engineers-who-showed-cars-could-be-hacked.html?mcubz=0>
- (Intel 2015) Intel Security White Paper Automotive Security Best Practice. Intel/Mcafee. June 2016. Available from: <https://www.mcafee.com/enterprise/en-us/assets/white-papers/wp-auto-motive-security.pdf>
- (Kashyap 2016) Kashyap, K.: It's Uber Vs. Ola For The Battle Of Supremacy. In: The Indian Market. Forbes online. September 21st 2016. Available from: <https://www.forbes.com/sites/krnkashyap/2016/09/21/its-uber-vs-ola-for-the-battle-of-supremacy-in-the-indian-market/#6ffe5799d99f>
- (La Vinh and Cavalli 2014) La Vinh, H., Cavalli, A. R.: Security attacks and solutions in vehicular ad hoc networks: a survey. *International Journal on Ad Hoc Networking Systems (IJANS)*, Vol 4, No. 16, pp. 1–20, 2014.
- (Lobe 2016) Lobe, A.: Hacker Alert – In a modern car today are computers and info systems that are easy to manipulate. How do the manufacturers deal with the security gap? (in German). Zeit online. August 25th 2016. Available from: <http://www.zeit.de/2016/34/elektroautos-steuerung-hacker-gefahr-sicherheit-hersteller>
- (Mahaffey 2015a) Mahaffey, K.: The New Assembly Line: 3 Best Practices for Building (secure) Connected Cars. Lookout Blog. August 6th 2015. Available from: <https://blog.lookout.com/tesla-research>
- (Mahaffey 2015b) Mahaffey, K.: Here Is How To Address Car Hacking Threats. TechCrunch. September 13th 2015. Available from: <https://techcrunch.com/2015/09/12/to-protect-cars-from-cyber-attacks-a-call-for-action/>

- (Markey 2015) Markey, E.J.: Tracking and Hacking: Security and Privacy Gaps Put American Drivers at Risk. 2015. Available from: https://www.markey.senate.gov/imo/media/doc/2015-02-06_MarkeyReport-Tracking_Hacking_CarSecurity%202.pdf
- (McMillan 2011) McMillan, R.: With Hacking, Music Can Take Control of Your Car. PCWorld online. March 11th 2011. Available from: https://www.pcworld.idg.com.au/article/379477/hacking_music_can_take_control_your_car/
- (Miller and Valasek 2014) Miller C., Valasek C.: A Survey of Remote Automotive Attack Surfaces. IOActive 2014. Available from: https://www.ioactive.com/pdfs/IOActive_Remote_Attack_Surfaces.pdf
- (Poulsen 2010) Poulsen, K.: Hacker disables more than 100 cars remotely. Wired online. March 17th 2010. Available from: www.wired.com/threatlevel/2010/03/hacker-bricks-cars
- (Ravindran et al. 2016) Ravindran, D., Hanisha, N., Punati, S.: Safety in Ride Sharing. Class Paper, Car IT and Cybersecurity, International Institute of Information Technology Bangalore (IIIT-B), May 2016
- (Sahithi et al. 2016) Sahiti, A., Ramya Reddy, D., Vedavyas, M.: OLA vs Uber. Report: Car IT and Cybersecurity, International Institute of Information Technology Bangalore (IIIT-B), May 2016
- (Schlesiger 2016) Schlesiger, C.: Flixbus – The creepy triumphal march of the startup. Wirtschaftswoche online. October 17th 2016. Available from: http://www.wiwo.de/unternehmen/dienstleister/flixbus-der-unheimliche-siegeszug-desstart-ups/v_detail_tab_print/14680498.html
- (Schultz 2016) Schultz, M.: Billion loss at Uber – The evil of button-press capitalism (in German). Spiegel online. August 26th 2016. Available from: <http://www.spiegel.de/forum/wirtschaft/milliardenverlust-bei-uber-das-uebel-desknopfdruck-kapitalismus-thread-505569-1.html>
- (Spehr 2016) Spehr, M.: Internet connection in the Audi A4, Behind the steering wheel, Google shows the world (in German), FAZ online. August 18th 2016. Available from: <http://www.faz.net/aktuell/technik-motor/motor/kommunikationstechnik-des-audi-a4-im-test-14387527/der-audi-a4-kommt-mit-14390627.html>
- (Scarfone and Mell 2007) Scarfone, K., Mell, P.: Guide to Intrusion Detection and Prevention Systems (IDPS) NIST. February 20th 2007. Available from: <https://www.nist.gov/publications/guide-intrusion-detection-and-prevention-systems-idps>
- (Sana 2016) Sana, S.: Ola, Uber Operating Taxis illegally in City, Delhi Government Tells HC. The Times of India. April 24th 2016. Available from: <http://timesofindia.indiatimes.com/city/delhi/Ola-Uber-operating-taxis-illegally-in-city-Delhigovernment-tells-HC/articleshow/51962298.cms>
- (Schorsh 2016) Legislative Deal between Ridesharing Companies and Taxis Dies – Florida Politics. Florida Politics. March 11th 2016. Available from: <http://floridapolitics.com/archives/204373-sources-legislative-deal-reachedridesharing-companies-taxis>
- (Serio and Wollschläger 2015) Serio, G., Wollschläger, D.: Networked Automotive Defense Strategies in the Fight Against Cyber Attacks (in German). ATZ elektronik, 06/2015
- (Sivaram 2016) Sivaraman, S.: Explaining Ola and Uber's Surge Pricing. The Hindu online. September 11th 2016. Available from: <http://www.thehindu.com/news/national/explaining-ola-and-ubers-surge-pricing/article8494839.ece>
- (Solon 2015) Solon, O.: From Car-Jacking to Car-Hacking: How Vehicles Became Targets For Cybercriminals. Bloomberg online. August 4th 2015. Available from: <https://www.bloomberg.com/news/articles/2015-08-04/hackers-force-carmakers-to-boost-security-for-driverless-era>
- (Stewart 2016) Stewart, J.: As Tesla grows up, it gives up on free charging. Wired online. July 11th 2016. Available from: <https://www.wired.com/2016/11/tesla-grows-gives-free-charging/>
- (Symantec 2015) Symantec IoT Team – Building Comprehensive Security Into Cars. Technical Report. Symantec. 2015. Available from: <https://www.symantec.com/content/dam/symantec/docs/white-papers/building-comprehensive-security-into-cars-en.pdf>
- (Vembo 2016) Vembo, D.: Connected Cars – Architecture, Challenges and Way Forward. Whitepaper Sasken Communication Technologies Pvt. Ltd. 2016. Available from: https://www.sasken.com/sites/default/files/files/white_paper/Sasken-Whitepaper-Connected%20Cars%20Challenges.pdf

- (Vestlund 2009) Vestlund, C.: Intrusion Detection Systems in Networked Embedded Systems. Linköping University. Available from: <https://pdfs.semanticscholar.org/10f9/455dde5674de051ae065f358b922cf8bec0f.pdf>
- (Vikas 2016) Vikas, S. N.: Ola relaunches Outstation Taxi Service in Delhi and Jaipur. ETtech online. May 11th 2016. Available from: <http://tech.economictimes.indiatimes.com/news/mobile/ola-outstation-taxi-service/52201260>
- (Wolfsthal and Serio 2015) Wolfsthal, Y., Serio, G.: Made in IBM Labs: Solution for Detecting Cyber Intrusion to Connected Vehicles, Part I. IBM. September 9th 2015. Available from: <https://securityintelligence.com/made-in-ibm-labs-solution-for-detecting-cyber-intrusions-to-connected-vehicles-part-i/>
- (Zetter 2015) Zetter, K.: Researchers Hacked A Model S, But Tesla's Already Released A Patch. Wired online. August 6th 2015. Available from: <https://www.wired.com/2015/08/researchers-hacked-model-s-teslas-already/>

Links

2015

- (URL1 2015) <https://www.wired.com/2015/08/uber-hires-hackers-wirelessly-hijacked-jeep/>
(URL2 2015) <https://www.businessinsider.in/Uber-hired-a-convicted-murderer-who-applied-with-a-fake-name-complaint-claims/articleshow/48550254.cms>

2016

- (URL1 2016) <https://techcrunch.com/2016/12/02/didis-cto-explains-why-chinas-ride-sharing-giant-has-a-data-advantage/>
(URL2 2016) <https://help.olacabs.com/support/dreport/205018962>
(URL3 2016) <https://www.olacabs.com/fares>
(URL4 2016) <https://www.uber.com/cities/bangalore/>
(URL5 2016) https://en.wikipedia.org/wiki/Ola_Cabs
(URL6 2016) https://www.olacabs.com/info/about_us
(URL7 2016) <http://www.olashare.com/>
(URL8 2016) <https://newsroom.uber.com/india/ubermotoblr/>
(URL9 2016) <http://indianexpress.com/article/technology/tech-news-technology/uber-vs-ola-heres-everything-that-has-happened-so-far-2780273/>
(URL10 2016) <http://indianexpress.com/article/technology/tech-news-technology/ola-launched-e-rickshaw-category-for-delhi-ncr/>
(URL11 2016) <http://www.bbc.com/news/technology-36139986>
(URL12 2016) <https://thetechportal.com/2016/05/04/ola-far-ahead-uber-india-according-truecaller-data/>
(URL13 2016) <https://inc42.com/buzz/ola-vs-uber/>
(URL14 2016) <https://blog.truecaller.com/2017/05/16/truecaller-insights-2016-q4-report-call-volume-for-the-e-commerce-cab-hailing-industry-in-india/>
(URL15 2016) <http://in.reuters.com/article/uber-ola-india-copy-idINKCN0XR033>
(URL16 2016) <http://www.welt.de/wirtschaft/article157748150/Uber-will-Fahrgaeste-per-Autopilot-chauffieren.html>

2017

- (URL1 2017) https://en.wikipedia.org/wiki/Real-time_ridesharing
- (URL2 2017) [https://en.wikipedia.org/wiki/Uber_\(company\)](https://en.wikipedia.org/wiki/Uber_(company))
- (URL3 2017) https://en.wikipedia.org/wiki/Didi_Chuxing
- (URL4 2017) <https://en.wikipedia.org/wiki/Lyft>
- (URL5 2017) <https://www.olacabs.com/fares>
- (URL6 2017) <https://www.uber.com/>
- (URL7 2017) <https://www.olacabs.com/>
- (URL8 2017) <https://www.forbes.com/sites/briansolomon/2017/01/04/didi-chuxing-invests-in-brazilian-uber-rival-reignites-ridesharing-war/>
- (URL9 2017) <http://www.manager-magazin.de/unternehmen/it/google-roboterwagen-fahren-aus-angst-vor-hackern-ohne-internet-a-1129346.html>
- (URL10 2017) <https://www.statista.com/chart/7424/uber-is-closing-in-on-volkswagen/>
- (URL11 2017) <http://blog.jana.com/blog/top-ride-sharing-apps-in-emerging-markets>
- (URL12 2017) <http://time.com/money/4851877/my-uber-got-into-a-wreck-can-i-sue/>



Connected Parking and Automated Valet Parking

10

This chapter discusses one of the most relevant and straightforward application of connected cars—connected parking. Everybody who is driving a car has had some experience with the difficulty to find a parking space and to park the car in narrow lots. Fortunately, technology is available to help, and it will potentially have a major impact on traffic, parking accidents and space utilization in cities. After a brief discussion of parking from a business perspective in Sect. 10.1, analyzing the main challenges, the chapter discusses the opportunities for connected parking in Sect. 10.2. A multitude of new apps provides information, often in real-time, about available parking spaces; manages the booking; often allows for cashless billing; and can be integrated with OEM's connectivity services. This chapter gives an overview of major players and discusses the core features and services of their solution. Section 10.3 presents parking assistance systems. The most sophisticated—as of today—automates the complete parking process; however, the driver still has to be in the car and has to oversee the process. The next step, automated valet parking (AVP), is discussed in Sect. 10.4. AVP systems turn the vehicle into a robot car that automatically finds parking space and maneuvers the car into a free slot. The first commercial systems will soon be available in high-end cars, and also will be deployed for carsharing. Sections 10.5 and 10.6 deal with the cybersecurity impact of connected parking and automated valet parking, analyzing the major cyber threats and look at potential solutions for increasing the cybersecurity, like intrusion detection and prevention (see also Chap. 6). Such systems are in place to protect large-scale IT infrastructure and recently have been applied to the cybersecurity of vehicles, which is discussed more in detail in Chap. 6. Section 10.7 finally wraps up with a conclusion and recommended further readings. Section 10.8 contains a comprehensive set of questions on connected parking and automated valet parking and the final section includes references and further readings.

10.1 Parking

The parking industry is comprised of many players and stakeholders (URL1 2017). Parking space at airports, railway stations, shopping malls, and public park-and-ride lots accounts for large real estate demand in urban areas. People spend a lot of time searching for parking space. Volkswagen (VW) estimates that up to 30% of the inner-city traffic is due to the search for parking space (Jungwirth 2016; Gerster 2016; Rees 2016). One can differentiate on-street parking, often managed by cities or freely available and off-street parking in special marked places, like huge parking lots, multistorey buildings, and park-and-ride facilities. These off-street parking areas are typically managed by large parking operators like Apcoa (URL21 2017). They lease the space, provide the necessary infrastructure, and manage the billing. The biggest car park operators in Europe are Apcoa, Germany, with a 10% market share; Q-Park, Netherlands (URL22 2017); and Contipark, Belgium (URL23 2017), as shown in Fig. 10.1.

The car park business and the market are characterized by the following:

- Market is extremely fragmented, especially in Europe.
- Car park operators typically do not own the properties themselves but operate and maintain them.
- There are also many regional small- and mid-sized companies, as well as cities that manage the car parks and park houses themselves.
- Market is very competitive, operators are struggling with low margins, and some are even under loss (Dierig 2012). Because of this, park operators are reluctant to invest which often leads to old IT and outdated infrastructure.
- Car park operators are looking for new services to boost their revenues and profits, for example, cleaning the cars while being parked.



Fig. 10.1 The European market for parking is fractured - some of the leading parking operators

- New parking operators like Park One (URL20 2017) are focusing on services like valet parking.
 - Cities cooperate with new players like Cleverciti, Parkpocket, and others (URL19 2017; URL3 2017), to provide real-time data about availability of parking space, both on-street and off-street, cutting down on search time and inner-city traffic.
-

10.2 Connected Parking

Finding parking space in a crowded city is a problem, which goes far beyond wasting time and it is one of the major reasons for the boom of carhailing and ridesharing providers. The main issues and challenges can be summarized as follows:

- Up to 30% of inner-city traffic is due to the search for parking space (Gerster 2016).
- Parking in cities can be very costly.
- Often, it is not clear where to find parking space.
- Looking out for parking while driving, especially, if one is alone in the car, is one of the main reasons for traffic accidents in the cities.
- Up to 40% of all accidents are related to parking (Gerster 2016).
- Theft and damage to the car in a parking lot is a major problem.
- Finding the car in a big parking lot is not easy.

The smartphone and the ideas of the shared economy (Laudon et al. 2010) also had a major impact on the way people park today.

Many start-ups have come up offering apps for connected parking (URL2 2017; URL8 2017; URL9 2017; URL10 2017; URL11 2017; URL15 2017):

- Parkopedia
- JustPark
- SpotHero
- ParkWhiz
- ParkingPanda
- BestParking

For on-street parking some of the popular apps are (URL12 2017; URL13 2017):

- Parker
- ParkMe
- ParkNow

The main idea is to provide information about available parking space, thereby, minimizing unnecessary traffic to search for a parking lot. Many apps use a community-/crowd-based mechanism to notify other drivers about available parking space. A particularly valuable information is the price of parking space which can vary

widely. Booking and reservation of a parking spot is another helpful service, often with attractive offers. Finally, the billing can be handled automatically, avoiding all sorts of hassles when trying to pay at the ticket machine because of not enough cash, no change, loss of park ticket, credit card not accepted, machine is down, and other possibilities. Also, electronic tracking of the invoices for parking is a nice feature that people appreciate in travel booking systems like Concur.

Big benefits can be achieved if the information is given in real-time. Some companies, e.g., the German start-up Ampido, are offering an Airbnb-type business model where one can advertise private car parks and rent them out via the platform ([URL16 2017](#)).

Parkopedia has undergone an interesting development. Started first as a kind of encyclopedia for parking space, aggregating available information about parking lots in various cities, it has added more and more features to the system, like real-time availability of parking, reserving space, billing, and predictive parking.

It is clear that the value of a parking platform grows with the reach (cities and regions), the number of parking lots, and the size of the user community. Predictive parking, the deployment of big data and machine learning to estimate free parking, is a promising area where parking apps can differentiate themselves from each other ([URL18 2017](#)). The best results can be achieved if data from all available sources is being combined, e.g.:

- Real-time traffic information
- Real-time data from parking operators
- Community-based information fed in through special apps or automatically via sensors (e.g., see Bosch IoT cloud)
- Statistical models taking into account time, date, special events, vacation time, etc.

Table 10.1 summarizes the core features of different parking solutions. Automotive OEMs have partnered up with connected parking apps to provide solutions for their car fleet, e.g., Daimler works with GottaPark ([URL14 2017](#)) and BMW integrates the services of Parkopedia ([URL2 2017](#)) into their connected drive infotainment system. Navigation and map providers, e.g., Tomtom ([URL7 2017](#)), also work with companies like Parkopedia to integrate parking value added services in their maps.

In Fig. 10.2 a different classification is shown based on the criterias on-street versus off-street and static versus dynamic real-time status of parking occupancy.

Real-time information about parking space can be gathered in various ways:

- Off-street: The parking operator can keep track and feed the information to a server.
- Specific sensors, embedded in the parking lot, can track, if a space is occupied or not. This can work both on-street and off-street, e.g., see projects by Siemens ([URL31 2017](#)), Bosch, ParkHere ([URL29 2017](#)).

Table 10.1 Features of parking apps compare (Chandrasekar et al. 2013; URL34 2017)

Parking app provider	Real-time parking and navigation	Parking reservation	Parking payment	On-street parking management	Region
Parkopedia	Yes	Yes	Yes	Yes	EU/US/APAC
ParkatmyHouse	Yes	Yes	Yes	No	EU/US
JustPark		Yes	Yes		EU
ParkNow	No	No	Yes		US
GottaPark	Yes	Yes	Yes	No	US
ParkingPanda	No	Yes	Yes	No	US
Streetline	Yes	Yes	Yes	Yes	EU/US

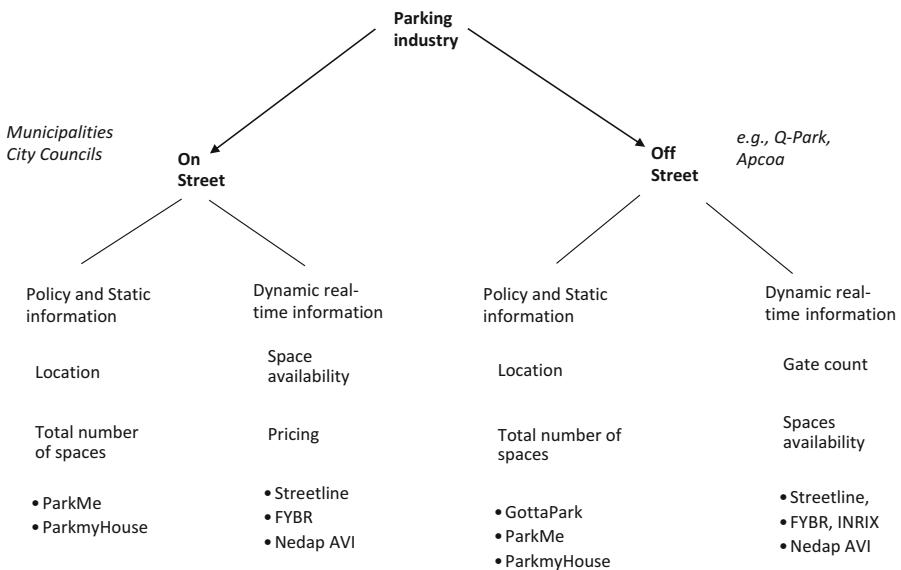


Fig. 10.2 Classification of parking apps/connected parking (Chandrasekar et al. 2013; URL34 2017)

- Car can gather information about free parking space when driving by using the in-built ultrasound sensors or cameras.
 - Car can measure the dimension of free parking space.
 - A car driver can explicitly notify others with a smartphone app about parking space availability.
 - Movement of the car can be tracked automatically and can be used to predict if a parking space will be available.

Several parking operators like Apcoa are experimenting with a touchless access to the park house, e.g., based on an embedded RFID chip which is used for authentication and automatic payment. The potential of connected parking has attracted various large companies like SAP, Cisco, and others (URL1 2014; URL1 2016). The IAA 2015 provided a platform for several interesting start-ups that are now collaborating with some of the large players (URL3 2017; URL9 2015).

The San Francisco based company Streetline (URL4 2017) has a full service offering for cities, parking space operators, and car owners. It includes community-based real-time parking information, map integration of free parking spaces, billing, software systems and dashboards for off-street parking for cities and parking operators, as well. The analytics platform Parksight helps to optimize throughput, efficiency, and pricing decisions.

Another major player of parking solutions in the US is INRIX (URL5 2017). In 2014, Porsche acquired a 10% stake. The company offers community-based parking through its cloud platform, real-time parking information and solutions for parking operators and cities as well. Other services include real-time traffic information, state-wide traffic analytics, traffic collisions, parking data and analytics, connected car services, as well as traffic count and population movement insights. INRIX works with automakers and government agencies to understand how people and commerce move across the world's transportation network (URL31 2017).

In Fig. 10.3 the concept of community-based parking is explained (Nicodemus and Auracher 2015; URL25 2017; URL26 2017). If a car passes by an empty parking space on the side street, it will automatically scan the dimension with the in-built ultrasound and/or radar sensors. This can be done up to a speed of 50 km/h, i.e., within the speed limits of city traffic. The sensors can not only detect if a parking space is available but also how big the area is, thereby, providing information about

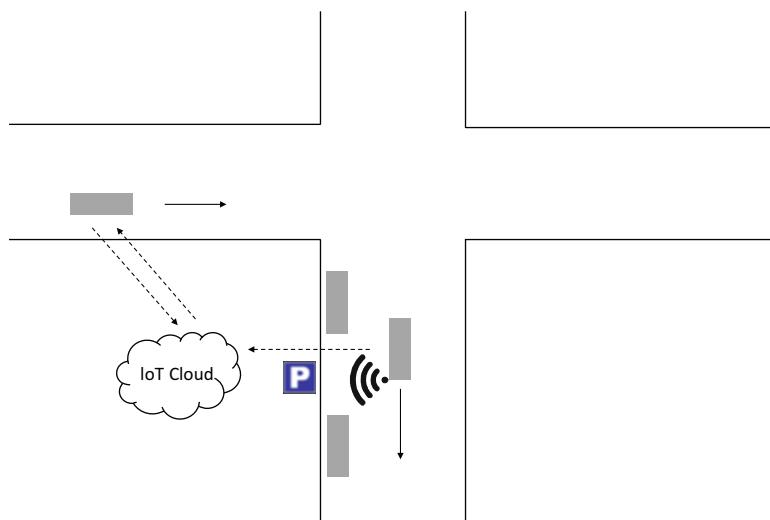


Fig. 10.3 Principle of Community-based parking (URL24 2017; URL26 2017; URL27 2017)

which type of cars could fit and which will not fit into the space. This information is gathered automatically and fed into the Bosch IoT cloud (URL27 2017). Here, the information is consolidated and made readily available to other cars which have subscribed to the cloud-based parking service.

In Fig. 10.3 the driver of the car on the left searches for parking space. The free parking space in the side street is not visible, but the information about a suitable parking lot which is currently free is available from the IoT cloud and shown on the map. The driver can decide to take it and pull into the parking space. If this happens, the system will automatically notify all other user that the parking space is now occupied.

The digital transformation of the parking industry leads to many new partnerships between OEMs, first tiers, start-ups, fleet managers, ride-hailing companies, etc. and has sparked interesting new business models. Figure 10.4 summarizes the different aspects of connected parking in a mind map (Nicodemus and Auracher 2015; Gebhardt 2016; URL24 2017). Frost and Sullivan have analyzed the parking industry and identified a point of conversion of partnerships and alliances as shown in Fig. 10.5 (Chandrasekar et al. 2013; URL35 2017).

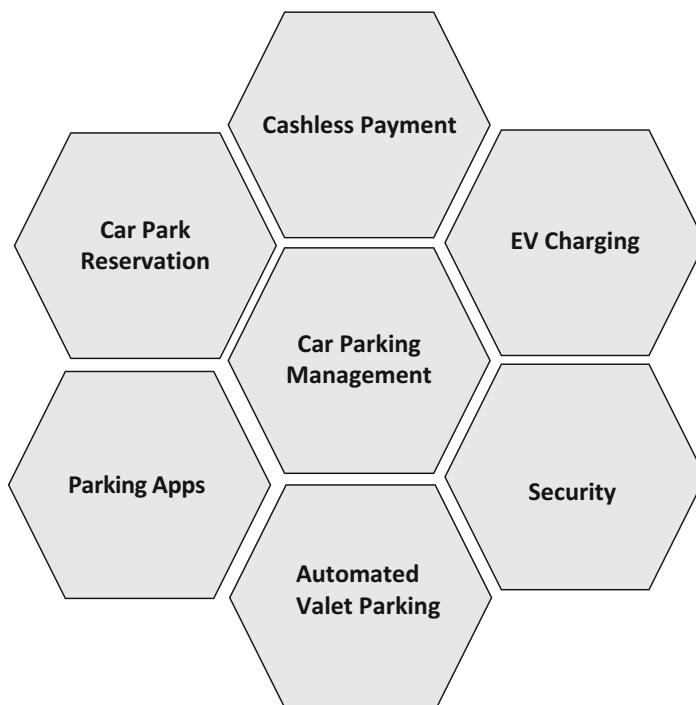


Fig. 10.4 Aspects of connected parking and the parking ecosystem (see also Nicodemus and Auracher 2015)

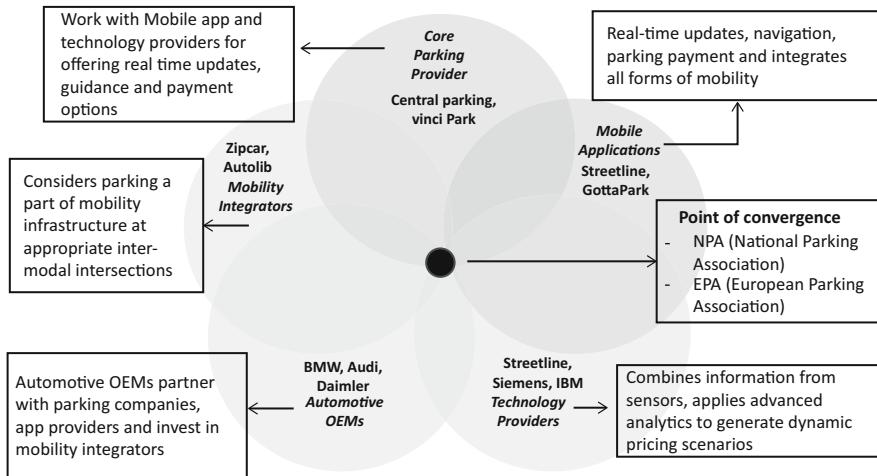


Fig. 10.5 Parking eco system (Chandrasekar et al. 2013)

10.3 Parking Assistance

Parking a car in a narrow parking slot can be challenging and is one of the main reasons for damages and repair work. Parking assistance systems can help. They are available for a couple of years now and especially popular in midrange to premium cars.

One can differentiate between different levels of functionality and complexity. The first and most simple system just alerts the driver of an obstacle via ultrasound sensors. A beep and/or visual signal is given, if the car comes too close to an obstacle in the rear, the front, or the side wall. These systems are particularly helpful in park houses with narrow parking lots. Rear cameras, which are mandatory in the USA since 2016, give additional overview and are helpful in combination with the ultrasound warning system.

The next level of parking assistance provides lateral control of the drive path by steering the car. The right sequence and pattern of steering maneuvers is often difficult. It can, however, be calculated precisely, and a computational algorithm can guide the car in an optimal way. These systems were first introduced in the early 1980s and are now widely available even in compact cars. Steering assistance requires the driver to constantly control the gas pedal and the breaking. The next level also relieves the driver from the longitudinal maneuvering. Steering and backing-off maneuvers are done automatically. However, the driver can apply breaks and overpower the parking assistance systems at any time.

Another level of automation is reached when the driver does not sit in the car anymore but oversees the park maneuver on his or her mobile phone.

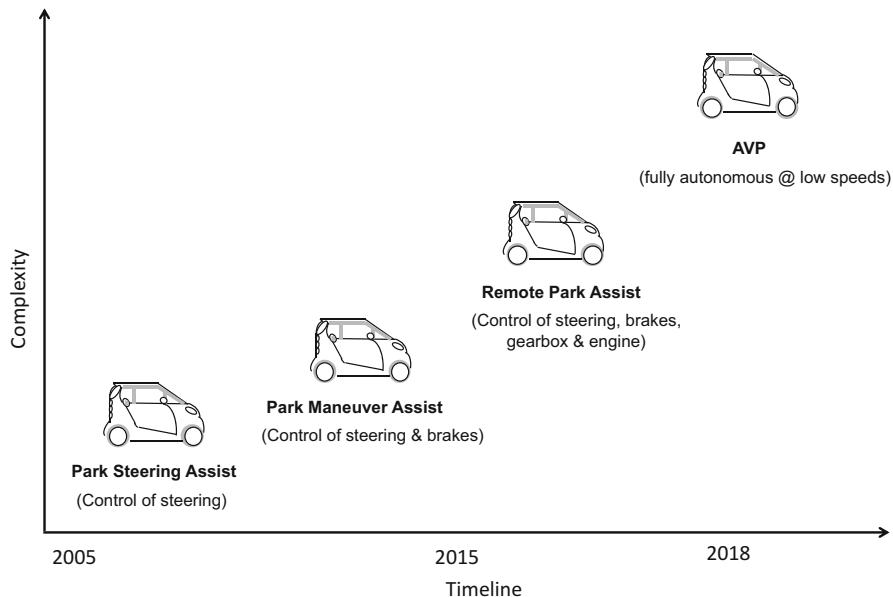


Fig. 10.6 Steps towards fully automated valet parking (see Nicodemus and Auracher 2015)

BMW and Mercedes offer systems for remote parking (Werle 2015). Mercedes uses the smartphone; BMW has developed a special key with an integrated display. As it is still not allowed to let the car drive completely alone, the driver has to oversee this process by constantly pressing a button. Various first-tier suppliers provide the technology, among them Bosch, Conti, Valeo, and ZF/TRW.

The highest level of automation is reached, if the driver only has to initiate the parking process, while the car finds the parking place, drives toward the slot, and maneuvers into the parking space without any further intervention and completely automatically. The concept is called automated valet parking (AVP). This chapter explores it in more detail in the next section. BMW demonstrated a remote parking assistance with AVP capabilities at the CES 2015 in Las Vegas (URL7 2015).

Figure 10.6 shows the evolution of automated valet parking driven by a combination of different sensor systems like ultrasound, radar and cameras.

10.4 Automated Valet Parking

Automated valet parking service (AVP) enables a vehicle to drive and park without any human interaction (Min and Choi 2013). This is most likely one of the first examples of fully autonomous driving being commercialized.

Bosch has introduced an automated valet parking functionality which combines both in-vehicle sensors (ultrasound sensors) and infrastructure-based technology (Gebhardt 2016; URL24 2017; URL25 2017).

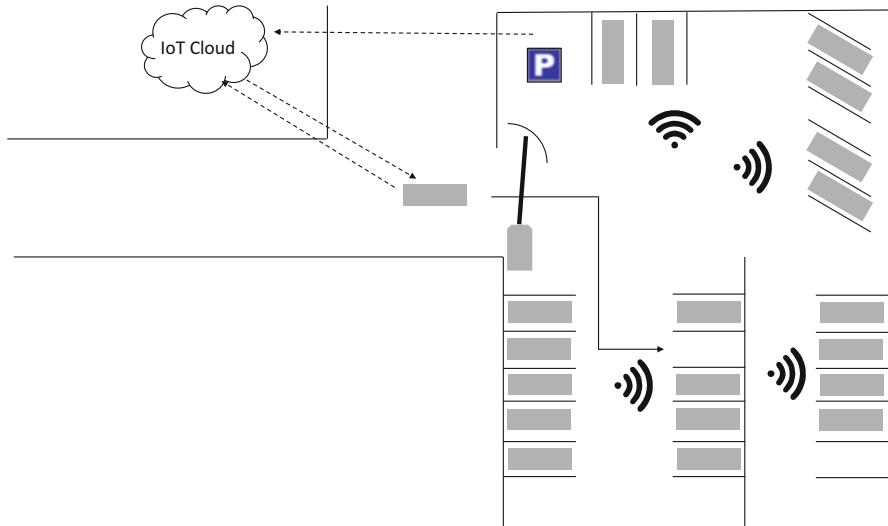


Fig. 10.7 Automated valet parking (AVP) concept of Bosch (see also Nicodemus and Auracher 2015)

The idea is shown in Fig. 10.7 and consists of the following:

- A car can be dropped off at the gate of a parking lot or park house which has been modified to allow AVP.
- The modifications to the parking lot are substantial but only need one time investments.
- The park house uses laser range finders to track cars and sensors in the ground to automatically notify free parking positions.
- The car relies on the onboard parking assistance systems and other ADAS features to control the maneuver (see Chap. 11).
- When the car is identified and allowed to enter the park house, it automatically connects to a WLAN.
- Through this WLAN the car communicates with the park house.
- The car gets the information about where to find a free parking lot and will be guided by the laser range finders and camera systems installed in the parking lot.
- The combination of both onboard sensors and outside sensors (park house infrastructure) increases the safety substantially.
- If the car faces an obstacle, the parking assistance system will automatically stop the car.
- Such a stop command can also come from the parking lot guidance system, if, for example, another car is approaching.
- The valet function will also automatically guide the car from the parking position back to the gate when the driver calls it.

Another AVP solution, called Park4U, is being offered by Valeo (URL33 2017). Park4U relies more on the vehicle sensors and does not need modifications to the parking lot itself. The car is automatically piloted to a free parking space by means of stereo cameras and ultrasound sensors in the vehicle.

The legal framework for automated valet parking, however, still has to be developed fully. Especially the clause of the Vienna Convention on Road Traffic (URL37 2017), that, at any given time the driver needs to be in control (steering, gas pedal, and break), i.e., should be able to step in and drive the car, is a problem. There are multiple initiatives to modify the Vienna protocol to allow for piloted driving. Here, automated valet parking can have a major impact and can be an important step towards full-autonomous driving as the speed of the car is typically low.

Currently, there are various activities in automated valet parking:

- Car2Go and Bosch—cooperation on AVP for carsharing (Gerster 2016; URL24 2017; URL38 2017) as shown in Fig. 10.8.
- BMW Drive Now also plans to introduce AVP. This is based on a hub model, where the car will automatically find back its way. Also, predictive parking algorithms are used to relocate cars.
- Smart city applications and showcases—see Ludwigsburg main station/park house with AVP functionality.
- AutoPles is a research project that demonstrated a proof of concept for combining automated valet parking and automated charging of electric vehicles (URL34 2017). The project partners were Trans Energy Partners, CTC cartech company GmbH, Böblingen, IPT GmbH, Weil am Rhein, Lapp Systems GmbH, Stuttgart, and Research Center for Information Technology (FZI), Karlsruhe.

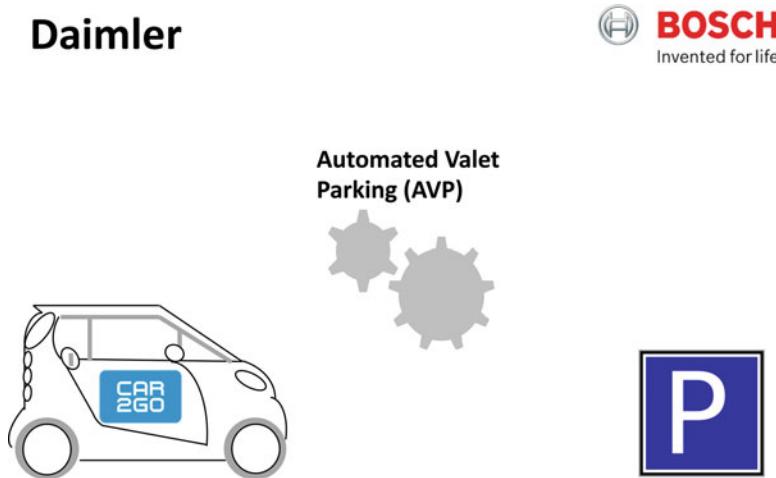


Fig. 10.8 Bosch and Daimler collaborate on automatic valet parking (Gräfe 2016; URL3 2015; URL38 2017)

There are a lot of other pilot projects and research activities going on; however, they are not always visible as automated valet parking is considered to be a highly competitive field, both for OEMs and for x-tiers, alike. A timeline for the introduction of automated valet parking in the overall context of automated, piloted driving, can be found in Freitag (2016).

Carsharing is a particularly promising area for automatic valet parking due to the following reasons:

- Dropping off the car and finding a parking place is a big issue, especially in metropolitan areas.
- Availability of suitable parking places can be the biggest hurdle for efficient use of carsharing.
- AVP allows for up to 20% higher utilization of parking space.
- Combination of parking and charging for shared electric cars is possible (see AutoPles project).

10.5 Cyber Threats

With the digital transformation of the economy, cybercrime, and cybersecurity are topics which make it into the news regularly (Berke 2015; Germis 2016). The cyber threats to connected vehicles are getting more and more attention in the general media, scientific community, and automotive industry (Currie 2015; Gerhager 2016; Greenberg 2013; Lobe 2016; Poulsen 2010; Solon 2015; Stockburger 2016).

Connected parking, remote parking, and automated valet parking solutions are vulnerable to cyberattacks (Chucholowski and Lienkamp 2014). Autonomous features potentially increase the consequences of an attack as the human driver cannot oversee all situations and, if necessary, overpower the machine (Markey 2015; URL30 2017). Cyberattackers could try to steal the car or gain unauthorized access to critical systems. Unauthorized, remote access to the trunk of the vehicle by sending a remote opening command introduces several potential new risks. AVP opens up the possibility of new cyberattack threats mainly due to the need of vehicle-to-infrastructure (V2I) communication with the infrastructure of the park house.

Cyber threats can only be dealt with in a holistic, life cycle-oriented manner that includes HW, SW, and people, like engineers, car owners, dealers, workshop staff, etc. (Besenbruch 2014; Weimerskirch 2016).

The Spy Car Act (Markey 2015; Weimerskirch 2016) has put pressure on the industry to fight against cyber threats and to look at efficient cybersecurity solutions. Quick and efficient responses to cyberattacks and vulnerabilities will be very important in the future (Zetter 2015; Markey 2015), especially with more and more of safety critical functions depending on connectivity (Vembo 2016).

Park house management systems often use a Windows-PC as an industrial park house management system. If the operating system is not patched properly or—worse—support has expired, as in the case of Windows XP, this can be a serious

threat (Haas et al. 2017). Malware could be introduced through the USB port or sent through the network and could compromise the attached subsystems, actuators, and sensors. In such a scenario, vehicles could receive false signals and guidance commands. Also, the in-house position guidance system could be disturbed and rendered useless for path planning.

The major cyber threats to connected parking and automatic valet parking have been outlined in (Haas and Möller 2017):

- Compromising the connection between smartphone/key and vehicle resulting in loss of control of the car, e.g., by man-in-the-middle attacks (Wolf et al. 2016; Wolf and Osterhues 2013).
- Compromising vehicle to infrastructure communication when the car is in transit from the point of dropping it off to the car park or park house.
- Attacks on emergency V2I communication protocols which enable an AVP car to receive commands coming from other top priority traffic participants like ambulance, police, etc. instructing the vehicle to stop and give way. A hacker could exploit this mechanism to gain control.
- The parking management system and car park infrastructure could be hacked by criminal groups trying to gain control of the car while it is out of sight from the owner.
- Attacks could also directly affect the sensors like blinding camera sensors, confusing camera auto control, relaying or spoofing signals, etc.

10.6 Intrusion Detection and Prevention

Intrusion detection systems (IDS) are a mechanism to detect any kind of intrusion into a system. Fallstrand and Lindstrom (2015) define such a system as follows “A system that detects malicious activities, policy violations or other such irregularities in the system and reports them.”

Modern automobiles have sophisticated advanced driver assistance functions like intelligent parking assistance, blind-spot detection, lane departure warning, adaptive cruise control, automatic emergency breaking, navigation systems with real-time updates and many more. It is crucial to ensure that these functions behave the way they are supposed to without any outside interference (Haas et al. 2017; Haas and Möller 2017). An IDS can be deployed to detect abnormal behavior, thereby minimizing the effect of cyberattacks exploiting vulnerabilities and malicious tampering with the system (Scarfone and Mell 2007; Serio and Wollenschläger 2015; Weimerskirch 2016; Wolfsthal and Serio 2015).

10.6.1 Types of Intrusion Detection Systems

There are multiple categories and types of intrusion detection systems (IDSs) (Vestlund 2009). The main categories are:

- *Host-based IDS (HIDS)*: This type of system resides on the host and examines the internal state by reviewing the logs of system calls, modification of files, etc.
- *Network-based IDS (NIDS)*: This type of system examines the data traffic between hosts in the network.
- *Hybrid IDS*: This type of system combines the use of HIDS and NIDS on various nodes and hosts for analysis purposes.

Note, that HIDS and NIDS are meant for specific applications only. They cannot be interchanged with one another. Hybrid systems however can be used for any kind of applications and they combine the advantages of HIDS and NIDS.

There are different ways of detecting an intrusion. These can also be applied to a connected car, as it can be seen as a computer network, both internally (networked ECUs) as well as externally as a network of connected cars, and infrastructure. The most important IDS are:

- *Signature-based IDS*: Uses rules to describe malicious behavior in order to detect an intrusion. It compares sequences of events, and patterns to the rules that it has stored in its existing database. The rules can be applied from single to multiple packets. The database needs to be constantly updated with new signatures for different kinds of intrusions. However, if there is a slight change in pattern, the IDS will most likely ignore it. The biggest problem occurs, if the attack does not contain any kind of signature. It will be not updated in the database; hence, the intrusion will be undetected for subsequent access/intrusion.
 - Advantage: Small attacks containing signatures can be easily detected and thwarted.
 - Disadvantage: Attacks without signature cannot be detected.
- *Anomaly-based IDS*: The system creates different profiles of usage over time. The IDS can examine the observed behavior and compare them with the different profiles created. If there is too much deviation from the profiles, then the system reports an intrusion. This can lead to false-positive alarms, depending upon the aggressiveness of IDS. False positives are events that are reported malicious, but in reality they are completely harmless.
 - Advantage: No pre-defined rules for detection of attacks are required; hence, new attacks can be detected.
 - Disadvantage: False-positive cases can arise, leading to confusion for the users. Establishment of normal profile usage is required, which is hard to achieve.

10.6.2 Attacks Against Connected Cars

Like any connected system, connected cars are vulnerable and face some classical attacks known from computer networks. Also, one has to keep in mind that the ECUs of a modern mid-to-high range car, infotainment units, and specialized ECUs for

advanced driver assistance systems (ADAS) (see also Chap. 11), like parking assistance (PA) and automatic valet parking (AVP) (see also Chap. 10), offer much more computational power than a desktop computer. Therefore, different kinds of cyberattacks are possible, such as:

- *Distributed Denial of Service (DDoS)*: This is one of the most serious attacks. A denial of service would prevent a user from accessing network services by clogging up the system resources, thus reducing the efficiency and performance of the network. In the connected car scenario, an attacker could create a large number of fake identities and could transmit dummy messages to a legitimate car to create a jam in the network. A distributed denial of service uses multiple cars from different locations and time slots to carry out the same attack.

Intrusion detection systems (IDS) can be employed to detect and prevent such attack, by using the anomaly-based method as described in the previous section. The normal utilization of system resources can be setup as a profile, and the behavior of an attack can be monitored against the profile. High deviation would mean that the car is being attacked, and appropriate measures can be taken.

- *Black Hole Attack*. Area where the network traffic is redirected, and there is no subsequent response. The reason could be that there is no node or the node refuses to respond. The attacker's node can fool its neighbors, thus gaining the right to forward the packets. Once the attacker node gets the packet, it can drop it or forward it to an incorrect node. Alheeti et al. (2015a, b) shows how to address this problem by building an intelligent IDS that uses proportional overlapping scores to derive a set of features which describe the normal or abnormal behavior of vehicles. A simple solution includes the packet sequence number in order to identify the packets that have been dropped.
- *Sybil Attack*: Is a very common attack, that occurs by malicious node impersonating them as some legitimate node and then sending wrong messages. In the vehicular context, a vehicle declares to be several other vehicles at the same time or in succession. A vehicle can claim to be in different positions, creating chaos and making the attack very dangerous. It can damage network topologies as well as use a significant amount of network bandwidth (La Vinh and Cavalli 2014). Certain solutions (Xiao et al. 2006) try to detect and localize the position of the Sybil node. They analyze the signal strength distribution and use that to estimate the distance between the Sybil and current node. If the distance measured through signal strength is not the same as being advertised, then the node is likely to be a Sybil node.
- *Bogus Messaging*: This attack can be orchestrated by an attacker or a legitimate user and simply consists of sending false messages in the network. This attack is beyond the scope of IDS. Other cryptographic schemes such as message authentication can be used instead.

- *Timing Attack:* Many functions in a vehicle are time critical (e.g., breaking, powertrain control), which require data transmissions in hard real time. When malicious nodes-/malware-inflicted systems receive a message, they do not forward it immediately but add some time slots to the original message to create a delay. Thus, other subsystems receive the message much later than they were supposed to. In certain cases, this can directly lead to accidents due to the delay of messages. This cannot be detected by signature-based or anomaly-based IDS. Hence, other methods such as data integrity are required to curb such attacks.

10.6.3 Artificial Neural Network-Based IDS Implementation

One way to implement intelligent intrusion detection systems is to use artificial neural networks (ANN) which can be trained to detect and classify malicious activities from the network. The multilayer perception model of an ANN is shown in Fig. 10.9 (see also Chap. 6).

A popular category of an ANN is the multilayer perceptron (MLP for short). These networks consist of nonlinear neurons based on the following activation model (Haykin 2009).

$$y = \sigma\left(\sum_{i=1}^n w_i x_i\right) = \sigma(\mathbf{w}^T \mathbf{x})$$

where $x_i \in \mathbb{R}^n$ represents an n -dimensional input vector, which is multiplied with weight factors w_i and mapped to an activation level in the range $[0,1] \subset \mathbb{R}$ by a nonlinear activation function $\sigma(\cdot)$.

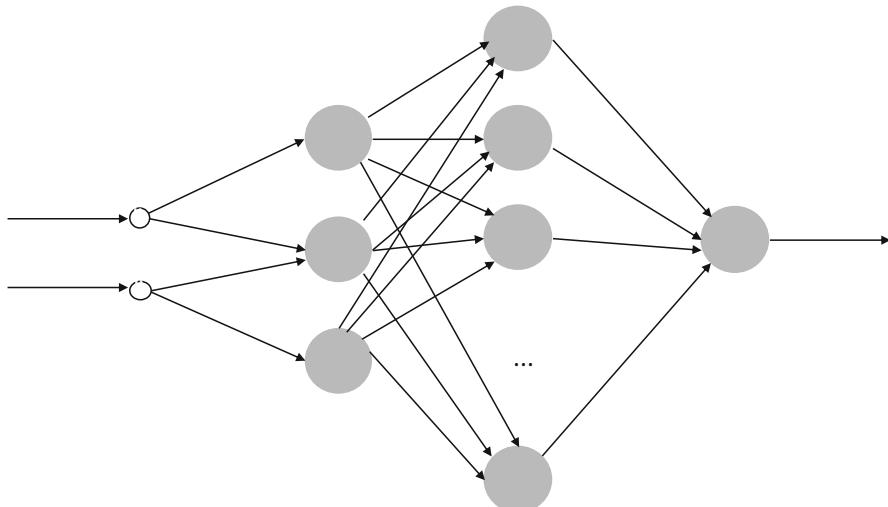


Fig. 10.9 Multilayer perceptron ANN model

The neurons form a multilayer structure, in which the signals of the input neurons are being propagated forward layer by layer. The MLP belongs to the class of feedforward nets.

A difficult design decision is the choice of network layers. While a proof exists that the universal approximation property can be achieved by just one layer, this is only an existence theorem (Haykin 2008). In practice, however, the choice of more layers often leads to more compact and smaller networks.

The first stage in designing ANN-based IDS is the data gathering and pre-processing part. Thus, all incoming data is collected, transformed, and normalized to standard units as illustrated in Fig. 10.10. The next step would be to extract features from this data. Features are characteristics of the data stream that can be measured such as the number of packets transferred (on the vehicle bus systems, between two vehicles, between vehicle and infrastructure, etc.), delay in transfer of packets, number of dropped packets, etc. Other features could be the information in the header of the packets including time-to-live, payload size and type, source and destination MAC, IP address, port, and so on.

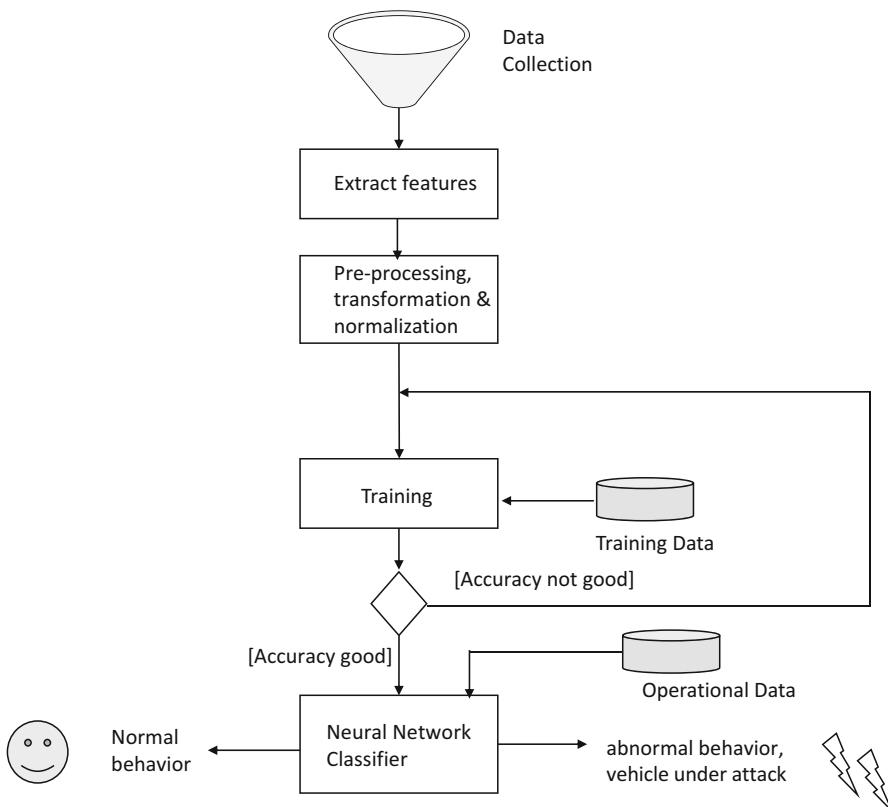


Fig. 10.10 Training of an ANN-based Intrusion Detection System (IDS) (Haas and Möller 2017; Alheeti et al. 2015a, b)

Finally, the trained ANN should be able to recognize and classify the data packets and control messages in the network in real-time as shown in Fig. 10.10. If malicious or abnormal behavior is detected, an alarm can be generated and reported (Alheeti et al. 2015b).

Many automotive manufacturers have taken up the challenge and are currently evaluating sophisticated intrusion detection systems, based on nonlinear classification schemes (like the neural network approach described above) and machine learning algorithms (see Fig. 10.10).

Several companies already offer IDS as commercially available automotive cybersecurity solutions (Serio and Wollenschläger 2015; Weimerskirch 2016; URL1 2015; URL4 2015; URL30 2017; URL36 2017):

- Samsung/Harman/TowerSec
- Continental/Argus
- Bosch/ETAS/ESCRYPT GmbH
- Cisco
- Honeywell
- IBM
- McAfee (formerly with Intel)
- Symantec
- Trilliu

Many first-tier suppliers have strengthened their cybersecurity capabilities through acquisitions, for example, Harman by acquiring TowerSec in 2016.

Figure 10.11 illustrates how an intrusion detection and prevention system (IDS/PS or IDPS for short) can be embedded into the E/E architecture of a modern vehicle. The telematic control unit (TCU) and the central gateway are

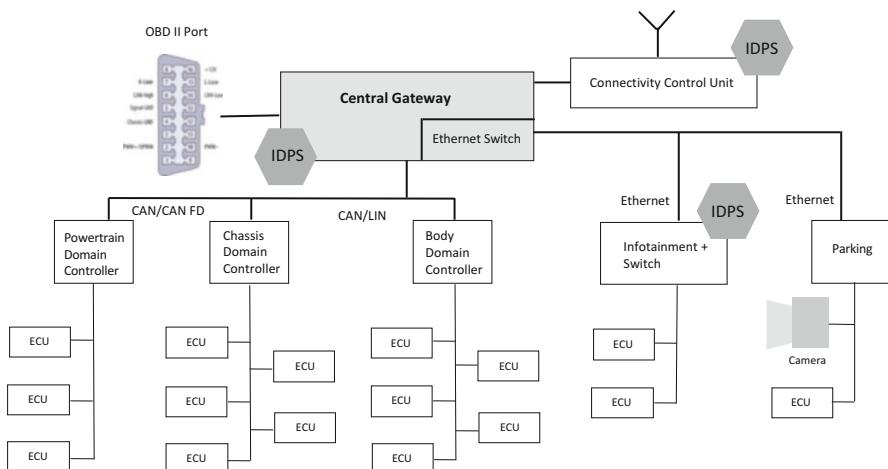


Fig. 10.11 Different options to integrate an IDPS into the E/E architecture and topology of a modern vehicle

straightforward choices for the IDPS, but also the OBD II port, critical ECUs, and high-speed bus systems, for example, Ethernet-based systems for ADAS functionality, can be good choices.

Automotive IDPS should have characteristics like real-time detection, low resource consumption, and continuous data processing. Also, a perfect balance between the computing and memory resources and the performance, detection capabilities, false positives, learning, cost, and quick response capabilities has to be found.

10.7 Conclusion and Recommended Readings

Parking is a difficult and tedious task. The search for available parking space accounts for a major part of inner-city traffic and can lead to a lot of frustration. Accidents while parking are a major cause of damage repair work.

The digital transformation affects the classical business of parking in a major way. Connectivity can help to identify available parking spaces upfront and often in real-time. Many apps are available in the market that allow for finding parking space, booking, and cashless billing; some of them communicate with the infrastructure automatically (opening gates and turnpikes). One can differentiate between on-street parking and off-street parking, e.g., parking lots at airports, shopping malls, or park-and-ride facilities. Parking assistance systems help to maneuver the car even into narrow parking lots. The first generation of systems could only warn, when a car came too close to obstacles; later systems were capable of controlling the lateral position of the car, the current generation of systems can also control longitudinal movements, and the driver just supervises the process. The highest level of automation is automated valet parking which allows the driver to drop off the car, while the car automatically enters the car park and finds a free parking lot. Such systems heavily rely on sensors and car-to-infrastructure communication while the speed is limited, typically to a max of 6 km/h.

10.7.1 Cyber Threats and Cybersecurity

Connected parking exposes many attack surfaces and is prone to cybersecurity threats. This chapter gave an overview of the main problems and discussed a solution based on intrusion detection and prevention. Modern vehicles are fully connected and are prone to cyberattacks as they expose a complex attack surface (see also Chap. 6). Intrusion detection systems can help to detect attacks by filtering the data streams of the connected car and classifying the data into normal or abnormal (i.e., potentially malicious).

This chapter has given a brief overview of the principle of intrusion detection, and we have discussed how these systems can be used to prevent cyberattacks on cars. Intrusion detection systems are based on nonlinear pattern recognition methods. A

popular and well-known way to implement them is to use machine learning, artificial neural networks and deep neural networks (DNNs), as explained in Chap. 6.

10.7.2 Recommended Readings

Balani (2015) gives a good overview of the general concept of an IoT cloud and discusses the IoT solutions of Microsoft Azure, GE's Predix IoT cloud, and Amazon's AWS IoT cloud in detail. Mahaffey (2015a, b) emphasizes the importance of a strong collaboration between cybersecurity researchers, automotive OEMs, and technology firms. He sees Tesla as an excellent example of how this collaboration can help to quickly respond to new security threats. Miller and Valasek (2014, 2015) give an in-depth analysis of vulnerabilities and cyberattacks. Miller and Valasek both work for Uber now. Pickhard et al. (2015) emphasize how important it is to focus on data security already in early phases of development. Cryptography plays a key role to secure the communication between cars, infrastructure, and the driver's smartphone (Wolf et al. 2016). Reuss et al. (2015) discuss the synergies between autonomous driving and e-mobility. The integration of AVP and charging will be an important function for electric vehicles in the future. Vembo (2016) gives an overview of the challenges and architecture of connected cars.

In URL1 (2015) a good overview of the key problems in automotive cybersecurity is given. The authors discuss best practices on how to prevent and mitigate cyberattacks. URL2 (2015) looks into the future and poses the question if the steering wheel will be needed at all.

Apart from the parking apps and solution providers discussed in this chapter, there are several others with interesting concepts, for example, ParkJockey (URL6 2017) and EasyPark (URL17 2017).

The IAA auto show in 2017, like in 2015, organized a special exhibition on new mobility. This exhibition provided a platform for start-ups and technology specialists in connected parking (URL28 2017).

10.8 Exercises

What is meant by the term *parking industry*?

Describe the main characteristics of the parking industry.

What is meant by the term *connected parking*?

Describe the main characteristics of connected parking.

What is meant by the term *community-based parking*?

Describe the main characteristics of community-based parking.

What features do you expect from a *connected parking app*?

Describe the main features of a connected parking app.

What methods are being used to *detect a free parking lot*?

Describe the most relevant method to detect a free parking lot.

What is meant by the terms *on-street* and *off-street parking*?

Describe the main difference between on-street and off-street parking.

What is meant by *predictive parking*?

Give an example for predictive parking.

What are the *parking assistance systems*?

Describe the main features used.

What *sensors are typically being used for parking assistance systems*?

Describe the sensor types used and give an example for the system.

What is meant by the term *remote parking*?

Describe the characteristics, benefits, and challenges of remote parking.

Explain why BMW requires *drivers to constantly press a key while the car parks remotely*.

Explain how the *parking assistance system deals with obstacles in the driving path*.

Explain the *difference between remote parking and automated valet parking*.

What *research initiatives and proofs of concept in AVP are you aware of*?

Describe the main research initiatives and proof of concept used.

What will be the *acceptance of AVP*?

Explain whether there are challenges or not for OEMs to convince their customers to give up the control of the parking process.

Will people *trust technology to park the vehicle*?

Will there be *new players and start-ups entering the AVP market*?

Characterize the new players.

What *demand in the market do you see for AVP*?

Explain your thoughts.

In what *regions and markets will AVP be introduced first*?

Which *markets will follow and what is the timeline*?

How much of a *competitive differentiation will AVP be for OEMs in the future*?

Give an example-based ratio.

What is the *evolutionary path from parking assist to AVP*?

Describe the commonalities and the differences.

What *new business models could emerge for AVP*?

Answer the question w.r.t. funding, location aware services, and parking infrastructure.

What *role will AVP play for carsharing, ridesharing, and electric cars*?

Give examples w.r.t. who will cooperate with whom, OEM, suppliers, and service providers.

What are the *biggest hurdles for introducing AVP from a technology viewpoint*?

Give an example and explain the example in detail.

What *sensors will be used for AVP*?

Describe the sensor types used and give an example for the system.

What is the *relationship of AVP sensors with ADAS and autonomous/piloted driving*?

Give an example and explain the example in detail.

Which *guidelines for the development of semiautonomous vehicles, official standards, and regulations do apply to autonomous parking technologies*?

Give an example and explain the example in detail.

How does the potentially more *predictable environment (light, weather, etc.) in a car park influence the required HW/SW?*

Give an example for the HW and SW requirements and explain the example in detail.

Does the car need *access to special sensors inside the parking structure?*

Give an example for the type of sensors and explain the example in detail.

What would be the *cost to set up a parking infrastructure?*

Give your thoughts and explain them.

How will the *car park's management system communicate with the car in order to find and allocate a free parking space and transmit the route to the car. Is this done using GPS or dedicated system/app or Wi-Fi, and how will one deal with weak signals?*

Describe a scenario and explain the chosen constraints.

How does the *human-machine interface (HMI) for AVP look like?*

Describe the scenarios for apps on smartphone, smartwatch, and others.

What are the *biggest obstacles for AVP from a legal viewpoint?*

Describe your thoughts and explain them, and comment on product liability issues in AVP.

What *legal framework is applicable for AVP?*

Describe your thoughts and explain them.

What *modifications are needed in the future?*

Describe your thoughts and explain them.

Can AVP have an *impact on car insurance models?*

Describe your thoughts and explain them.

What kind of *objects can block the path of an AVP system and how can it recognize these objects and avoid accidents in such situations?*

Describe scenarios for another vehicle, a pedestrian, a bicycle, a kid, and an animal.

What are the *cybersecurity concerns in connected parking?*

Describe how an AVP system can be attacked.

What *cybersecurity threats does a connected car face?*

Describe possible attack scenarios.

Why are *connected cars interesting to cybercriminals?*

Describe your thoughts and explain them.

What is meant by the term *intrusion detection system?*

Describe the main characteristics of an IDS and how it can be implemented.

What is meant by the term *artificial neural network?*

Describe the basic principle of an ANN and how the training process can be done.

What other *nonlinear pattern recognition methodologies are you aware of?*

Give examples and describe them in detail.

How to *implement an IDS with a ANN?*

Describe your thoughts and explain them.

What *commercial solutions for IDS are available, and what are the limitations?*

Give examples and describe them in detail.

How does an *automotive IDS differ from a classical computer network IDS?*

- Describe your thoughts and explain them.
- What *impact will AVP have on the car parking space, especially in city areas?*
- Describe your thoughts and explain them.
- What role does *AVP play in smart cities?*
- Describe your thoughts and explain them.
- Are there *social implications of AVP, and how can these implications be compared to the social impact autonomous driving will have?*
- Describe your thoughts and explain them.

References and Further Reading

- (Alheeti et al. 2015a) Alheeti, K. M. A., Gruebler, A., McDonald-Maier, K. D.: An intrusion detection system against malicious attacks on the communication network of driverless cars. In: Proceedings 12th Annual IEEE Consumer Communications and Networking Conference (CCNC), pp. 916-921, 2015
- (Alheeti et al. 2015b) Alheeti, K. M. A., Gruebler, A., McDonald-Maier, K. D.: An Intrusion Detection System against Black Hole Attacks on the Communication Network of Self-Driving Cars. In: Proceedings 6th International Conference on Emerging Security Technologies (EST), pp. 86-91, 2015
- (Balani 2015) Balani, N.: Enterprise IoT – A Definite Handbook, Self-published, Kindle Edition, 2016
- (Brisbourne 2014) Brisbourne, A.: Tesla's Over-the-Air Fix: Best Example Yet of the Internet of Things? Wired online. February 2014. Available from: <http://www.wired.com/insights/2014/02/teslas-air-fix-best-example-yet-internet-things/>
- (Berke 2015) Berke, J.: Hacker attacks on companies – When cyberattacks lead to bankruptcy (in German). Wirtschaftswoche online. November 25th 2015. Available from: <http://www.wiwo.de/unternehmen/it/hackerangriffe-aufunternehmen-wenn-cyberattacken-in-den-bankrott-fuehren/12632916.html>
- (Besenbruch 2014) Besenbruch, D.: Electronic Systems – Protection against Manipulation (in German), ATZ elektronik, 7/2014
- (Chandrasekar et al. 2013) Chandrasekar, P, Barua, N, Zia, Y.: Future of Vehicle Parking Management Systems in North America and Europe. Frost & Sullivan. October 1st 2013. Available from: <https://de.slideshare.net/FrostandSullivan/parking-management-26752963>
- (Chucholowski and Lienkamp 2014) Chucholowski, F., Lienkamp, M.: Teleoperated Driving – Secure and Robust Data Connections (in German). ATZ elektronik, 01/2014
- (Currie 2015) Currie, R.: Developments in Car Hacking, December 5th 2015. SANS Institute. Available from: <https://www.sans.org/reading-room/whitepapers/internet/developments-car-hacking-36607>
- (Dierig 2012) Online Parking is too cheap in Germany (in German). Welt online. October 8th 2012. Available from: <http://www.welt.de/wirtschaft/article109690967/Parken-ist-in-Deutschland-viel-zu-billig.html>
- (Fallstrand and Lindstrom 2015) Fallstrand, D., Lindstrom, V.: Automotive IDPS: Applicability analysis of intrusion detection and prevention in automotive systems. Master' Thesis. Chalmers University of Technology. Available from: <http://publications.lib.chalmers.se/records/fulltext/219075/219075.pdf>
- (Freitag 2016) Freitag, M.: Robotic Cars - German Manufacturers in Pole Position (in German). July 26th 2016. Available from: <https://www.manager-magazin.de/unternehmen/autoindustrie/roboterautos-deutsche-autobauer-fuehrena-1104783.html>
- (Gebhardt 2016) Gebhardt, M.: This is how we park tomorrow (in German). Zeit online. May 10th 2016. Available from: <https://www.zeit.de/mobilitaet/2016-04/autonomes-fahren-parken-bosch>

- (Gerhager 2016) Gerhager, S.: Why auto makers might soon get into the focus of blackmailers (in German). Focus online. October 17th 2016. Available from: http://www.focus.de/auto/experten/autowirtschaft-warum-autohersteller-fokus-von-erpressern-geraten-koennte_id_6081085.html
- (Gräfe 2016) Bosch and Daimler rely on automatic parking searches (in German). Stuttgarter Nachrichten online. March 10th 2016. Available from: <http://www.stuttgarter-nachrichten.de/inhalt.stuttgart-bosch-und-daimler-setzen-aufautomatische-parkplatzsuche.6cf6485f-67e5-47f3-817f-8acd5d12e707.html>
- (Gerster 2016) Assistance systems: Bosch drives automated parking (in German). Automobilwoche, March 2016
- (Greenberg 2013) Greenberg, A.: Hackers reveal nasty new car attacks-with me behind the wheel. Forbes online. July 24th 2013. Available from: <https://www.forbes.com/sites/andygreenberg/2013/07/24/hackers-reveal-nasty-new-car-attacks-with-me-behind-the-wheel-video/#64771b28228c>
- (Germis 2016) Germis, C.: Each week 6000 attacks from the Internet against VW (in German). FAZ online. August 18th 2016. Available from: http://www.faz.net/aktuell/wirtschaft/unternehmen/jede-woche-6000-cyberangriffe-gegen-vw-14393188-p2.html#/pageIndex_2.8
- (Haas et al. 2017) Haas, R., Möller, D., Bansal, P., Ghosh, R., Bhat, S.: Intrusion Detection in Connected Cars. In: Proceed. IEEE/EIT 2017 Conference, pp. 516-519. Ed.: Izadian, A., Catalog No. CFP17EIT-USB 978-1-5090-4766-6, 2017
- (Haas and Möller 2017) Haas, R., Möller, D.: Automotive Connectivity, Cyber Attack Scenarios and Automotive Cyber Security. In: Proceed. IEEE/EIT 2017 Conference, pp. 635-639. Ed.: Izadian, A., Catalog No. CFP17EIT-USB. 978-1-5090-4766-6, 2017
- (Haykin 2009) Haykin, S.: Neural Network and Learning Machines. 3rd edition. Pearson Education, 2009
- (Jungwirth 2016) Presentation of Johann Jungwirth and personal discussion at the CeBIT 2017, Hannover, March 2017
- (La Vinh and Cavalli 2014) La Vinh, H., Cavalli, A. R.: Security attacks and solutions in vehicular ad hoc networks: a survey. In: International Journal on AdHoc Networking Systems (IJANS), Vol 4, No. 16, pp. 1-20, 2014
- (Laudon et al. 2010) Laudon, K., Laudon, J., Dass, R.: Management Information Systems, Pearson Publ., 2010
- (Lobe 2016) Lobe, A.: Hacker Alert – In a modern car today are computers and info systems that are easy to manipulate. How do the manufacturers deal with the security gap? (in German). Zeit online. August 25th 2016. Available from: <http://www.zeit.de/2016/34/elektroautos-steuerung-hacker-gefahr-sicherheit-hersteller>
- (Mahaffey 2015a) Mahaffey, K.: The New Assembly Line: 3 Best Practices for Building (secure) Connected Cars. Lookout Blog. August 6th 2015. Available from: <https://blog.lookout.com/tesla-research>
- (Mahaffey 2015b) Mahaffey, K.: Here Is How To Address Car Hacking Threats. TechCrunch. September 13th 2015. Available from: <https://techcrunch.com/2015/09/12/to-protect-cars-from-cyber-attacks-a-call-for-action/>
- (Markey 2015) Markey, E.J.: Tracking and Hacking: Security and Privacy Gaps Put American Drivers at Risk. 2015. Available from: https://www.markey.senate.gov/imo/media/doc/2015-02-06_MarkeyReport-Tracking_Hacking_CarSecurity%202.pdf
- (Miller and Valasek 2014) Miller C., Valasek C.: A Survey of Remote Automotive Attack Surfaces. IOActive 2014. Available from: https://www.ioactive.com/pdfs/IOActive_Remote_Attack_Surfaces.pdf
- (Miller and Valasek 2015) Miller, C., Valasek, C.: Remote exploitation of an unaltered passenger vehicle. August 10th 2015. Available from: <http://illmatics.com/Remote%20Car%20Hacking.pdf>
- (Min and Choi 2013) Min, K-W., Choi, J-D.: Design and implementation of autonomous vehicle valet parking system. In: Proceedings 16th International IEEE Conference on Intelligent Transportation Systems – (ITSC 2013), 2013

- (Nicodemus and Auracher 2015) Connected Parking. EPoSS Workshop on Smart Systems Integration. June 19th 2015. London. Available from: http://www.ivu-bw.de/pdfs/2015/1/DEKRA_ConnectedParking_2015-05-19_Download.pdf
- (Pickhard et al. 2015) Pickhard, F., Emele, M., Burton, S., Wollinger, T.: New thinking for safely networked vehicles (in German). ATZ elektronik, 7/2015
- (Poulsen 2010) Poulsen, K.: Hacker disables more than 100 cars remotely. Wired online. March 17th 2010. Available from: www.wired.com/threatlevel/2010/03/hacker-bricks-cars
- (Rees 2016) Rees, J.: Mobility – Never have to park yourself (in German). Wiwo online. May 6th 2016. Available from: <https://www.wiwo.de/technologie/mobilitaet/mobilitaet-nie-mehr-selber-einparken-muessen/13529696.html>
- (Reuss et al. 2015) Reuss, H.-C., Meyer, G., Meurer, M.: Roadmap 2030 Synergies of Electromobility and Automated Driving (in German). ATZ Elektronik, 7/2015
- (Scarfone and Mell 2007) Scarfone, K., Mell, P.: Guide to Intrusion Detection and Prevention Systems (IDPS). NIST. February 20th 2007. Available from: <https://www.nist.gov/publications/guide-intrusion-detection-and-prevention-systems-idps>
- (Serio and Wollschläger 2015) Serio, G., Wollschläger, D.: Networked Automotive Defense Strategies in the Fight against Cyberattacks (in German). ATZ elektronik, 06/2015
- (Solon 2015) Solon, O.: From Car-Jacking to Car-Hacking: How Vehicles Became Targets For Cybercriminals. August 4th 2015. Bloomberg online. Available from: <https://www.bloomberg.com/news/articles/2015-08-04/hackers-force-carmakers-to-boost-security-for-driverless-era>
- (Stockburger 2016) Stockburger, C.: IT security of cars: You have no choice but to trust the manufacturers (in German). Spiegel online. November 1st 2016. Available from: <http://www.spiegel.de/auto/aktuell/hacker-angriffe-man-hatkeine-andere-wahl-als-den-autoherstellern-zutrauen-a-1092224.html>
- (Vembo 2016) Vembo, D.: Connected Cars – Architecture, Challenges and Way Forward. Whitepaper Sasken Communication Technologies Pvt. Ltd. 2016. Available from: <https://www.sasken.com/insights/white-papers/connected-cars---architecture-challenges-and-way-forward-0>
- (Vestlund 2009) Vestlund, C.: Intrusion Detection Systems in Networked Embedded Systems. Linköping University. Available from: <https://pdfs.semanticscholar.org/10f9/455dde5674de051ae065f358b922cf8bec0f.pdf>
- (Weimerskirch 2016) Weimerskirch, A.: Cybersecurity for Networked and Automated Vehicles (in German). ATZ elektronik, 03/2016
- (Werle 2015) Werle, K.: World in digital change – the game changer – BMW smartphone on wheels (in German). Manager Magazin. November 23rd 2015. Available from: <http://www.managermagazin.de/unternehmen/artikel/gamechanger-bmw-sieger-in-wettbewerb-von-bain-und-mm-a-1063812.html>
- (Wolf and Osterhues 2013) Wolf, M., Osterhues, A.: Secure Messages – Modern Cryptography for Protecting Control Devices (in German). ATZ elektronik, 02/2013
- (Wolf et al. 2016) Wolf, A., Greiff, S., Obermaier, R.: Vehicle access systems of tomorrow (in German). ATZ Elektronik 03/2016
- (Wolfsthal and Serio 2015) Wolfsthal, Y., Serio, G.: Made in IBM Labs: Solution for Detecting Cyber Intrusion to Connected Vehicles, Part I. Available from: <https://securityintelligence.com/made-in-ibm-labs-solution-for-detecting-cyber-intrusions-to-connected-vehicles-part-i/>
- (Xiao et al. 2006) Xiao, B., Yu, B., Gao, C.: Detection and localization of Sybil nodes in VANETs, in DIWANS 06, Los Angeles, CA, pp. 1–8, 2006.
- (Zetter 2015) Zetter, K.: Researchers Hacked A Model S, But Tesla's Already Released A Patch. Wired online. August 6th 2015. Available from: <https://www.wired.com/2015/08/researchers-hacked-model-s-teslas-already/>

Links

2014

(URL1 2014) https://www.cisco.com/c/dam/en_us/solutions/industries/docs/parking_aag_final.pdf

2015

- (URL1 2015) <https://www.mcafee.com/enterprise/en-us/assets/white-papers/wp-automotive-security.pdf>
- (URL2 2015) https://www.wiwo.de/unternehmen/auto/digitalisierung-der-autoindustrie-kuenftig-braucht-man-das-lenkrad-nicht-mehr/v_detail_tab_print/11602152.html
- (URL3 2015) <https://www.bosch-presse.de/pressportal/de/en/bosch-and-daimler-automate-parking-mercedes-with-built-in-valet-42989.html>
- (URL4 2015) <https://www.symantec.com/content/dam/symantec/docs/white-papers/building-comprehensive-security-into-cars-en.pdf>
- (URL5 2015) <https://www.congress.gov/bill/114th-congress/senate-bill/1806/all-info>
- (URL6 2015) <https://www.cisco.com/c/en/us/solutions/industries/smart-connected-communities/city-parking.html>
- (URL7 2015) <https://www.digitaltrends.com/cars/bmw-automated-parking-technology-ces-2015/>
- (URL8 2015) <http://www.theiet.org/sectors/transport/documents/automotive-cs.cfm>
- (URL9 2015) <https://newmobility.world/de/>

2016

(URL1 2016) <http://news.sap.com/sap-iot-seeks-better-parking-with-new-solution/>

2017

- (URL1 2017) <https://en.wikipedia.org/wiki/Parking>
- (URL2 2017) <https://en.parkopedia.com>
- (URL3 2017) <https://parkpocket.com>
- (URL4 2017) <https://www.streetline.com>
- (URL5 2017) <http://inrix.com>
- (URL6 2017) <https://www.parkjockey.com/>
- (URL7 2017) <https://www.tomtom.com/>
- (URL8 2017) <https://spothero.com/>
- (URL9 2017) <https://www.parkwhiz.com/>
- (URL10 2017) <https://www.parkingpanda.com/>
- (URL11 2017) www.bestparking.com/
- (URL12 2017) <https://www.parkme.com/>
- (URL13 2017) <https://www.park-now.com/>
- (URL14 2017) www.gottapark.com/
- (URL15 2017) <https://www.justpark.com/>
- (URL16 2017) <https://www.ampido.com/>
- (URL17 2017) <https://easyparkgroup.com>
- (URL18 2017) <http://parknav.com>

- (URL19 2017) <https://www.cleerciti.com/>
- (URL20 2017) <https://www.park1.com>
- (URL21 2017) <https://apcoa.com>
- (URL22 2017) <https://www.q-park.com>
- (URL23 2017) <http://www.contipark.de/de-DE/>
- (URL24 2017) <http://www.bosch-mobility-solutions.com/en/highlights/connected-mobility/connected-and-automated-parking/>
- (URL25 2017) <https://www.bosch.com/>
- (URL26 2017) <https://www.bosch.com/explore-and-experience/connected-parking-success-factor-development/>
- (URL27 2017) <https://www.bosch-iot-suite.com/>
- (URL28 2017) <https://www.iaa.de/>
- (URL29 2017) <http://park-here.eu>
- (URL30 2017) <https://argus-sec.com>
- (URL31 2017) <http://www.mobility.siemens.com/mobility/global/en/urban-mobility/road-solutions/integrated-smart-parking-solution/pages/integrated-smart-parking-solution.aspx>
- (URL32 2017) <https://en.wikipedia.org/wiki/INRIX>
- (URL33 2017) <http://www.valeo.com/en/park4u-automated-parking/>
- (URL34 2017) <http://www.emobil-sw.de/en/activities-en/current-projects/project-details/autoplets-automated-parking-and-charging-of-electric-vehicle-systems.html>
- (URL35 2017) <https://www.slideshare.net/FrostandSullivan/parking-management-26752963>
- (URL36 2017) <http://www.trillium.co.jp>
- (URL37 2017) https://en.wikipedia.org/wiki/Vienna_Convention_on_Road_Traffic
- (URL38 2017) <http://www.bosch-presse.de/pressportal/de/de/bosch-und-daimler-zeigen-fahrerloses-parken-im-realnen-verkehr-116096.html>



Advanced Driver Assistance Systems and Autonomous Driving

11

This chapter discusses advanced driver assistance systems (ADAS) and autonomous driving. ADAS are systems to help the driver in the driving process. When designed with a safe human-machine interface (HMI), they could increase vehicle safety and more in general road safety. Autonomous driving is based on increasing vehicle automation that leads to autonomous or self-driving or semiautonomous vehicles. Self-driving vehicles are one of the major drivers of change in the automotive industry. This has been discussed in Chap. 2, showing how major OEMs responded, for example, the creation of the CASE organization in Daimler, the introduction of Chief Digital Officer (CDO) positions in BMW and VW, and others. Section 11.1 builds on the introductory treatment of ADAS in Chap. 4 and gives an overview of commercial ADAS functionalities and the sensors being used. Section 11.2 gives a quick recap of the main ADAS functions like lane keeping, lane departure warning, and others in more detail. The section refers to the situation of objects moving across, either in front or behind, the vehicle and advanced methods for pedestrian and object detection.

ADASs are part of active safety initiatives and have been able to drive down the number of fatal accidents (see Chap. 2). Camera-based ADAS systems need sophisticated image processing and analysis algorithms, for example, image processing for lane keep assistance with preprocessing, edge detection, and line segmentation. Therefore, Sect. 11.3 discusses the basic principles of image processing and important algorithms for this vast topic and also shows, how MATLAB and Simulink can be used for rapid prototyping of camera-based ADAS functionality by using the Image Processing Toolbox.

Section 11.4 takes a birds-eye look at the transition from ADAS to autonomous driving summarizing the essential steps in a mindmap, while Sect. 11.5 gives a quick overview of the legal framework and liability issues for autonomous driving.

In Sect. 11.6 a typical ADAS software architecture is shown and various middleware technologies that are currently being evaluated for use in the design of autonomous cars. One of these middleware technologies is SOME/IP, which was developed by BMW and is also available as a standard AUTOSAR implementation. Autonomous cars will not only rely on onboard sensors but need information from infrastructure, maps, and other vehicles. This defines a complex attack surface for cyber threats which is analyzed in detail in the Sect. 11.7. The focus is on cyber threats and functional safety. Cybersecurity solutions like the ones discussed in the previous chapters can be deployed to secure autonomous driving. Section 11.8 wraps up with a summary, and recommended readings, while Sect. 11.9 contains a comprehensive set of questions on advanced driver assistance systems and autonomous driving. Finally, the last section includes references and suggestions for further reading.

11.1 Advanced Driver Assistance Systems

Active and passive safety is an area of intense research and continues to be a field where automakers can differentiate themselves from each other. Active safety includes brake assistance, traction control, electronic stability programs, and ADAS, while passive safety systems include seat belts, airbags, crashworthiness, and so forth. Active safety systems reduce the probability of accidents and sometimes even avoid accidents altogether, while passive safety systems help to reduce the impact on the human passengers. With rapid advances in the field of sensors, mechatronics, and computer vision, driver assistance functionalities like:

- Adaptive cruise control (ACC)
- Speed limit assistance
- Blind spot detection (BSD)
- Driver monitoring and drowsiness detection systems
- Emergency brake assistance
- Intelligent headlamp control
- Intelligent parking assistance
- Lane departure warning
- Night vision
- Obstacle and pedestrian detection
- Traffic sign recognition, etc.

have become affordable even in the midrange segment of cars.

Chapter 11 discusses parking assistance systems as one of the popular and widely deployed driver assistance functions. Figures 11.1 and 11.2 illustrate such a ParkPilot functionality in a modern car, a current version of the Seat Leon. Sensors monitor the front and back drive path and alert the driver if the car comes too close to an obstacle. This is a very helpful feature as visibility is often constrained, and many obstacles are not clearly visible from the driver's seat.



Fig. 11.1 ParkPilot ADAS function in a Seat Leon—front view



Fig. 11.2 ParkPilot ADAS function in a Seat Leon —side and rear view

Advanced driver assistance systems cover a wide range of application scenarios. The mindmap of Fig. 11.4 differentiates between vehicle support and driver support systems. The latter systems provide valuable information to the driver, improve perception, and detect critical conditions that affect the driver's performance. Typical systems in this category of ADAS are (see also Chaps. 4 and 5):

- *In-vehicle navigation system*: These systems guide the driver visually and acoustically (text-to-speech) based on built-in maps and sophisticated route planning. A satellite navigation system provides autonomous geo-spatial positioning with global coverage. It allows small electronic receivers to determine their location (longitude, latitude, and altitude) to within a few meters using time signals transmitted along a line-of-sight by radio from satellites.
- *Drowsiness detection*: Monitors the driver by evaluating different parameters, e.g., steering movements and alerts if a critical level of tiredness has been reached. A tired driver might dose off for a few seconds which can lead to severe accidents on the highway.
- *Automotive night vision*: It is a system to increase a vehicle driver's perception and seeing distance in darkness or poor weather beyond the reach of the vehicle's headlights. Night vision systems are currently offered as optional equipment on certain premium vehicles.

Vehicle support systems include:

- *Adaptive cruise control (ACC)*: Adaptive cruise control (also called radar cruise control) is an optional cruise control system for road vehicles that automatically adjusts the vehicle speed to maintain a safe distance from vehicles ahead. In its basic configuration, it makes no use of satellite or roadside infrastructures or of any cooperative support from other vehicles; the control algorithm relies on sensor information from onboard sensors only. The extension to cooperative cruise control requires either fixed infrastructure as with satellites, roadside beacons, or mobile infrastructures as reflectors or transmitters on the back of other vehicles ahead. These systems use either a radar or laser sensor setup allowing the vehicle to slow down when approaching another vehicle ahead and accelerate again to the preset speed when traffic permits.
- *Lane departure warning (LDW), lane keep assistance (LKA), and Lane change assistance (LCA)*: An LDW system is designed to warn a driver when the vehicle begins to move out of its lane (unless a turn signal indicates the wish to leave the lane) on freeways and arterial roads. These systems are designed to minimize accidents by addressing the main causes of collisions: driver error, distractions, and drowsiness. Lane keep assistance systems actively control the lateral movements of the car to stay in a given lane. The next step in higher automation is lane change assistance (LCA). A LCA system can not only keep a lane but change it automatically when the driver sets the turn signal indicator. This requires an active monitoring of the traffic approaching from behind. The system will look for a safe gap in the traffic flow and automatically change the lane when possible by controlling the steering wheel.

- *Collision avoidance (pre-crash) system (CAS)*: Is an automobile safety system designed to avoid accidents or at least reduce the severity of an accident. Also known as a pre-crash system, forward collision warning system or collision mitigating system, radar, laser, and camera sensors are used to detect an imminent crash.
- *Automatic parking (AP)*: This is an autonomous car-maneuvering system that steers the car from a traffic lane into a parking place to perform parallel parking, perpendicular parking, or angle parking. These ADAS functions enhance the comfort and safety of driving in constrained environments, where a lot of care and experience is required to steer the car. The topic was discussed in detail in Chap. 10.

The market for ADAS systems is growing substantially, and the margins are still relatively high.

The market is dominated by:

- Automotive OEMs
- Device manufacturers and suppliers (1st Tier, 2nd Tier...)
- Start-ups focusing on special aspects like sensors, artificial intelligence, or frugal implementation of ADAS features
- Semiconductor companies

The device manufactures provide a multitude of components like radar and lidar sensors, camera systems, infrared sensors, alarm systems, illumination and sound alert systems, display units, and others.

Most automotive OEMs offer some ADAS functionality in their cars either optional or as standard. Premium OEMs like BMW, Audi, Daimler, etc. provide sophisticated ADAS features.

Major suppliers in the ADAS space are Continental, Bosch, Delphi, ZF/TRW, Visteon, Mobileye (acquired by Intel), and Valeo ([URL9 2017](#); [URL10 2017](#); [URL1 2017](#)).

Semiconductor companies are also active and moving up the value chain, among them NXP/Freescale Semiconductors (ICs), Infineon Technologies (ICs), and Intel which strengthened its position with the takeover of the Israeli ADAS specialist Mobileye.

Figure 11.3 illustrates the main sensor categories that are being used for advanced driver assistance functions:

- Ultrasound
- Camera/video
- Camera/infrared
- Radar (near and long distance)
- Lidar

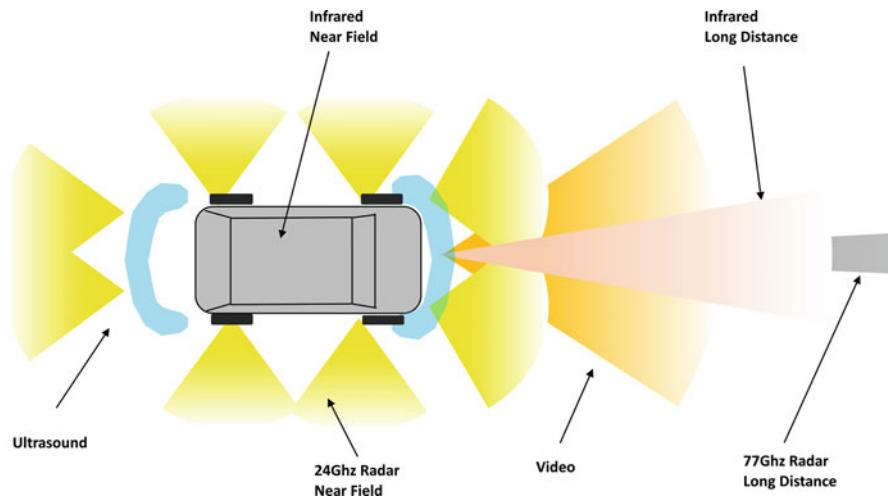


Fig. 11.3 Sensor application for different ADAS functions (see URL22 2017)

These sensors differ in resolution, range, accuracy, and price. With rapid advances in hardware and software, the prices for ADAS functions are coming down significantly.

Sensors were also discussed in Chap. 4, Sects. 4.2.4 and 4.2.7, and ADAS functions in Chap. 5, Sects. 5.4.1 and 5.5. The discussion in these sections is built on this material.

Figure 11.4 shows a mindmap (Müller and Haas 2014) which groups the different ADAS functions into two main categories, vehicle support and driver support.

11.2 Lane Departure Warning, Lane Keep Assistance, Obstacle Detection, and Crossing Assistance

11.2.1 Lane Keeping and Lane Change Assistance

The basic lane detection approach was described in Chaps. 4 and 5. In Fig. 11.5 the lane keep assistance function in a Seat Leon is shown. As per the current law, the system is not allowed to take over full control but requires the driver to touch and move the steering wheel every 30 s or so, as shown in Fig. 11.6. If the driver fails to do so, the car will send a warning, both visually (see Fig. 11.6) and acoustically and ultimately slow down and stop if no activity of the driver is being detected.

There are two main types of lane assistance systems:

- Systems which warn the driver (lane departure warning system, LDW) if the vehicle is leaving its lane (visual, audio, and/or vibration warnings)
- Systems which warn the driver and, if no action is taken, automatically take steps to ensure the vehicle stays in its lane (lane keeping assistant, LKA)

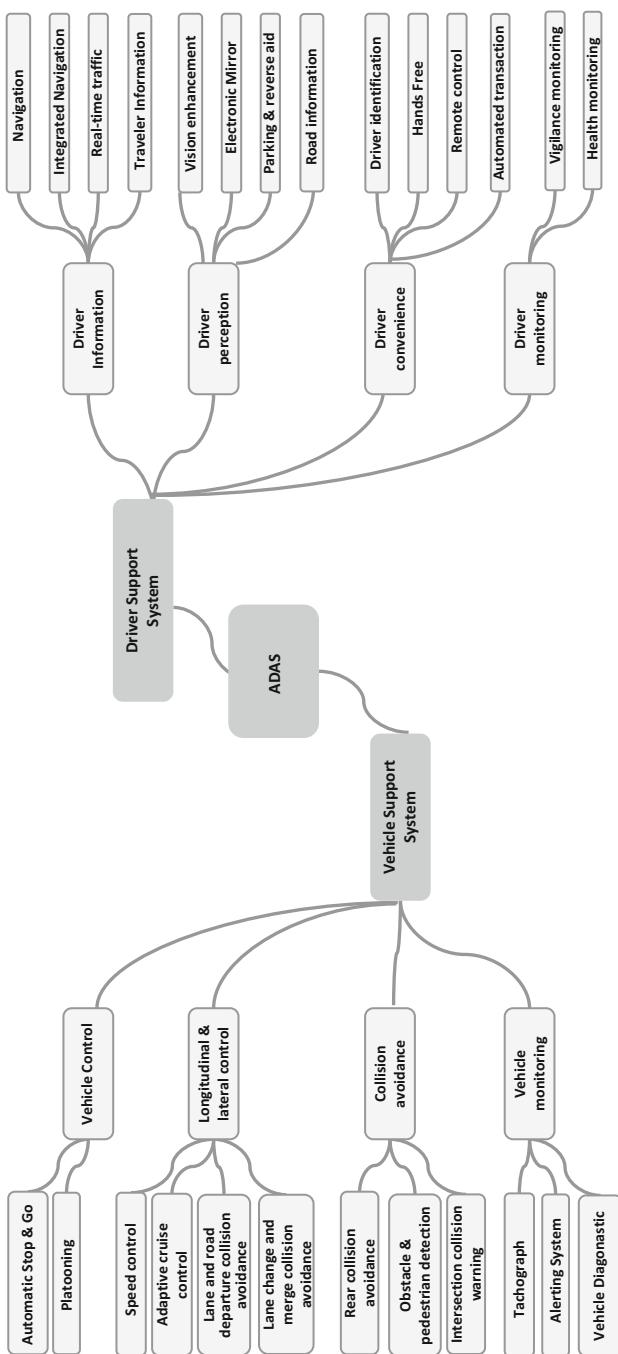


Fig. 11.4 Mindmap of ADAS functions (Müller and Haas 2014)



Fig. 11.5 Lane keeping assistant (LKA) ADAS function in a Seat Leon



Fig. 11.6 Lane keeping assistant (LKA) ADAS function in a Seat Leon; driver must take up control again

LDW can help prevent single vehicle roadway departure, lane change or merge collisions, and rollover crashes, as described below:

- *Single vehicle roadway departure:* LDW gives a warning as the car crosses the shoulder lane marking. Without the system, the car may be driven off the shoulder and crash into off-road obstacles, for example, light poles, signs, guardrails, trees, and stopped vehicles.
- *Lane change/merge:* LDW issues a warning as the car crosses center lane markings on multilane roadways, including solid lines, double lines, dotted lines, dashed lines, and raised pavement markers. Without the system, the car may be driven into an adjacent lane, resulting in a head-on or sideswipe collision.
- *Rollovers:* LDW may also prevent some crashes that would be categorized as rollover crashes. For example, if the vehicle drifts out of the lane onto the shoulder, the car could roll over if a sudden recovery maneuver is made.

LDW may also:

- Assist the driver in consistently keeping a vehicle in the lane, thereby reducing lane departure crashes.
- Reinforce driver awareness of vehicle position in the lane to maintain a more central lane position and improve the driver's attentiveness to the driving task.

LDW cannot prevent all single vehicle roadway departure crashes. These are warning devices and do not actively prevent crashes, they warn the driver so he/she can maneuver the car to prevent a crash. For example, crashes involving vehicle loss of control due to slippery roads and excessive speed on turns would not be prevented with these systems. Also, the systems will not prevent crashes due to intentional lane changes, which involve the driver's failure to see another vehicle in the adjacent lane or by a vehicle being in a blind spot. Some collision warning systems (CWS) have blind spot sensors to help prevent these types of crashes.

LDW should work under various operational scenarios:

- *Normal system start-up operation:* When the driver turns the ignition switch to start the vehicle, the LDW performs a power-up self-test, and the driver scans the warning indicator to determine any system malfunctions. If necessary, the driver may alert fleet maintenance for corrective action. When the vehicle reaches the minimum LDW tracking speed on a roadway with lane boundary markings, lane tracking begins.
- *Warning/alert situations:* When travelling at or above the minimum LDW tracking speed, a driver may unintentionally drift out of the lane. Then, the LDW issues a warning.
- *System fault conditions:* When the LDW cannot track the lane or a system fault occurs, the driver is notified via the lane-tracking indicator. This inability to track lanes may be due to a lack of lane markings, poor quality of lane markings, poor

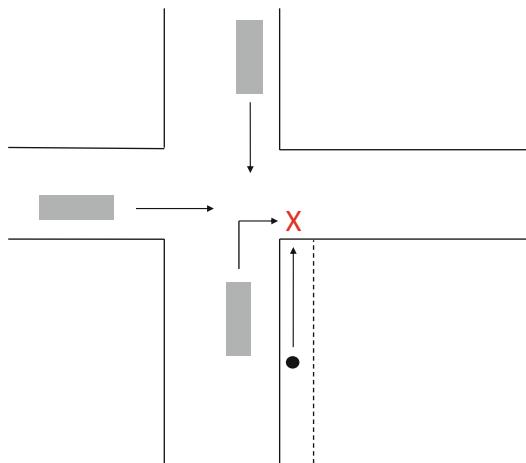
visibility, or a dirty/icy windshield. Although LDW cameras typically view the road through a portion of the windshield swept by the wipers, the driver can manually clean the windshield area in front of the LDW camera to see if the LDW begins to track. Some LDW may display various messages when certain types of faults or other conditions are detected, such as *Calibration in Progress*.

- *Well-marked roads:* The most commonly encountered roadway markings include single and double solid lines, dashed and dotted lines, and raised pavement markers where LDW should detect lane departures and issue warnings to a driver travelling over the minimum tracking speed.
- *Roads with missing or degraded lane boundary markers:* If lanes have missing or degraded lane markings, the driver may not receive a warning as the vehicle progresses outside of the lane, depending on the particular LDW used. On roads with only one set of markers, the driver should receive a warning when the warning threshold is crossed on that side, even if the system cannot detect the lane boundary on the other side.
- *Delivery points, arterials, and collectors:* Currently available LDW systems will not operate at delivery points and roads where the car travels at speeds below the minimum LDW tracking speed. Currently available LDW systems are made primarily for highway driving and will not function at lower speeds associated with some local roads.
- *Wet roads:* Due to reflections on wet road surfaces, LDW may occasionally be unable to detect lane markings; however, the lane-tracking indicator will show that the system is not providing warnings under these conditions.
- *Mud-/ice-/snow-covered roads:* When lane markings are not visible on roads covered by mud, ice, or snow, the lane-tracking indicator will show that the system is inactive. LDW may be beneficial in low visibility conditions (e.g., rain, fog, and falling snow) when lane markings are present.

As outlined in the previous section, lane keep assistance systems actively control the lateral movements of the car, thereby making sure that the car stays in the lane. Lane change assistance systems can perform a lane change maneuver on their own if the driver sets the turn signal indicator. This is a complex process which involves sophisticated image and sensor processing:

- *Analysis of traffic approaching from behind:* Vehicles with different speeds are approaching and need to be detected. The velocity needs to be estimated.
- *Space detection:* The system has to look for a safe gap in the traffic flow, based on the relatively velocity of vehicles approaching.
- *Maneuver control:* If an appropriate gap in the traffic flow is detected, the system will automatically change the lane by controlling the steering wheel.
- *Switch back to LKA mode:* Control is handed over to lane keep assistance.

Fig. 11.7 Turn assistance, avoiding collisions with pedestrians and cyclists



11.2.2 Turn Assistance

A serious cause of accidents is the collision with pedestrians or cyclists in busy city traffic (URL13 2015; URL30 2017). Especially, when turning into a side street, the driver has to look for crossing pedestrians or cyclists. Unfortunately, it is quite common that a cyclist, pedestrian, or animal is overlooked approaching from the side path.

Turn warning systems, turning assistance systems, and obstacle detection systems can identify these dangerous situations and help the driver to prevent accidents which often have severe consequences (see Fig. 11.7).

If an object is moving across, either in front or behind the vehicle, it is relatively easy to detect this. Also, if an object moves slowly towards a vehicle, the ADAS image analysis system has no problem to recognize this as the object will appear bigger when it approaches the vehicle. But the problem arises when pedestrians or cyclists are present in the blind spot. To avoid this situation, surround view systems give a complete view around the car.

Although, intersections represent a relatively small portion of a cyclist's travel route, these areas are where a cyclist is most at risk of getting hit by a car (URL29 2017).

Another possible challenge arises when the cyclist is moving very fast and intersects with the car's path. In this case, one can deduce that the cyclist is moving fast from optical flow but predicting whether he will intersect with the car's path or not is difficult. The key is to estimate the relative velocity of the cyclist as accurately as possible.

11.2.2.1 Pedestrian Detection and Object Detection Challenges

Pedestrians are often difficult to detect due to the following:

- Various styles of clothing
- Presence of occluding accessories

- Frequent occlusion between pedestrians
- Shadows
- Poor light conditions
- Different speeds
- Different shapes

11.2.2.2 Existing Approaches to Object/Pedestrian Detection

Object detection and tracking algorithms can be classified into object-based and non-object-based depending on object properties such as feature, model, and speed. The object-based approach analyzes the video on a frame-by-frame basis and detects the object based on its shape and motion. When no information is available about shape or motion, it is much more difficult to detect objects. In this case object detection is done based on previously calculated data such as other object information, location, time, and environment. These types of objects are known as indistinguishable objects, and the method is known as non-object-based detection.

The part-based model represents the body/object as deformable configurations of individual parts which are in turn modeled separately in a recursive manner. One way to visualize the model is a configuration of body/object parts interconnected by springs. The spring-like connections allow for the variations in relative positions of parts with respect to each other.

Another approach is to model pedestrians as collections of parts. Part hypotheses are firstly generated by learning local features, which include edge features, the orientation features, etc. These part hypotheses are then joined to form the best assembly of existing pedestrian hypotheses. This approach is attractive, but part detection itself is a difficult task.

A straightforward implementation is capturing and resizing the image (cropping the image inside a bounding box) to a fixed height and width. Features are extracted from these resized regions. The obtained features are then clustered to get the root filters. Given a root filter, $k \times d$ part filters are initialized at twice the spatial resolution in order to capture part details more precisely. Individual part locations are selected in two stages.

A further approach is the greedy initialization. The obtained part filters are greedily matched to image regions in order to maximize the energy map. The energy map is the squared norm of positive filter weights in each filter cell. The image regions which are matched are not considered later for matching other parts.

Ultimately, the re-defilement approach is applied using a stochastic search. After all the parts are matched, these are displaced one at a time randomly to maximize the amount of energy covered. Displacement results in a penalty which is proportional to the magnitude of the displacement. When no more energy can be covered, this phase is restarted. This process is repeated many times to avoid the selection of local maxima (Bhattacharjee 2013).

11.3 Image Processing and Image Analysis

ADAS functions depend on the combined evaluation of different sensors, like:

- Camera
- Infrared
- Lidar
- Radar
- Ultrasound

This section shows how image processing and image analysis can be applied to implement ADAS functions.

11.3.1 Computer Vision and Machine Vision

Computer vision and machine vision are terms that are being used synonymously (Davies 2012). The subjects deal with the analysis, design, and implementation of algorithms, hardware, and software for handling complex vision problems that humans or animals deal with every day. Machine vision has evolved rapidly and has benefited from the explosive growth in semiconductor performance and computer architecture.

There are two main applications for machine vision: navigation and manipulation.

- Navigation is the process of moving from one position to another.
- Manipulation is the process of actively handling objects, for example, by means of a robot manipulator.

Navigating in an unstructured, unknown environment is a complex task which is being mastered by humans and animals in a seemingly effortless manner. For a robot this is a challenge, as it has to maneuver through the environment avoiding obstacles, walls, stairs, pits, and others. One of the authors worked in a research group with researchers from Carnegie Mellon and Stanford university at the Daimler-Benz Technology Center in Berlin which developed a mobile robot in the 1990s. The computational complexity was so high that much of the calculations had to be done on a separate server, outside the robot platform, sending the results back to the robot.

Today, even smartphones have enough computational capabilities to run complex image processing and analysis programs, and indeed, some frugal ADAS applications use the smartphone's cameras and processing capabilities for obstacle detection, traffic sign recognition, and lane departure warning.

The subject of machine vision still offers a lot of unsolved research problems. Interdisciplinary by nature and being part of the vast field of artificial intelligence, machine vision is currently one of the busiest and most interesting research domains.

Computer graphics deals with the virtual construction of scenes on the screen from geometric information and description. Computer vision can be regarded as the inverse process of interpreting an image which is being captured by sensors like a stereo camera.

Often, complex image analysis problems can be broken down into simpler tasks and simplified by assumptions and knowledge about the physical scene. This is an important aspect if it comes to implementation of ADAS functions like lane keep assistance, which need to identify markings on the road.

Therefore, Sect. 11.3.2 discusses some basic principles of image processing like digital images, color models, conversion from one color space to another, spatial filtering, edge detection, and thresholding. Also, the section briefly presents the concept of morphological operations. Section 11.3.3 gives an overview of the detection of moving objects with sub sections like object tracking algorithms, and others. Section 11.3.4 introduces the optical flow algorithm, and Sect. 11.3.5 explains the implementation of image processing algorithms in MATLAB.

11.3.2 Basic Principles of Image Processing

11.3.2.1 Digital Images

An image can be considered to be a function from \mathbb{R}^2 to \mathbb{R} , mapping every point in the 2D plane to a particular real value of light and color perception. These values can be grayscales, for example, from 1 to 255, normalized intensity levels in the interval $[0,1]$, 0 and 1 for pure black-and-white (b/w) pictures or color vectors. If the values are normalized grayscales, the image $G(x,y)$ is described by the mapping

$$G : \mathbb{R}^2 \mapsto \mathbb{R}$$

$$(x, y) \mapsto G(x, y) \in [0, 1].$$

A single point on the plane and its corresponding value is called a *pixel*.

The digital image is being derived by quantizing the 2D space and mapping the real color or intensity values to discrete values. The image is characterized by its spatial resolution, the color model, the resolution, and the number scheme for the color or intensity values, for example, 8 bit values for red, green, and blue color. The spatial resolution is defined by the sensor, e.g., the elements of a charge-coupled device (CCD) sensor. There is a delicate trade-off between resolution, size of the sensor pixel element, cost, noise, and sensitivity to light.

A black-and-white image $G_{BW}(x,y)$ with a spatial resolution of 1024 x 1024 is a matrix G_{BW}

$$(g_{ij}) \quad i = 1, \dots, 1024; j = 1, \dots, 1024; g_{ij} \in \{0, 1\}.$$

An image consisting of intensity information will be denoted by $I(x,y)$.

11.3.2.2 Color Models

There are different color models like (Gonzalez et al. 2008):

- RGB – represents the red, green, and blue component of the image.
- NTSC – refers to the National Television System Committee standard used for color and monochrome television sets. Images are represented by three components: luminance (Y), hue (I), and saturation (Q).
- YCbCr – The YCbCr color space is used extensively in digital video. In this format, luminance information is represented by a single component, Y, and color information is stored as two color-difference components, Cb and Cr. Component Cb is the difference between the blue component and a reference value, and component Cr is the difference between the red component and a reference value.
- CMY – Cyan, magenta, and yellow are the secondary colors of light.
- CMYK – based on the CMY color model adding black as a fourth color for creating true black which is the predominant color in printing.
- HSV – refers to hue, saturation, and value. It is one of the several color systems used by people to select colors from a palette.
- HSI – refers to hue, saturation, and intensity. Decouples the intensity component from the color – carrying information (hue and saturation) in a color image.

RGB is a simple additive model where the colors are being generated by mixing three channels of basic colors that can generate any desired color. This is well suited to computer graphics, but humans tend to describe color in a different way, by hue, saturation, and brightness. The *HSI* model provides a natural way to describe colors as a 3-tuple (H,S,I) , where H is a description of the color tone, the hue value, S refers to the saturation level, and I denotes the value of the intensity/brightness. It is easy to derive the $I(x,y)$ intensity image from the *HSI* color model as the last position in the tuple corresponds directly to the I value.

The different color models can be converted into each other. It will be shown how this is done for *RGB* to *HSI* as this is often necessary for further processing of the camera images. Note, that the I value in the *HSI* model directly responds to the intensity level of an intensity image $I(x,y)$. Similarly, an *HSI* color model can be converted into the corresponding *RGB* model.

A color in the *HSI* model can be described as vector in the color unity circle (Gonzalez et al. 2008). This vector has the angle Θ and the length S .

Let (R,G,B) be a *RGB* color value. The corresponding *HSI* values can be computed in the following steps. Firstly, the H value depends on the relation between the B and G values:

$$H = \begin{cases} \Theta & \text{if } B < G \\ 360 - \Theta & \text{if } B > G \end{cases}$$

The value Θ is computed directly from the *RGB* values as follows:

$$\Theta = \cos^{-1} \left(\frac{\frac{1}{2}[(R - G) + (R - B)]}{\sqrt{(R - G)^2 + (R - B)(G - B)}} \right)$$

The saturation level is

$$S = 1 - \frac{3}{(R + G + B)} \min(R, G, B).$$

The intensity level is given by the arithmetic mean of the *RGB* values

$$I = \frac{1}{3}(R + G + B).$$

11.3.2.3 Spatial Filters

Often, images need to undergo a sequence of preprocessing steps, e.g., to remove unwanted artifacts and noise. This can be achieved with filters in the spatial or frequency domain. A spatial filter operates on the neighborhood of a particular pixel and replaces the pixel with a function of the neighboring pixels, for example, a weighted sum of these pixels. Often, a neighborhood of one pixel in all directions is chosen.

Let W be the 3×3 neighborhood matrix around a point (x,y) :

$$W = \begin{bmatrix} g(x-1, y-1) & g(x-1, y) & g(x-1, y+1) \\ g(x, y-1) & g(x, y) & g(x, y+1) \\ g(x+1, y-1) & g(x+1, y) & g(x+1, y+1) \end{bmatrix}$$

If one replaces every pixel with the sum

$$g_0(x, y) = \frac{1}{9} \sum_{i=-1}^1 \sum_{j=-1}^1 g(x+j, y+i)$$

noisy peaks will be averaged, and the picture will look smoother.

Figures 11.8 and 11.9 show the result of spatial filtering for removing salt-and-pepper noise (Gonzalez et al. 2008) which was added to the original image in Fig. 11.8 and then removed with a median filter with a 3×3 window (one pixel each to the left, right, top, and bottom). The result is shown in Fig. 11.9.

11.3.2.4 Canny Edge Detection Technique

Edge detection is one of the major steps in image processing. Edges define the boundary region of an image. A good edge detection technique maximizes the probability of detecting real edges and minimizes the detection of false edges. The Canny edge detection (CED) technique is known for 30 years, and still very popular. It was created by John Canny in the 1980s (Davies 2012).



Fig. 11.8 Noisy picture of a road (Interstate I5 in California near Bakersfield) distorted by salt-and-pepper noise



Fig. 11.9 Picture after noise cancellation with a spatial median filter

Edges are detected as spikes in the intensity level of the picture. This corresponds to a derivative of the intensity level.

As a preparation for discussing the Canny edge detection technique, the approximation of derivatives in discrete images has to be explained briefly.

Let G be a discrete image with intensity/gray levels $G(i,j)$. We are interested in the discrete approximation of the gradients in the x and y direction.

Let $g(j,j)$ be an arbitrary pixel in the image G . The 3×3 neighborhood around this pixel can be arranged in the form of the matrix

$$Z = \begin{pmatrix} z_1 & z_2 & z_3 \\ z_4 & z_5 & z_6 \\ z_7 & z_8 & z_9 \end{pmatrix}$$

Here z_k , $k = 1, \dots, 9$ denotes the values of the neighboring pixels, and z_5 is the value of the center pixel.

The partial derivatives G_x and G_y can be approximated by differences, using approximation methods like Sobel, Prewitt, or Roberts. With the Sobel method, one would get the following approximations:

$$G_x = (z_7 + z_8 + z_9) - (z_1 + 2z_2 + z_3)$$

$$G_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$$

The Canny edge detection algorithm works as follows:

- First, to remove small noise, image smoothing is done by using a Gaussian filter with a specified standard deviation σ .
- An edge in the image may point towards different directions; the Canny edge detection technique uses four filters to detect horizontal, vertical, and two diagonal edges in an image. The length of the edge gradient is

$$G = \sqrt{G_x^2 + G_y^2}$$

and the direction is represented by

$$\Theta = \tan^{-1} \left[\frac{G_x}{G_y} \right]$$

where G_x and G_y are the gradients in the x and y directions. As the image consists of discrete pixels, the gradient has to be approximated by one of the methods described before.

- Next, an image thresholding operation is performed. If the value of the magnitude image is less than the predefined threshold, then it is set to zero.
- Non-maximum suppression is done to reduce the edge breadth.

- Based on the result of the last stage for detecting edges and non-edges, two thresholds T_1 and T_2 are chosen, where $T_1 < T_2$. Pixel values greater than T_2 are defined as edges, and pixels with values less than T_1 are defined as the non-edge region. Pixel values in between T_1 and T_2 are considered to be edges if they are connected with an edge pixel.
- Finally, edge linking is performed.

Lines can be detected with the Hough transform, which takes a model of the line, maps it to the discrete set of points, and then computes the distances to this model line.

11.3.2.5 Image Thresholding

Image thresholding is used for removing unwanted information and noise. There are various approaches:

- Histogram based, i.e., any change in the histogram is analyzed.
- Clustering based, i.e., gray level samples are classified into two regions—background and foreground.
- Entropy based (thresholding is done based on entropy normalization).
- Spatial based, where thresholding is performed based on pixel correlation information.
- Object attribute based, i.e., thresholding is done based on similarity.

11.3.2.6 Morphological Operation

Dilation and erosion are fundamental to morphological image processing (Gonzalez et al. 2008). Dilation is an operation that “grows” or “thickens” objects in a binary image, and erosion thins or shrinks objects in a binary image. The thickening or shrinking process is determined by a structural element (Gonzalez et al. 2008). A translation of the set A by a point b is denoted as

$$A_b = \{c | c = a + b, a \in A\}$$

The dilation of A by the set B is defined as (URL27 2017)

$$A \oplus B = \bigcup_{b \in B} A_b$$

The dilation operator is commutative and associative, i.e.,

$$A \oplus B = B \oplus A = \bigcup_{a \in A} B_a$$

$$A \oplus (B \oplus C) = (A \oplus B) \oplus C$$

The set B is also called a structuring element that is used to complete/fill structures in the set A , e.g., broken lines, holes, etc. Computationally, the center of

the structuring element will be moved across all the pixel positions of set A . Wherever a 1-pixel overlaps a 0-pixel, this will be turned into a 1.

The erosion of the set A by another set B is (URL28 2017) described as

$$A \ominus B = \bigcap_{b \in B} A_{-b}$$

It also is commutative and associative. The erosion thins or shrinks the image in the sense that only those pixels which are completely covered by the center of the structuring element will be set to 1, and all other pixels are cleared and set to 0.

Example: Three pixels p_1, p_2 , and p_3 in a straight line form a structuring element, the set B , which can be arranged in the matrix

$$\begin{pmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

Now let us look at the set

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

The result of the erosion operation is

$$A \ominus B = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

The block of ones has been thinned to a line of single pixels. Mixing dilation and erosion yields the following relationships (Davies 2012):

$$A \oplus B \ominus B \neq A$$

$$A \oplus B \ominus B \subseteq A$$

$$A \ominus B \oplus B \supseteq A$$

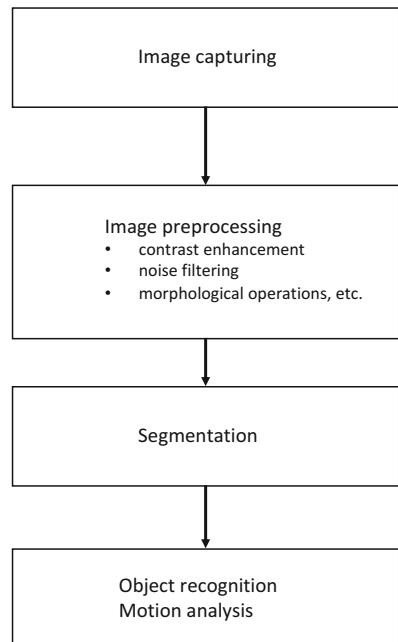
Figure 11.10 summarizes the different steps of image processing and analysis. After capturing, the image has to be preprocessed (enhancements, noise filtering, etc.). Then regions and segments have to be identified like lines (see in Fig. 11.11). Finally, the result of segmentation can be used to recognize objects. If the objects are moving, a motion analysis can be performed to find out in which direction they are moving. Object recognition is a complex process that often depends on extracting invariant features from the scene. In the next section, we will take a closer look at motion.

Figure 11.11 shows a typical road scene somewhere in the USA. The individual RGB image frames are converted into grayscale images. After that Canny edge detection is performed. Proper thresholding should be done to remove the unwanted regions. The final step uses a Hough line detection technique to extract the road markings which appear as lines (Bhattacharjee 2013). These lines and various parameters from the car like speed, steering angle, etc. could be the input for a lateral control algorithm that keeps the car in the lane.

11.3.3 Detection of Moving Objects

Object detection is an active field of research in computer vision (Rich and Knight 1991; Gonzalez and Woods 2008; Davies 2012; Haykin 2009). In this section, we

Fig. 11.10 Image processing and analysis



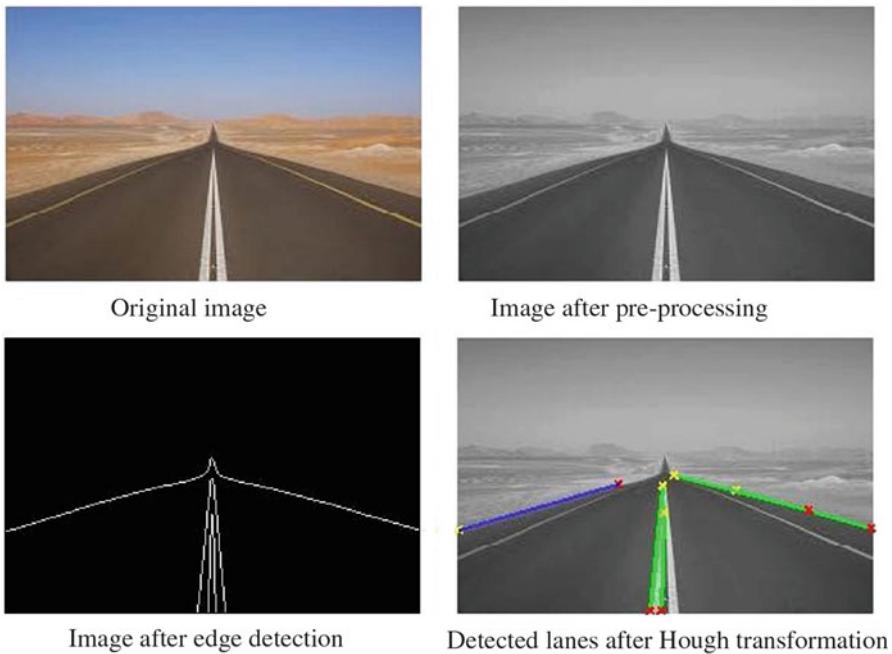


Fig. 11.11 Image processing for lane keeping assistance - preprocessing, edge detection, and line segmentation

present an approach for detecting moving objects and the direction of movement based on the analysis of Bhattacharjee (2013). The algorithms perform well under different ambient conditions like rain, fog, shadow, etc. and can detect a specific object based on the distance.

The object detection methods presented here are based on several, well-known image processing techniques and algorithms like edge detection, color space conversion, and morphological operations (Joshi 2009; Jain 2000; Gonzalez and Woods 2008). The following section gives a brief overview.

11.3.3.1 Object Tracking Algorithms

Several object tracking algorithms have been developed. Some of them are listed below:

1. Mean-shift tracking algorithm
2. Optical flow algorithm
3. Background subtraction algorithm

Mean-Shift Tracking Algorithm

The concept of the mean-shift tracking algorithm is based on the probabilistic histogram. It assumes that the object in the next frame will be located somewhere

in the vicinity of its location in the current frame. The algorithm is based on the concept of brute force tracking. The tracking method is described below:

- At first a window is defined and the target histogram is obtained.
- From the very next frame, the object tracking method begins by identifying those locations where the histogram distribution is most similar compared to the target histogram.
- For each frame, the same procedure repeats.

Optical Flow Algorithm

The optical flow can be described by the concept of relative motion between the observer and the image. Mathematically, the optical flow is a velocity field which basically wraps one image into another very similar image (Horn and Schunk 1981; Gonzalez and Woods 2008).

The literature discusses various implementations of optical flow, like Lucas-Kanade, Horn's optical flow method, Buxton-Buxton method, Black-Jepson method, general variation method, phase correlation method, and so forth (Gonzalez et al. 2008; Joshi 2009).

Background Subtraction Algorithm

Background subtraction is a method for identifying moving objects in a sequence of video frames. There are different types of background subtraction algorithms (Gonzalez and Woods 2008; Joshi 2009). In the field of computer vision, background subtraction is one of the popular methods for moving object detection because of its moderate computational effort (Gonzalez and Woods 2008). Noise can reduce the performance, and false detection is another critical issue. However, the background noise can be removed using several filters along with the main tracking algorithm; one such approach is mentioned below:

- At first the initial checkings (the aim is to check whether the camera and other related subsystems are working properly or not) need to be performed.
- After that the system checks whether anything like dust particles, leaves, paper, etc. is present in front of the camera lens (filter 1).
- The next stage deals with bad weather conditions such as rain and fog as they have a significant impact and reduce the image quality (filter 2).
- A specific problem is the presence of shadows which can lead to false detections. One way to deal with this is to remove shadows in an early stage of image processing (filter 3).
- Next background subtraction is performed to identify only those objects that are actually moving.

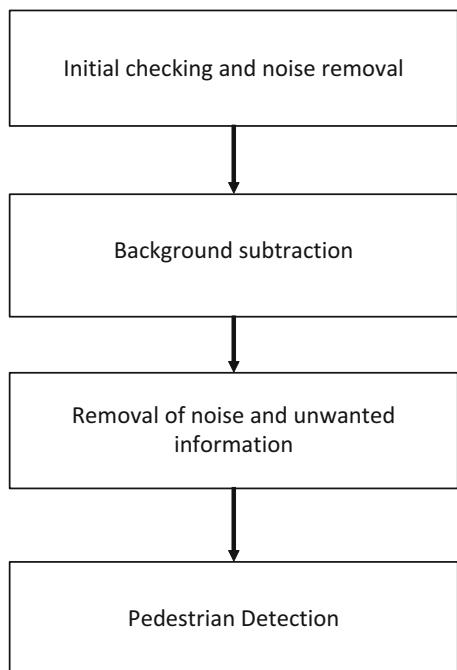
- Background subtraction is one of the simplest tracking algorithms. It's easy to implement but one of the major problem is information loss. One can mix several other algorithms with the actual background subtraction algorithm. For instance, an optical flow algorithm can be used to detect the direction of motion.

Figure 11.12 illustrates the concept of the detection mechanism.

The following steps have to be preformed:

- *Initial checking*: The purpose of the initial checking is to check whether the camera is working properly or not and to detect the presence of any occluding object on the lens of the camera. The latter condition can be recognized by computing the standard deviation of the image matrix.
- *Noise removal*: The presence of noise can degrade the performance of the system. Noise can be due to bad weather, rain, and fog. If visibility is poor, the problem can be dealt with by using histogram equalization or contrast enhancement (Davies 2012).
- *Shadow removal*: Shadows are another important aspect to consider. We are only interested in moving objects not the static objects, so the shape of the shadow changes continuously from place to place. This situation can be easily handled using the background subtraction method. After frame differencing, the shadow will appear as a small noisy portion.

Fig. 11.12 Object/pedestrian detection



By eliminating small pixel clusters, the shadow can be easily removed. Another approach is to remove the shadow at the early stages (before the background subtraction method is applied), but the problem with this approach is that, if the clothes of a person are black, then it will be difficult to distinguish between shadow (represented by a dark region) and non-shadow regions (Bhattacharjee 2013).

While tracking a car, again, the same problem may occur. The color of the car and its tires may be black and hence difficult to distinguish.

Sometimes a person may walk through a shadow region. Also, when two moving objects are very close to each other, the shadows can overlap.

There are many approaches for shadow removal (Davies 2012; Gonzalez and Woods 2008), but the problem is that most of them are condition specific; they fail to detect/remove multiple shadows of the same object. Also, the detection procedure may take a long time.

Therefore, to reduce the effect of shadows, the image frame is being converted into the *HSI* color space (RGB to *HSI*), and after that the average normalized value of each pixel is converted into the binary image. This approach is useful for minimizing the shadow effect, but it only works well on roads.

- *Background subtraction:* Using background subtraction the image foreground is extracted for further analysis. The purpose is to identify only moving objects as fast as possible. We use the frame differencing model which is defined by

$$|G(x, y, t+1) - G(x, y, t)|$$

where $G(x, y, t+1)$ is the frame at time $t+1$ and $G(x, y, t)$ is the frame at time t .

The problem with this method is that after the frame differencing operation, the effect of noise remains very high. Noise removal is done by setting a proper threshold. We use the global thresholding method (Otsu's method to minimize the intra-class variance of the black and white pixels), which calculates the value of the threshold for each frame dynamically (Gonzalez and Woods 2008).

- *Noise removal and removal of small area:* For further noise reduction, the area of each pixel cluster is calculated. At this stage, generally noisy regions (noise clusters) are fewer compared to the real object clusters (true clusters). Small noisy pixel clusters can be removed using standard cluster detection algorithms (or using a proper filter) and that way a noise free region is obtained.
- *Specific object detection:* Background subtraction and noise removal eliminates static objects and noise from the data frame. Furthermore, at this stage various cluster detection algorithms can be used to compute the size of remaining objects. One can also use a training dataset and a proper learning technique to classify these objects.

11.3.4 Optical Flow Algorithm

There are several different versions of optical flow algorithms. In this section, we discuss Horn's optical flow algorithm in detail (Horn and Schunck 1981).

11.3.4.1 Horn's Optical Flow Algorithm

Optical flow can be obtained from the relative motion between the object and the observer. The motion of the brightness patterns in the image is determined directly by the motions of corresponding points on the surface of the object.

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, z + \Delta z, t).$$

Assuming that the movement is very small, and then by using Taylor series, the following approximation holds

$$I(x + \Delta x, y + \Delta y, z + \Delta z) = I(x, y, t) + \left(\frac{\partial I}{\partial x}\right)\Delta x + \left(\frac{\partial I}{\partial y}\right)\Delta y + \left(\frac{\partial I}{\partial t}\right)\Delta t + \dots$$

Combining the above equations yields

$$\left(\frac{\partial I}{\partial x}\right)\Delta x + \left(\frac{\partial I}{\partial y}\right)\Delta y + \left(\frac{\partial I}{\partial t}\right)\Delta t = 0$$

Now dividing each side by Δt gives

$$\left(\frac{\partial I}{\partial x}\right)\frac{\Delta x}{\Delta t} + \left(\frac{\partial I}{\partial y}\right)\frac{\Delta y}{\Delta t} + \left(\frac{\partial I}{\partial t}\right)\frac{\Delta t}{\Delta t} = 0$$

From this equation, the velocity components can be defined as

$$\begin{aligned}\frac{\Delta x}{\Delta t} &= u \\ \frac{\Delta y}{\Delta t} &= v\end{aligned}$$

This yields

$$I_x u + I_y v + I_t = 0.$$

The equation can be written in another way

$$(I_x, I_y) \cdot (u, v)^T = -I_t.$$

The components of the movement in the direction of the brightness gradient (I_x, I_y) equal

$$-\frac{I_t}{\sqrt{I_x^2 + I_y^2}}.$$

If neighboring points on the objects have similar velocities, then the velocity field of the brightness patterns in the image varies smoothly almost everywhere. Discontinuities in flow can be expected where one object occludes another.

Additional constraints can be expressed by minimizing the square of the magnitude of the gradient of the optical flow velocity.

One can minimize the sum of the squares of the Laplacians of the x and y component of the flow:

$$\nabla^2 u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$$

$$\nabla^2 v = \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}$$

Images may be sampled at intervals on a fixed grid of points. In Horn's paper (Horn and Schunck 1981), it is assumed that the measured brightness is $I_{i,j,k}$ at the intersection of the i -th row and the j -th column in the k -th image frame. Each measurement is the average over the area of a picture cell and over the length of the time interval. Therefore, the intensity values can be expressed as

$$I_x = \frac{1}{4}(I_{i,j+1,k} - I_{i,j,k} + I_{i+1,j+1,k} - I_{i+1,j,k} + I_{i,j+1,k+1} - I_{i,j,k+1} + I_{i+1,j+1,k+1} - I_{i+1,j,k+1})$$

$$I_y = \frac{1}{4}(I_{i+1,j,k} - I_{i,j,k} + I_{i+1,j+1,k} - I_{i,j+1,k} + I_{i+1,j,k+1} - I_{i,j,k+1} + I_{i+1,j+1,k+1} - I_{i,j+1,k+1})$$

$$I_t = \frac{1}{4}(I_{i,j,k+1} - I_{i,j,k} + I_{i+1,j,k+1} - I_{i+1,j,k} + I_{i,j+1,k+1} - I_{i,j+1,k} + I_{i+1,j+1,k+1} - I_{i+1,j+1,k})$$

By approximating the Laplacian of each u and v , we receive

$$\nabla^2 u \sim k(\bar{u}_{i,j,k} - u_{i,j,k})$$

$$\nabla^2 v \sim k(\bar{v}_{i,j,k} - v_{i,j,k})$$

where \bar{u} and \bar{v} are the local averages, one can write:

$$\begin{aligned} \bar{u}_{i,j,k} &= \frac{1}{6}(u_{i-1,j,k} + u_{i,j+1,k} + u_{i+1,j,k} + u_{i,j-1,k}) \\ &\quad + \frac{1}{12}(u_{i-1,j-1,k} + u_{i-1,j+1,k} + u_{i+1,j+1,k} + u_{i+1,j-1,k}) \end{aligned}$$

$$\begin{aligned}\bar{v}_{i,j,k} = & \frac{1}{6}(v_{i-1,j,k} + v_{i,j+1,k} + v_{i+1,j,k} + v_{i,j-1,k}) \\ & + \frac{1}{12}(v_{i-1,j-1,k} + v_{i-1,j+1,k} + v_{i+1,j+1,k} + v_{i+1,j-1,k})\end{aligned}$$

The value of the Laplacian 2D filter is

$$\begin{pmatrix} 0,5 & 1 & 0,5 \\ 1 & -6 & 1 \\ 0,5 & 1 & 0,5 \end{pmatrix}$$

By averaging these values, one gets

$$\begin{pmatrix} \frac{1}{12} & \frac{1}{6} & \frac{1}{12} \\ \frac{1}{6} & -1 & \frac{1}{6} \\ \frac{1}{12} & \frac{1}{6} & \frac{1}{12} \end{pmatrix}$$

It is important to minimize the sum of errors in the equation for the rate of change of brightness

$$\epsilon_b^2 = I_x u + I_y v + I_t.$$

The estimate of the departure from the smoothness in the velocity flow is

$$\epsilon_c^2 = (\bar{u} - u)^2 + (\bar{v} - v)^2$$

As the image brightness measurement will be corrupted by quantization error and noise, one cannot expect ϵ_b^2 to be zero. This quantity will tend to have an error magnitude which is proportional to the noise in the measurement. For normalization, a weight factor α^2 (also known as regularization factor) is chosen. The total error to be minimized is

$$\epsilon^2 = \alpha^2 \epsilon_c^2 + \epsilon_b^2.$$

To minimize the equation, one needs to differentiate with respect to ϵ^2 for finding out suitable values of the optical flow velocity (u, v)

$$\frac{\partial \epsilon^2}{\partial u} = -2\alpha^2(\bar{u} - u) + 2(I_x u + I_y v + I_t)I_x$$

$$\frac{\partial \epsilon^2}{\partial v} = -2\alpha^2(\bar{v} - v) + 2(I_x u + I_y v + I_t)I_y$$

If one sets these two derivatives to zero one will get

$$-2\alpha^2(\bar{u} - u) + 2(I_x u + I_y v + I_t)I_x = 0$$

$$\alpha^2(\bar{u} - u) = (I_x u + I_y v + I_t)I_x$$

$$(\alpha^2 + I_x^2)u + I_x I_y v = (\alpha^2 \bar{u} - I_x I_t)$$

Similarly, for v ,

$$I_x I_y u (\alpha^2 + I_y^2) v = (\alpha^2 \bar{v} - I_y I_t)$$

Therefore, the determinant of the coefficient matrix is

$$\begin{aligned} (\alpha^2 + I_x^2)(\alpha^2 + I_y^2) - (I_x I_y)(I_x I_y) &= \alpha^4 + \alpha^2 I_x^2 + \alpha^2 I_y^2 + I_x^2 I_y^2 - I_x^2 I_y^2 \\ &= \alpha^4 + \alpha^2 I_x^2 + \alpha^2 I_y^2 = \alpha^2(\alpha^2 + I_x^2 + I_y^2) \end{aligned}$$

These equations can be rewritten in the following format:

$$(\alpha^2 + I_x^2 + I_y^2)u = (\alpha^2 + I_y^2)\bar{u} - (I_x I_y)\bar{v} - (I_x I_t)$$

$$(\alpha^2 + I_x^2 + I_y^2)v = (\alpha^2 + I_x^2)\bar{v} - (I_x I_y)\bar{u} - (I_y I_t)$$

$$(\alpha^2 + I_x^2 + I_y^2)u = (\alpha^2 + I_y^2)\bar{u} - (I_x I_y)\bar{v} - (I_x I_t)$$

$$(\alpha^2 + I_x^2 + I_y^2)u = (\alpha^2 + I_x^2 + I_y^2)\bar{u} - (I_x^2 u) - (I_x I_y)\bar{v} - (I_x I_t)$$

$$(\alpha^2 + I_x^2 + I_y^2)(u - \bar{u}) = -(I_x^2 \bar{u} - I_x I_y \bar{v} - I_x I_t)$$

$$(\alpha^2 + I_x^2 + I_y^2)(u - \bar{u}) = -(I_x \bar{u} + I_y \bar{v} + I_t)I_x$$

which yields

$$(\alpha^2 + I_x^2 + I_y^2)(u - \bar{u}) = -(I_x \bar{u} + I_y \bar{v} + I_t)I_x$$

$$(\alpha^2 + I_x^2 + I_y^2)(v - \bar{v}) = -(I_x \bar{u} + I_y \bar{v} + I_t)I_y$$

α^2 plays a significant role only for areas where the brightness gradient is small, preventing wrong adjustments to the estimated flow velocity due to noise in the estimated derivatives. This parameter should be roughly equal to the expected

noise in the estimation of $(I_x^2 + I_y^2)$. If one allows α^2 to be equal to zero, then one can obtain the solution of a constrained minimization problem. Applying this to the previous equation yields

$$(I_x^2 + I_y^2)(u - \bar{u}) = -(I_x\bar{u} + I_y\bar{v} + I_t)I_x$$

By solving the equation iteratively, we can write the velocity estimation equation as

$$\begin{aligned} u^{n+1} &= \bar{u}^n - \frac{(I_x\bar{u} + I_y\bar{v} + I_t)I_x}{\alpha^2 + I_x^2 + I_y^2} \\ v^{n+1} &= \bar{v}^n - \frac{(I_x\bar{u} + I_y\bar{v} + I_t)I_y}{\alpha^2 + I_x^2 + I_y^2} \end{aligned}$$

11.3.4.2 Centroid-Based Approach

At IIITB, a simple centroid based optical flow algorithm has been developed and the results are promising. Further details are mentioned below. The centroid in a plane can be computed by taking the average position of all the points in the plane. Similarly, in image processing the centroid is defined as the weighted average of image pixel intensities. It is represented by a vector that specifies the center of mass of the region. The first element represents the horizontal coordinate, while the second element represents the vertical coordinate.

If one combines the concept of centroid with the concept of Horn's algorithm, then for each different region, the centroid will be calculated, and both the concept of image centroid and Horn's optical flow depend on the image intensity. Hence, each region will be considered as a part, and the centroid of that part will be calculated, and again based on that information, the optical flow will be calculated. This integrated approach can track small movements of the object and also can detect an occluding object (Bhattacharjee 2013).

Background subtraction and the centroid-based optical flow algorithm can be used together too. Here, the background subtraction will be used to identify moving objects, and the optical flow algorithm can be used for detecting the direction of motion. In such a system, background subtraction will run as the main process, and optical flow will run as a subroutine. This approach will reduce computation time and improve the performance.

Figure 11.13 illustrates the results for various motion detection techniques analyzed by Bhattacharjee (2013).

11.3.5 Implementation Using MATLAB

MATLAB is a powerful package for numerical computation which was developed by MathWorks Inc. (URL2 2017). The popularity is due to many built-in functions

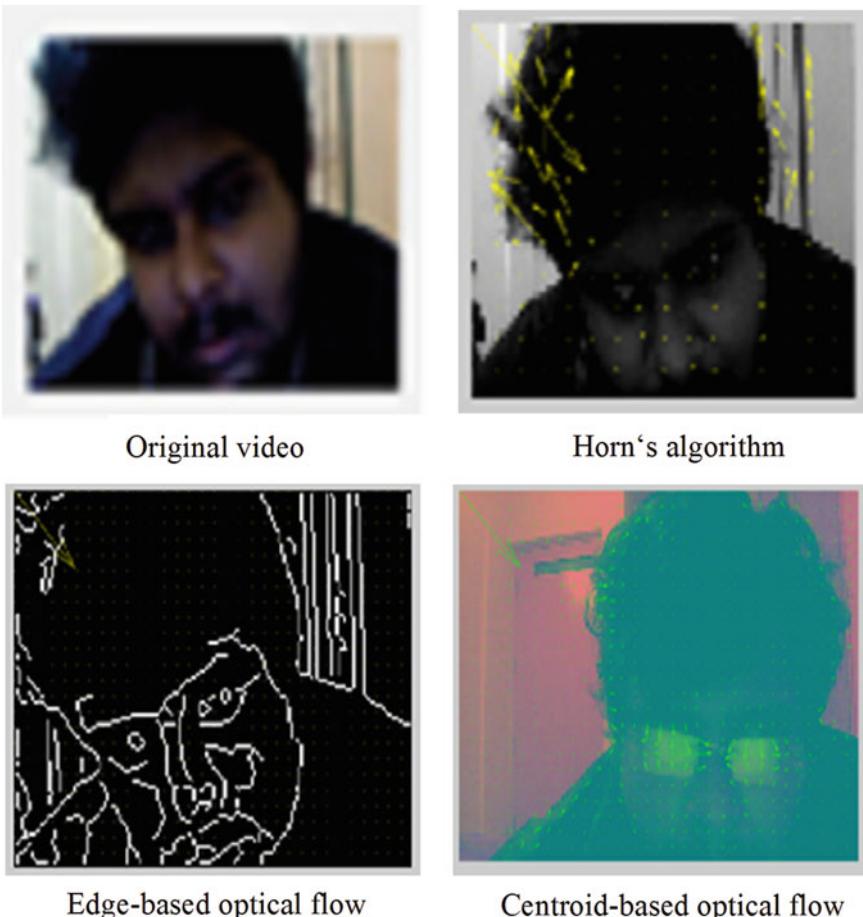


Fig. 11.13 Detection of moving objects and motion (see Bhattacharjee 2013)

for the efficient handling of matrix computations, powerful visualization, rich customization capabilities, the possibility to integrate C code, and a huge set of toolboxes for all sorts of engineering applications (Pratap 2006).

Although the focus is clearly on numerical computations, symbolic computations offered by computer algebra systems like Mathematica, Maple, and Derive are also possible with MATLAB's symbolic computation toolbox.

Together with the block-oriented simulation environment, Simulink, MATLAB offers the possibility for rapid analysis and prototyping of algorithms for a vast range of engineering applications, particularly:

- Signal processing
- System identification
- Control systems (including robust and nonlinear design methodologies)

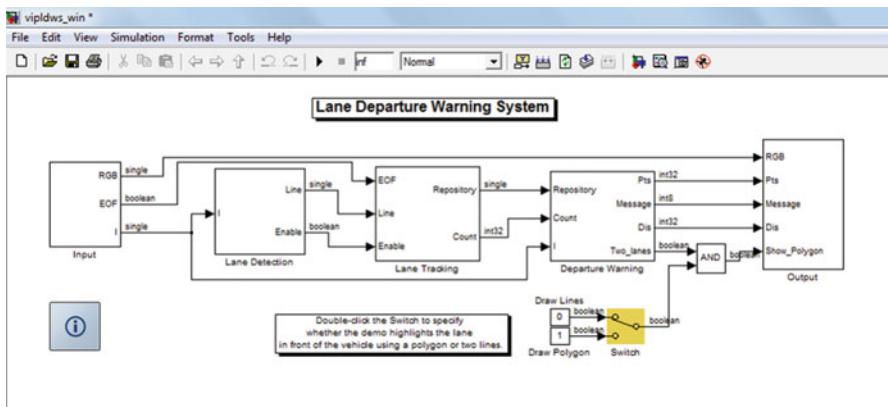


Fig. 11.14 Simulink implementation of a lane departure warning system (Gaonkar et al. 2011)

- Image processing
- Optimization problems

With the additional auto-code generation functionality (e.g., TargetLink), MATLAB programs can be directly converted to executable C and C++ code for different embedded target platforms. Figure 11.14 shows a Simulink implementation of a lane departure warning system that was implemented as a student project in the 2011 Mechatronics class at IIIT-B.

11.3.5.1 Image Processing Toolbox

The Image Processing Toolbox (IPT)TM provides a comprehensive set of standard algorithms, functions, and apps for image processing, analysis, and visualization. One can perform image enhancement, image de-blurring, feature detection, noise reduction, image segmentation, and geometric transformations. Many toolbox functions are multi-threaded to take advantage of multicore and multiprocessor computers.

The Image Processing Toolbox supports a diverse set of image types, including high dynamic range, gigapixel resolution, and tomographic. Visualization functions offer the possibility to explore an image, examine a region of pixels, adjust the contrast, create contours or histograms, and manipulate regions of interest (RoIs). With the toolbox, one can restore degraded images, detect and measure features, analyze shapes and textures, and adjust color balance.

Device-independent color management enables the user to accurately represent color independently from input and output devices. This is useful when analyzing the characteristics of a device, quantitatively measuring color accuracy, or developing algorithms for several different devices.

The toolbox provides a comprehensive suite of algorithms and visualization functions for image analysis tasks such as statistical analysis, feature extraction, and object recognition.

11.3.5.2 Images in MATLAB

MATLAB's Image Processing Toolbox supports four types of images:

- RGB images
- Indexed images
- Intensity images
- Binary images

RGB images consist of three matrices, each one carrying the respective R , G , and B values. Indexed images consist of a pixel matrix and a color map. The pixel matrix contains entries (indices) into the color map. An indexed image is a matrix where every entry provides an index into an $m \times 3$ matrix of RGB values. The color map has three entries for the RGB values for each index. Grayscale images often contain 256 levels of gray levels that can be either normalized to the interval $[0,1]$ or represented using integers or byte values. A black-and-white image corresponds to a matrix of logical 0s or 1s.

The power of the IPT stems from the combination of MATLAB's optimized matrix and array operations with special image processing functions which are implemented as MATLAB m-functions. These functions provide a set of built-in operations for low-level, medium-level, and even higher-level image processing and analysis. Moreover, these functions can be even combined with algorithms from special toolboxes like neural networks and genetic optimization.

Images are loaded with the `imread` functions that can process a variety of different formats like PNG, TIFF, GIF, and JPEG. The `imshow` function displays the images on the screen, and `imwrite` stores images back on the file system.

Special MATLAB functions for low-level image processing are histogram processing, filtering, edge detection, and frequency domain transforms.

Medium-level functions are image compression and reconstruction, line, and contour detection based on the Hough transformation. There is a rich set of morphological operations and segmentation functions.

Finally, higher-level operations for object recognition are principal component analysis (PCA) and nearest neighbor operations. Many functions from other toolboxes like the neural network toolbox or the genetic optimization toolbox can be deployed to implement pattern recognition and object recognition functions.

Table 11.1 presents an overview of some of the main functions. A detailed description is given in (Gonzalez et al. 2008). MathWorks also provides details on the image processing toolbox functions on their website ([URL2 2017](#); [URL31 2017](#)).

11.3.5.3 Statistical Functions

Statistical functions let one analyze the general characteristics of an image by:

- Computing the mean or standard deviation
- Determining the intensity values along a line segment
- Displaying an image histogram
- Plotting a profile of intensity values

Table 11.1 Some important functions of MATLAB's Image Processing Toolbox

MATLAB function	Description	Example
imread	Reads an image file from disk	<code>g = imread('anImage.jpeg')</code> reads the image <code>anImage.jpeg</code> from the file system and stores it in <code>g</code>
imwrite	Writes an image file to disk	<code>imwrite(g, 'filename.gif', 'compression', comp_value, 'resolution', res_value)</code> Stores the image <code>g</code> in the file <code>filename.gif</code> on the disk. Compression parameters and resolution is specified in the values <code>comp_value</code> and <code>res_value</code>
imshow	Displays image file on screen in a window	<code>imshow(g, 10)</code> displays an image <code>g</code> with 10 intensity levels
imhist	Computes histogram of the given image	<code>h = imhist(g)</code> <code>h</code> is the histogram of the image <code>g</code>
histeq	Histogram equalization	<code>g = histeq(f, n)</code> computes an equalized histogram of <code>f</code> with <code>n</code> intensity levels
im2bw	Converts an image into a b/w image	<code>bw = im2bw(g)</code> converts the image <code>g</code> into a black-and-white image <code>bw</code> which consist of 0s and 1s
iminfo	Shows the key information about an image	
imfilter	Linear spatial filtering	<code>g = imfilter(f, w, 'replicate')</code> computes the linear filter result of applying the filter functions <code>w</code>
fspecial	Special spatial filter	<code>g = fspecial('gaussian', [3 3], sigma)</code> Gaussian filtering of image <code>f</code> generating the resulting image <code>g</code> . The filter size is 3×3 , <code>sigma</code> denotes the standard deviation
fft2	Fast Fourier transform	<code>g = fft2(f)</code> Computes the FFT of image <code>f</code>
edge	Edge detection	<code>g = edge(f, 'canny', T, sigma)</code> computes a new image <code>g</code> that contains edges being detected by the Canny algorithm. <code>T</code> is a vector with the thresholds T_1 and T_2 , $T = [T_1 T_2]$, <code>sigma</code> is the standard deviation of the smoothing filter
imresize	Resize an image	<code>g = imresize(f)</code>

11.3.5.4 Edge Detection Algorithm

The toolbox offers a wide range of edge detection algorithms that are capable of identifying object boundaries in an image. These algorithms include the Sobel, Prewitt, Roberts, and Canny approach. The powerful Canny method can detect true weak edges without being distracted by noise.

11.3.5.5 Morphological Operators

Morphological operators enable one to detect edges, enhance contrast, remove noise, segment an image into regions, thin regions, or compute skeletons. Morphological functions in the Image Processing Toolbox (IPT) include:

- Distance transform
- Erosion and dilation
- Labeling of connected components
- Opening and closing
- Reconstruction
- Watershed segmentation

The following MATLAB code performs the morphological dilation operation. The original image in Fig. 11.15 is read from the file system, resized, converted to black and white (Fig. 11.16) and then dilated. The result is shown in Fig. 11.17.

```
g = imread('thumb_streetscene.jpg');
g_small = imresize(g, [300 200]);
g_bw = im2bw(g_small); g_d = imdilate(g_bw,ones(3));
figure, imshow(g_small);
figure, imshow(g_bw); figure, imshow(g_d);
```

11.3.5.6 Matlab Object Tracking Functions and Blocks

MATLAB provides computer vision based algorithms for object detection and tracking. They can either be used as a single function or be combined with other functions for performing complicated operations. Some of the MATLAB functions and blocks are listed below (URL32 2017):

Fig. 11.15 Image of a street in the USA after resizing



Fig. 11.16 Image converted to black and white



Fig. 11.17 Image after dilation operation



`assignDetectionToTracks`: It uses James Munkres's variant of the Hungarian assignment algorithm in the background.

`configureKalmanFilter` and `vision.KalmanFilter` class: As the name suggests, it uses Kalman filter algorithm in the background.

`vision.HistogramBasedTracker`: This function uses histogram based mean shift (CAMShift) algorithm for object tracking.

`vision.PointTracker`: This function uses the Kanade-Lucas-Tomasi (KLT) algorithm in the background.

Apart from tracking functions MATLAB also provides blocks for more complicated applications. Some of the key blocks for tracking are:

- Optical Flow
- Block Matching
- Template matching

Furthermore, some of the other MATLAB inbuilt functions can also be incorporated into an ADAS design. For instance, the `vision.ForegroundDetector` function uses Gaussian mixture models to detect the foreground. This function can be used to detect most of the moving objects in the frame. The `vision.PeopleDetector` function uses special features for people detection. This function can be deployed for pedestrian detection or crossing alert based systems.

11.4 Autonomous Driving

The topic of self-driving or autonomous cars is one of the most active areas of research (Dudenhöffer 2016; Johanning and Mildner 2015; Maurer et al. 2015; Siebenpfeiffer 2014). Most automotive OEMs and many suppliers are working in this field as we have outlined in the beginning of this chapter and throughout the book. Self-driving cars will provide mobility for the young and elderly and visually impaired, people without driver's license, and those who would like to spend travel time more efficiently. The potential benefits have triggered a race for the pole position. The complexity of the technology is enormous, and the legal framework has yet to evolve (Maurer et al. 2015). The hope is to cut down on accidents significantly as self-driving cars will have much more information available than a human driver, and the software controlling the car does not suffer from fatigue, intoxication, or distraction that human drivers are so prone too. Moreover, driving the car for hours might be a waste of time. This time can be used efficiently in a self-driving vehicles for all sorts of activities (office work, reading, online shopping, watching multimedia, etc.), and this is why the prospect of these use cases has

attracted many IT giants like Google and Apple to work in this field (Abraham et al. 2016). Both companies are interested in autonomous driving although the exact implementation of their business model might differ.

A race is on and lots of new cooperation models and alliances have come up (Dudenhöffer 2016; Freitag 2016).

In Table 11.2 the transition from driver assistance to autonomous driving is illustrated. One can differentiate five levels (1–5) of various degrees of assistance and a level 0 without any assistance (URL11 2015; URL21 2017). The first level (simple) includes assistance functions of active safety like ABS and ESP; the next level (partial automation) supports the driver with steering like lane keep assist, keeping the distance to the front car like adaptive cruise control or parking. Level 3 (conditional automation) already provides major assistance with maneuvering the car, long distance drive control, and remote parking features. However, the driver still has to supervise/monitor everything and from time-to-time resume control to show his or her attention. On level 4 (high automation), there are defined used cases like highway travel where the vehicle can drive automatically. Eventually, level 5—full autonomy—does not involve any human interference any more. The car does everything on its own, and the passenger can concentrate on other activities. Here the control of the car is completely handed over to the machine as shown in Table 11.2.

Level 5 autonomy means that the car will be able to take a passenger from one point to another without any manual intervention. It is clear that the step from level 3 (conditional automation) to 4 (high automation) and finally to level 5 (full automation) is not just a linear step but an abrupt jump of complexity with major impact on the whole automotive technical and legal ecosystem. Connectivity will play a major role as the self-driving car can process a huge amount of information and can get crucial warnings well in advance (e.g., about any obstacles on the road, even if it is behind a curve where it is not visible). Traffic signs will actively communicate with cars approaching, and connected cars can negotiate the way of right at crossings automatically. Cybersecurity becomes also crucial as autonomous cars will communicate with infrastructure, the cloud, and other cars (see also Chap. 6).

In order to make autonomous driving a reality, one needs a combination of different methodologies:

- Adaptive software systems that can be updated over the air and communicate with software systems in the clouds.
- Artificial intelligence and machine learning to actively learn and improve the performance.
- Car-to-infrastructure and car-to-car communication providing valuable information about the traffic situation.
- Different sensors (vision), with different capabilities, accuracy, and response times will generate a detailed picture of the environment.
- High-definition maps will give precise information about the surroundings, including detailed information in the third dimension (e.g., the elevation of a street border).

Table 11.2 Transition steps to autonomous driving (URL11 2015)

Level 0	Level 1 Assisted	Level 2 Partial automation	Level 3 Conditional automation	Level 4 High automation	Level 5
Driver only	Driver continuously performs the longitudinal or lateral dynamic driving task	Driver must monitor the system at all times	Driver does not need to monitor the system at all times	Driver is not required during defined use case ^a	Full automation No driver required during entire journey
No intervening vehicle system active	Other driving task is performed by the vehicle	System performs longitudinal and lateral driving task in a defined use case ^a	Driver must be capable of resuming dynamic driving task	System performs the longitudinal and lateral driving task in all situations in a defined use case ^a	System performs entire dynamic driving task on all road types, speed ranges, and environmental conditions
				Recognizes its limits and requests driver to resume the dynamic driving task with sufficient time margin	

^aUse cases refer to road types, speed ranges, and environmental conditions

- Powerful new bus systems to transport the increased multimedia sensor information.
- Scalable software architectures and middleware for processing sensor input, implementing fusion algorithms, pre- and post-processing, and analysis and machine learning.
- Semiconductor solutions for fast image and digital signal processing (Nvidia, Qualcomm, etc.).
- Sensor fusion algorithms to combine different sensor sources.

The fundamental aspects of advanced driver assistance functions that form the basis for autonomous driving are illustrated in Fig. 11.18 (Müller and Haas 2014; Reif 2014).

Camera sensors, image processing, and analysis will play a major role as these technologies have made huge progress over the last two decades and also are relatively cheap. Therefore, Sect. 11.3.5 gave insight into algorithms and their rapid prototyping in MATLAB/Simulink. Moreover, advances in mobile and autonomous robots have yielded a lot of results that are now valuable for self-driving cars (Bekey 2005; Corke 2011; Hertzberg et al. 2012; Kaplan 2016).

The OEMs and technology suppliers follow different strategies in implementing ADAS, high automation and full autonomy. In the case of adaptive cruise control, for example, some OEMs rely on a stereoscopic camera, while others use long-range radar in conjunction with a mono-vision camera (URL12 2015; URL13 2016; URL25 2017). Some highly automated vehicles may deploy three or more lidars in conjunction with additional sensors and GPS to give the vehicle a 360° view of its surroundings. Others might not use lidar at all, operating with a combination of radar and camera systems instead (URL5 2017). Whatever the specific choices might be, OEMs will rely on improved processing speeds to handle the large amount of data from the sensors.

The interior design of autonomous cars will differ substantially from classic cars as the driver's seat, steering wheel, and pedals are not needed any more. Instead the space can be used otherwise. The interior might look more like a living room where passengers can sit face-to-face in a meeting room atmosphere.

Automotive OEMs are experimenting with these additional degrees of freedom, and many exciting ideas have come up. In this regard, Fig. 11.22 shows VW's vision of a flexible interior design which was presented at the IAA 2017, and Fig. 11.24 shows Daimler's EQ vision. VW's concept car is both an autonomous vehicle as well as a "normal" car which is being driven by a human driver as it can be seen from Fig. 11.19 and Fig. 11.20. The steering wheel is retractable and can be pulled back if the car switches from human-driven by a human self-driving mode as shown in Figs. 11.21 and 11.22.

Such flexibility is very useful as the driver can opt for autonomous mobility after drinking alcohol, being tired or generally not feeling fit for driving. A car could also enforce the autonomous mode if it senses that the driver is intoxicated with alcohol.

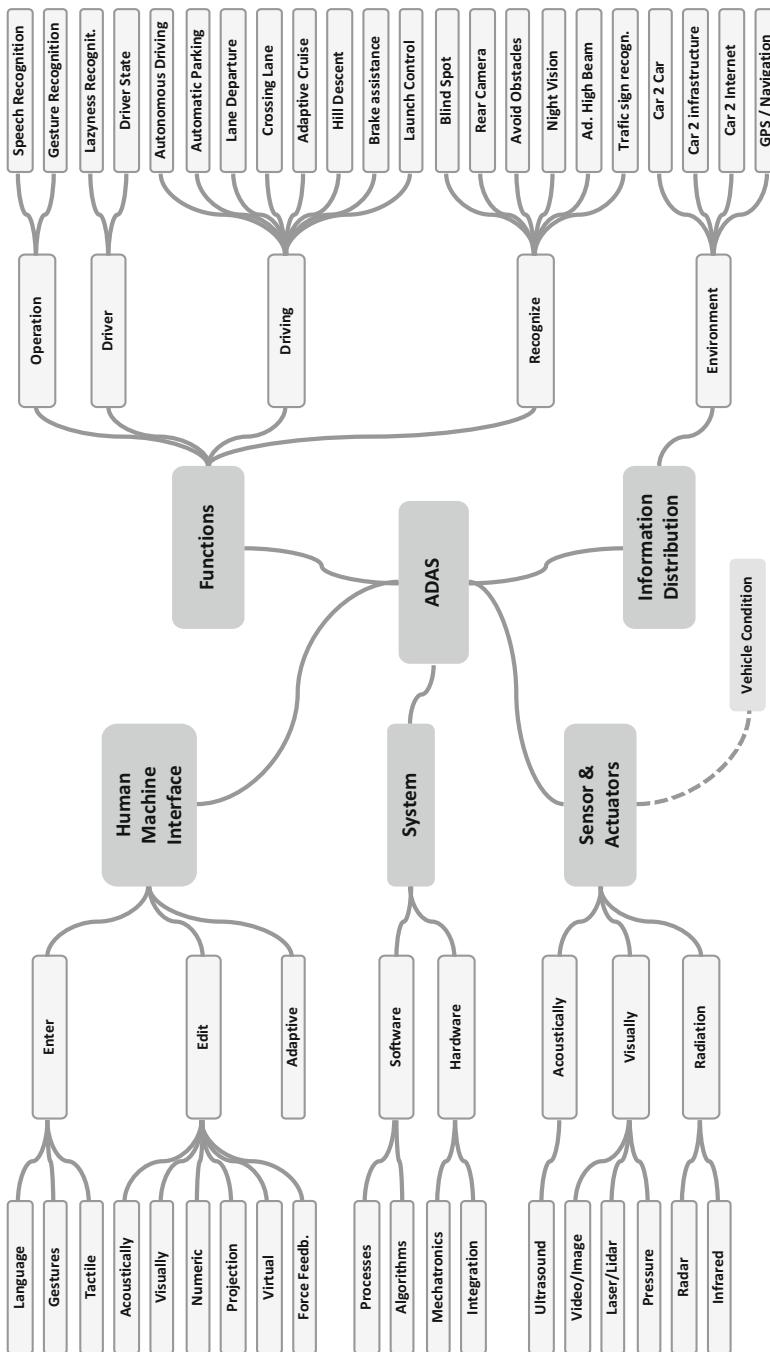


Fig. 11.18 Mindmap for ADAS and autonomous driving (Müller and Haas 2014)



Fig. 11.19 VW's study of an autonomous car at the IAA ([URL11 2017](#))



Fig. 11.20 Human driver uses a steering wheel to control lateral movements

Fig. 11.21 Steering wheel automatically retracts if one switches to autonomous driving mode



Fig. 11.22 If the car drives autonomously, the steering wheel is retracted and does not occupy space in the interior design

Fig. 11.23 Sensor on top of a car to map surroundings for HD maps



The key challenge for any automated driving system is to manage and combine the significant amounts of data coming from the different sensors and to create a consistent model from this data which can be used to make decisions about driving behavior. A common solution to this problem is the creation of hierarchical sensor fusion architectures as described in Balani (2015).

Most sensors are equipped with a dedicated processing unit that creates a digital representation of the raw, often analog sensor data. Figure 11.23 shows LIDAR sensors on top of a car which are used to generate high-definition 3D maps for TomTom (URL8 2017).

Sensor fusion combines the outputs of multiple sensors. For example, the data from two cameras can be combined to extract depth information (also known as stereo vision). Similarly, data from different sensor types with overlapping fields of view can be merged to improve object detection and classification and to create a more precise model (Balani 2015; URL5 2017).

It is also possible to add data from external systems which feed their data into the cloud giving access to detailed map, traffic, and weather data. The addition of data from a V2X gateway is also possible. The result is a detailed 3D map of the car's surrounding environment (Balani 2015). This map is object-based and includes lane markers, other vehicles, pedestrians, cyclists, street signs, traffic lights, and so on. The detailed map is embedded within a larger, less detailed map which is required for navigation. Both model perspectives are updated in real-time, although at different intervals creating a virtual reconstruction of the real world based on sensor data (Balani 2015).



Fig. 11.24 Daimler's EQ vision of an electric and autonomous car

Short-range communications technology such as vehicle-to-vehicle and vehicle-to-infrastructure communication, collectively referred to as V2X, can be effectively applied to complex driving environments to enhance the safety of autonomous vehicles. V2X technology can supplement onboard sensors to gather and transmit environmental data, enabling the car to, for example, peer around corners and negotiate road intersections better than a human driver would (URL5 2017).

V2X technologies, which are today being developed in parallel with self-driving technologies, will enhance the performance and overall safety of autonomous vehicles (URL5 2017). The complexity of the driving environment will likely govern the introduction sequence of partially autonomous features as well. For instance, highways are less complex driving environment than urban streets or parking lots, which are full of non-standard infrastructure and involve a high level of interaction with other vehicles, pedestrians, and objects (URL5 2017).

Also, low-speed environments, such as traffic jams, may present fewer risks than high-speed driving scenarios. Clearly, autonomous driving requires huge investments and only a few companies can develop the technology in-house (Schaal 2017) which is one of the reasons why so many new alliances are coming up (Freitag 2016).

All these factors will influence the pace of adoption over the coming years. The technology will not gain commercial scale overnight; and, in fact, it may take several years before OEMs will be able to offer autonomous features at a price that is both acceptable to consumers and profitable for the manufacturer (Grünweg 2016; Maurer et al. 2015; URL11 2015). Form (2015) gives an overview of timelines and makes a prediction when the first fully autonomous cars can be seen on the road. The challenges can be clearly seen from the experience with Tesla's autopilot function (URL15 2016; Becker 2016).

11.5 Regulations, Public Acceptance, and Liability Issues

The previous section focused on technical challenges of autonomous vehicles. This section briefly deals with equally important aspects, like the regulatory frameworks, public acceptance, and liability issues (URL11 2015).

As with any new method of transportation, the regulatory environment plays a crucial role in its adoption. Public opinion has a significant impact too. In addition, there must be greater clarity on issues of liability. The incidents with Tesla have shown how sensitive the topic can be (Becker 2016).

11.5.1 Regulations and On-Road Approval

Autonomous driving on public roads is currently restricted by law. According to the Vienna Convention of Road Traffic (URL17 2017), which was ratified by over 70 countries as the foundation for international road traffic regulation, a driver must be present and in control of a moving vehicle at all times (URL8 2015). This clearly would not allow any self-driving vehicles to be deployed.

In May 2014, an expert committee of the United Nations added a new rule to the Vienna Convention: *Systems that autonomously steer a car are permissible if they can be stopped by the driver at any time.*

This recent addition represents a significant step forward in the development of automated driving, and several countries are now reviewing national legislation to allow self-driving vehicles in specific circumstances where automated technology has been proven to be sufficiently mature and safe. Early policy reviewers and therefore the countries that are likely to feature early adoption include the USA, UK, and New Zealand (URL8 2015).

It is important to note that the USA has not signed the Vienna Convention. Recent statements by the National Highway Traffic Safety Administration (NHTSA), like the response to Google's inquiry about self-driving cars, indicate a favorable position regarding self-driving technologies (URL10 2016; URL18 2017).

However, NHTSA admits that there is still a lot of work to be done in order to come up with a comprehensive legal framework for autonomous driving.

11.5.2 Toward a Statutory Framework for Autonomous Driving

The ministry of transportation in Germany and its counterparts in Europe are currently working on the statutory framework that would allow piloted and highly automated driving (Form 2015).

The governments still have to pass a comprehensive legislation. The framework is complex as it has to deal with various new issues like ethical dilemmas, etc. (Maurer et al. 2015).

Several traffic rules and regulations will need modifications:

- Road Traffic Act
- The highway or autobahn code
- Driving license regulations
- Road Traffic Licensing Regulations

Associations like VDA (German Automotive industry) and ACEA (European Automobile Manufacturers Association) are discussing the issue too. Both organizations have published white papers outlining their position on the matter ([URL9 2016](#); [URL16 2015](#)).

11.5.3 Acceptance of Autonomous Driving and Ethical Difficulties

Several studies have attempted to measure public acceptance of autonomous driving, with substantial research undertaken in Germany and the USA (Maurer et al. [2015](#); [URL8 2015](#)). Findings suggest that people have mixed feelings about self-driving vehicles. On one hand there are many advantages, like saving valuable time, less accidents, and mobility for the elderly and visually impaired; and on the other hand, the technology has to mature, the legal framework is not yet clear, and self-driving cars will certainly have a disruptive effect on the logistics, transport, and mobility industry. Already, driverless trucks are being tested on the roads of Nevada, and ride-hailing companies experiment with autonomous taxis. If there are no adequate replacement jobs for drivers of trucks, taxis, busses, and so forth, there will be a strong opposition (Eckert [2016](#); Haas [2014](#)).

Accidents due to system's limitations like Tesla's autopilot failure have attracted a lot of attention, even though humans might have failed in a similar scenario and human behavior is one of the main causes for severe accidents (Form [2015](#); Seeck [2015](#)).

The biggest hurdle to public acceptance is probably ethics. For self-driving vehicles one has to define in detail how the vehicle will react in various situations, recognizing that passengers, other road users, and pedestrians could be hurt because of the self-driving vehicle's decision ([URL8 2015](#)).

As vehicles gradually become more automated, liability is a further concern that must be addressed. If a self-driving vehicle is involved in a road traffic accident, who is liable for the damage caused: the driver of the vehicle, the vehicle owner, or the manufacturer?

At present, liability is based on the premise that the person using the vehicle is responsible for its safe operation ([URL8 2015](#)).

According to current regulatory frameworks (see Maurer et al. [2015](#)):

- Liability for damage to property and person is with the driver or vehicle owner.
- Liability for the vehicle—accountability for manufacturing errors including constructional defects, manufacturing defects, and faulty instructions—is with the manufacturer.

The key is a modification of traffic regulation stating that the driver does not violate his obligations and it can not be regarded as negligence if the control task is transferred to the system. This implies that the driver cannot be prosecuted according to criminal law if the accident is caused by technical failure. The financial liability will be taken care by the automobile liability insurance.

In this regard, it will be important to have devices that record any relevant information as there is no human driver to interrogate and report as a witness. This of course also raises data protection and privacy issues (Jung and Kalmar 2015; Reuter 2015).

Currently the insurance products for self-driving vehicles are, however, nonexistent (URL8 2015). In the context of a fully self-driving vehicle, liability has to be reconsidered. There can be no such thing as driver liability. And so long as the vehicle owner can prove correct maintenance of the vehicle, liability for damage to property and person shifts to the manufacturer (URL8 2015).

There are a couple of options for manufacturers to reduce their liability. The key question, however, is if a wrong decision on the part of the automatic controller could be considered to be a product defect, which would result in a liability of the OEM. Similar discussions are ongoing regarding robots—could one press charges against a robot?

The manufacturer may be able to insure its liability, and this will require the support of insurance providers convinced by the evidence that autonomous driving results in fewer, and less lethal, accidents than conventional driving (URL8 2015).

Clearly, it will be necessary to overcome obstacles of regulation, public acceptance, and liability can be deployed. Insurance companies are now recognizing the need to be prepared. By modifying and extending the existing terms of insurance, vehicle insurance companies may play a crucial role in accelerating the adoption of autonomous vehicles (URL8 2015).

11.5.4 Test on the Autobahn

Some countries have already addressed issues of regulation allowing self-driving vehicles on their streets, at least for initial trials. The technology is seen a competitive advantage and nobody wants to stay behind. In the US, California and Nevada have passed special legislations for autonomous vehicles.

Also, Germany wants to be among the leaders in autonomous driving. The Ministry of Transportation and Digital Infrastructure (BMVI) has developed a comprehensive strategy that deals with all relevant questions and issues.

The digital test field “Autobahn” on the A9 in Bavaria is a test environment for experimenting with connected and automated driving, (see Fig. 11.25). The test track is open for all innovative companies, OEMs, suppliers, IT product companies, and research institutions. The goal is a certificate “Tested on German Autobahn” (URL15 2017). The test track is close to Audi’s headquarter in Ingolstadt. Figure 11.27 shows Audi’s concept car which was presented at the IAA 2017.

Test drives and the gathering of huge amounts of data are extremely important to validate higher levels of autonomous driving (Pickhard 2016).

Fig. 11.25 Germany allows testing of autonomous vehicles on a stretch of the A9 motorway



11.6 E/E Architectures and Middleware for Autonomous Driving

Higher-level ADAS functions and autonomous driving require a fundamentally different E/E architecture (Forster 2014; Hudelmaier and Schmidt 2013; Kern 2012; Lang 2015; Weiß et al. 2016). The size of program code required will increase significantly, and the amount of data transferred on the bus systems grows exponentially as vehicle manufacturers move from ADAS to partial autonomy, and finally to full autonomy (URL8 2016).

In Fig. 11.26 the E/E architecture and ECU topology of a modern car with various bus systems and automotive Ethernet are shown. The ADAS features form a special domain that is connected with a high-speed Ethernet bus system. Ethernet was developed in the computer network domain in the 1970s and has now become feasible for vehicle applications (e.g., camera-based ADAS) through the Broadcom R-Reach technology with unshielded twisted pair wires (Arndt 2015; Matheus and Königseder 2015).

Many proven technologies from computer networks, like TCP/IP, could be transferred from this field to automotive E/E and deployed in this context (Ernst 2016; Matheus and Königseder 2015; Schaal 2012; Schaal and Schwedt 2013; Weber 2013; Weber 2015).

ADAS sensors like cameras provide huge amounts of data, and the hard real-time requirements of higher levels of autonomy demand high throughput, short latencies, and a high degree of flexibility.

On the CAN bus the signals are passed to all connected ECUs, regardless if the information is relevant or not (Streichert and Traub 2012). Modern automotive E/E architectures rely on high-speed, scalable bus systems and a so-called middleware which handles the complex interaction between the different communication partners (sensor nodes, electronic control units and actuators).

ADAS systems require a carefully designed, reusable, and scalable automotive software system which can be achieved by service-oriented architectures (SOA). The transition to autonomous driving will generate an even bigger need for software architecture. Currently, there is a lot of research going on in the field of ADAS software architectures (Fürst 2016; Lamparth and Bähren 2014; Thiele et al. 2013; Wagner 2015 and URL12 2016).

Service orientation is a well-accepted standard in classical software development (Balzert 2011; Schäfer 2010). It offers interesting possibilities to partition and structure ADAS software functions (Wagner 2015). To enable bandwidth efficiency, automotive IP networks—in contrast to static CAN communication—are set up in a dynamic and service-oriented way (Schaal 2012).

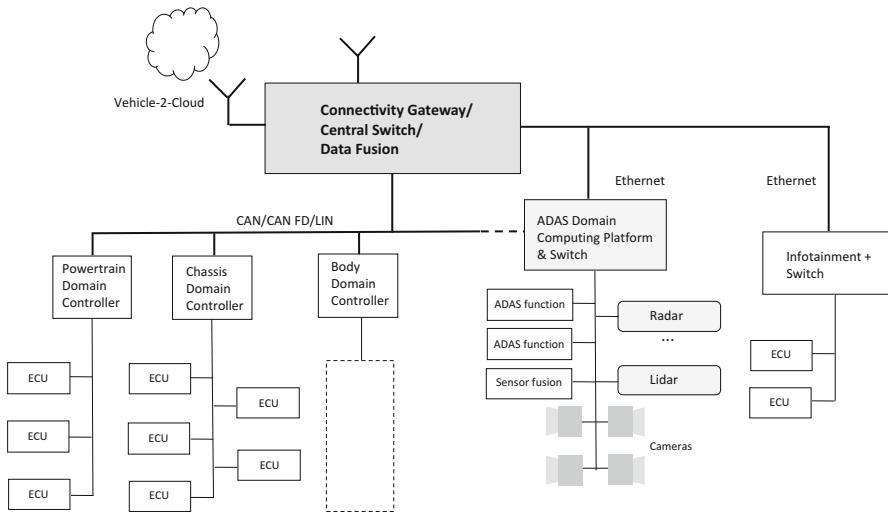


Fig. 11.26 A vehicle E/E architecture for ADAS and autonomous driving



Fig. 11.27 Audi presented a concept car for autonomous driving at the IAA 2017

The middleware plays a very important role in the implementation of sophisticated ADAS functions and autonomous driving capabilities. Figure 11.28 illustrates a typical ADAS software architecture. The top layer consists of various ADAS functions like LKA and AEB. These software modules communicate with a

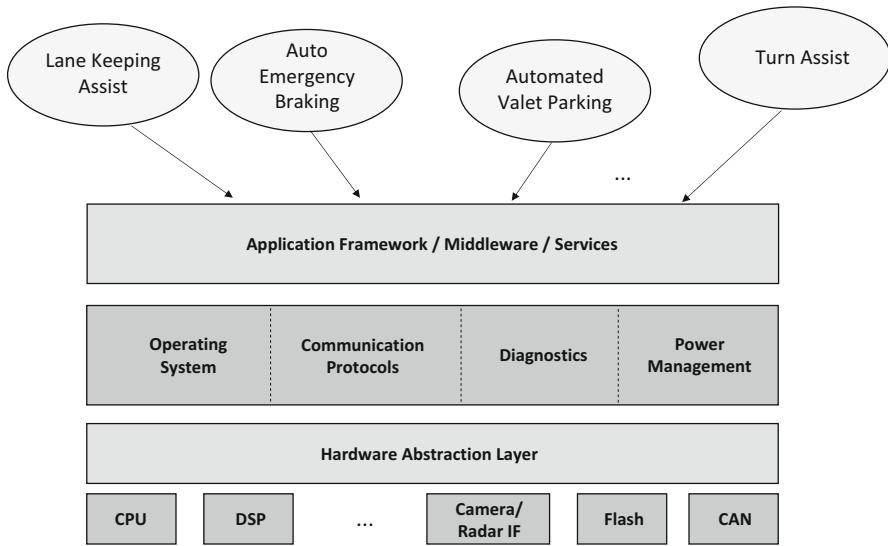


Fig. 11.28 Middleware for ADAS functions

middleware that orchestrates the interaction and takes care of managing the information flow. The next layer consists of the operating system drivers, communication protocols, diagnostics, power management and core algorithms for signal processing and sensor fusion. The actual hardware forms the lowest layer. This layer is encapsulated by a hardware abstraction layer (HAL); therefore that the functional layers do not interact with the hardware directly (Tanenbaum and Bos 2015).

There are many middleware concepts which are studied in the computer science field of distributed systems (Schill and Springer 2012; Silberschatz et al. 2010; Tanenbaum and Van Steen 2017). However, the utilization in the automotive domain is restricted by cost, performance, reliability, functional safety, and real-time constraints. The middleware for ADAS and autonomous driving functions should carefully balance the traffic on the network, using broadcast, public, and subscribe filters to restrict the distribution of unnecessary information.

BMW has developed and specified Scalable Service Oriented Middleware over IP (SOME/IP), which is an open protocol that fulfills automotive requirements offering publish and subscribe and remote procedure call mechanisms (Matheus and Königseder 2015; URL11 2016). Figure 11.29 illustrates the concept. The SOME/IP middleware uses a TCP/IP stack on an Ethernet connection (BroadR-Reach or T-Base (Arndt 2015)). If a software component or application needs to call a remote function across ECU boundaries, the middleware will establish a TCP/IP-based client/server communication.

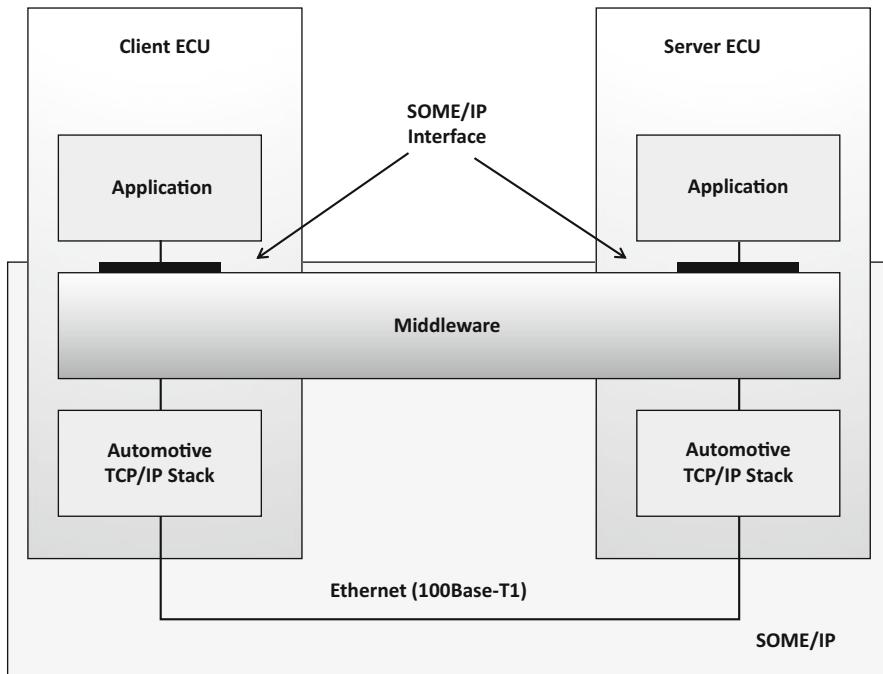


Fig. 11.29 SOME/IP (URL11 2016)

SOME/IP offers interfaces for service-oriented communication. This distinguishes it from the pure signal-based broadcast communication systems like CAN (URL9 2015; URL11 2016).

SOME/IP interaction is roughly subdivided into three areas: service discovery (SD), remote procedure call (RPC), and access to process data. SD lets ECUs find services or offer their services in the network which are accessed via RPC as shown in Fig. 11.30. In addition to that, it is possible to set up notifications for specific events.

Figure 11.30 depicts the communication pattern in SOME/IP. The RPC mechanism allows a classic remote procedure call across the communication network. It abstracts from all details like finding the server process/application, and managing the data transfer. Apart from this, there is a publish/subscribe mechanism which notifies applications if an event is detected. Only those applications that have been registered will be informed. This allows for a careful utilization of bandwidth. In the context of ADAS or autonomous driving, it could mean, for example, a specific function would only register for those events which are relevant and would only be notified if the corresponding sensors would have generated such an event. Publish/subscribe mechanisms are also very helpful if the system is being reconfigured, or in the scenario of a graceful downgrading of subsystems which do not work properly and have to be shut off (Matheus and Königseder 2015).

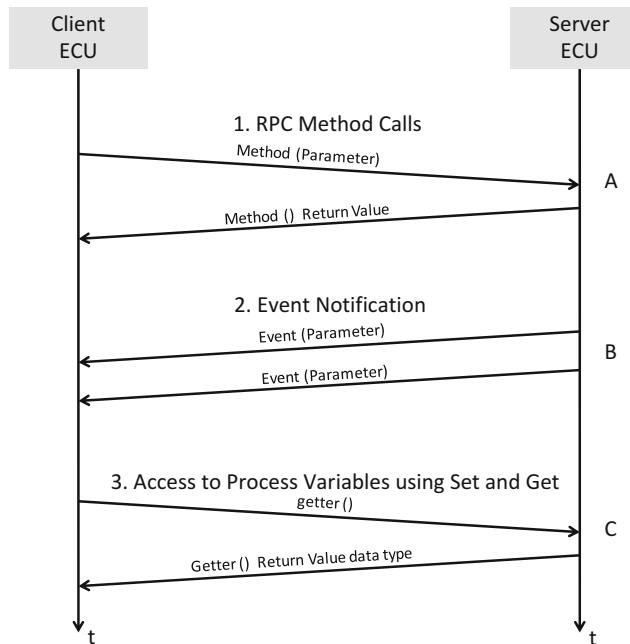


Fig. 11.30 Communication in SOME/IP ([URL11 2016](#))

Besides SOME/IP, which is already an integral part of AUTOSAR, several other middleware standards are currently being used in autonomous cars, chief among them ([URL10 2015](#); [URL11 2016](#); [URL19 2017](#); [URL20 2017](#)):

- Data Distribution Service (DDS)
- Automotive Data and Time-Triggered Framework (ADTF)
- Robot Operating System (ROS)
- Common API (GENIVI/Open Source)

Data distribution service (DDS) is a middleware standard which was defined by the Object Management Group (OMG). The company or vendor RTI provides an implementation which they call Connex DDS, specifically for applications in autonomous driving ([URL20 2017](#)).

The ADTF framework is very popular in Germany, for example in Volkswagen and Audi. ROS was developed as a standard operating system for robots. It supports many robot-specific features, like standard message definitions for robots, robot geometry library, robot description language, preemptable remote procedure calls, diagnostics, localization, mapping, and navigation ([URL19 2017](#); [URL20 2017](#)) which can be directly used in autonomous driving.

The AUTOSAR standardization committee is currently reexamining communication middleware ([Fürst 2016](#); [Weber 2013](#); [URL23 2017](#)) to embed further middleware standards in the AUTOSAR stack.

11.7 Cybersecurity and Functional Safety

ADAS and autonomous driving are potential targets for cyberattackers, with potentially far reaching consequences.

Cybercriminals and hackers could exploit various cyberattack surfaces like:

- In-vehicle Infotainment System (IVI)
- Telematic Control Unit (TCU)
- V2X communication infrastructure (V2V, V2I)
- Connected smartphones and mobile apps
- Connection between smartphone/key and vehicle
- Software stacks, e.g. middleware (exploiting design flaws, backdoors, etc.)
- Over-the-air updates of software and firmware (SOTA, FOTA)
- ADAS sensors
- HW/SW supply chain (backdoors, spy chips, HW vulnerabilities, software flaws, etc.)
- Data recording devices
- Backend systems
- GPS devices (compromising the localization)
- OBD-II port/remote diagnostics
- Electric powertrain (battery systems, charging infrastructure, communication protocols between charging infrastructure and vehicle)

The complexity of the software in an autonomous vehicle creates a special threat which is well known from operating systems. There might be design flaws that are known to a few developers, so-called zero day vulnerabilities, which could be exploited for a zero-day attack.

Also the middleware could be a target for cyberattacks. In this regard, Herold et al. (2016) and Wolf et al. (2015) discuss cybersecurity issues of SOME/IP.

Another mode of attack, which is especially harmful for self-driving cars, is the direct attack on sensors like blinding cameras, confusing camera auto control, relaying or spoofing signals.

Malware could be introduced through various ports of the network and could compromise the attached subsystems, actuators, and sensors. In such a scenario, vehicles could receive false signals, intentionally misguiding them.

Autonomous cars rely on sophisticated machine learning technologies. If one knows the algorithms, their sensitivity to noise and potential vulnerabilities, this could be exploited, increasing the overall risk.

Cybersecurity flaws will have an immediate impact on the functional safety.

Functional safety is described by specific automotive safety integrity levels (ASIL) (URL4 2016). This was discussed in detail in Chap. 4. The ASIL scheme distinguishes four ASIL levels, A to D, according to the severity of a failure (URL5 2016). The functioning of the rear camera, for example, refers to ASIL B, anything affecting the breaks is ASIL D.

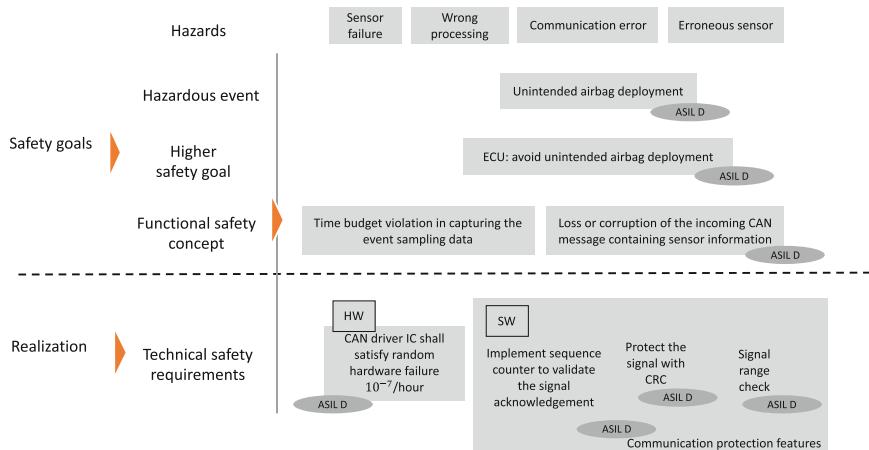


Fig. 11.31 Functional safety and automotive software (see Vivekanandan et al. 2013)

KPIT has developed a framework for handling safety goals (Vivekanandan et al. 2013). Figure 11.31 illustrates this approach for the event of unintended airbag deployment if a special child seat is present. The safety goal (SG) is to prevent such an unintended inflation of the airbag which is classified as a severe event falling into the category of ASIL level D. The functional safety concept is to prevent and mitigate a loss of the incoming CAN message that contains this information. On the implementation level, there are HW and SW requirements to fulfill. The CAN driver has to operate within a specific failure mode for random hardware failure and the signals have to be checked for range and integrity, for example, by a cyclic redundancy check (CRC). Furthermore, a sequence counter will validate that the signal is acknowledged in a given time frame.

A cyberattack on the CAN bus could scramble up and delay such a signal too and lead to wrong values. Hence, a similar approach could be employed to check the validity and timing of the signal value, even though there is no hardware failure or glitch in timing but a cyberattack that leads to functional safety concerns.

In Fig. 11.32 the concept of simultaneous engineering of cybersecurity and functional safety is shown which has been proposed by various researchers (Nause and Höwing 2016; Serio and Wollschläger 2015; URL26 2017) as cybersecurity breaches can have an immediate effect on the functional safety of automotive systems (Klauda et al. 2015; Solon 2015).

This is why it is crucial to integrate cybersecurity as a design goal into the automotive R&D process (Haas and Möller 2017; Mahaffey 2015a, b; Nause and Höwing 2016; Sushravya 2016). This approach is also reflected in the comprehensive framework of Fig. 11.33.

As outlined in the previous sections, autonomous driving requires a close integration of information from various sources, for example, sensors, infrastructure, other cars, cloud, and so forth. The reliability, availability, and cyber-safety are

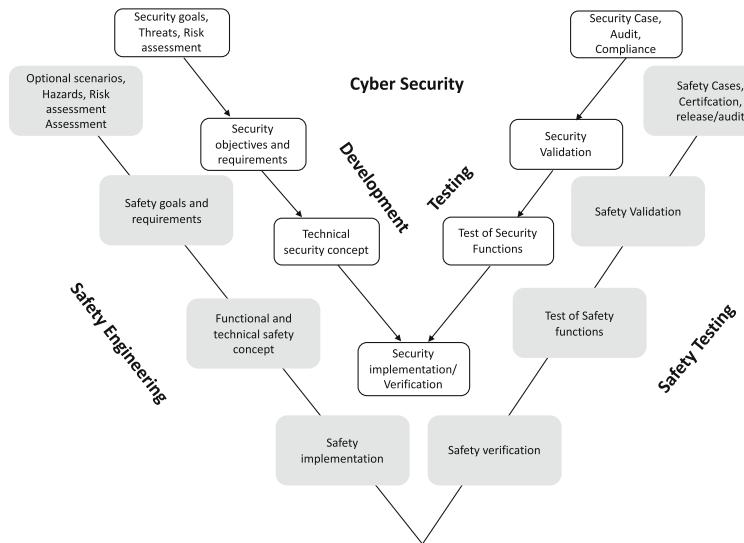
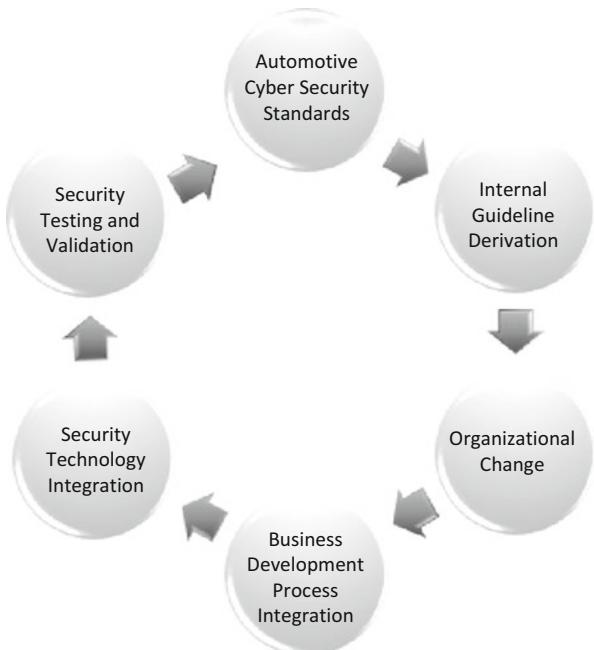


Fig. 11.32 Functional safety and cybersecurity (see Klauda et al. 2015; URL26 2017)

Fig. 11.33 Cybersecurity needs a holistic approach



important design aspects. Serio and Wollschläger (2015), Wolfsthal and Serio (2015) and Currie (2015) analyze the cyber threats to connected cars and suggest various solutions for intrusion detection.

Cybersecurity for autonomous cars can only be achieved with a holistic and integrated approach (Pickhard et al. 2015; Serio and Wollschläger 2015; Weimerskirch 2016) as discussed in Chap. 6 where one has to look at hardware (HW) failures, software (SW) vulnerabilities with regard to cybersecurity and system safety simultaneously. Reuter (2015) argues that functional safety is closely related with the security of data in a modern car and warns of the liability consequences if this topic is not dealt with properly.

If a cyber attack happens involving a new malware or attack vector, it is important to react swiftly and to update the signatures of intrusion detection and prevention systems as soon as possible. In this regard, over-the-air updates and security operation centers are crucial to fight cyberattacks (Zetter 2015; Brisbourne 2014). Unfortunately, over-the-air updates themselves are prone to attacks and require sophisticated cryptographic methods and key management to avoid security breaches.

11.8 Summary, Conclusion, and Recommended Readings

This chapter discussed *advanced driver assistance* functions and *autonomous driving*. ADAS has been around for a while. Adaptive cruise control, emergency breaking, blind spot detection, lane departure warning and lane keeping assist, as well as remote and automatic parking are just a few of many ADAS functions which make a car safer and easier to drive. As an example, we discussed the lane keep assistance function in a Seat Leon car.

Camera sensors are an inexpensive yet efficient way of sensing the environment. The chapter gave an overview of image processing and analysis for camera-based advanced driver assistance systems and showed how these algorithms can be rapidly implemented with MATLAB/Simulink. Automatic code generation is also possible on various target platforms. Hardware-in-the-loop (HIL) systems have evolved to test ADAS functions, and they will play an important role in the reliability analysis of higher automation and autonomous driving.

Non-functional requirements in ADAS like flexibility, modularization, responsiveness, reliability, testability, security, and so forth have had an impact on automotive E/E and software architectures. Middleware technologies like SOME/IP provide the flexibility and scalability needed. SOA and Message-oriented middleware (MOM) help to carefully utilize the bandwidth of the bus system.

The legal framework for autonomous driving has to still evolve, but there is a lot of pressure as countries do not want to fall behind in such an innovative domain. The key is the modification of the Vienna Convention which requires a human to be in charge at any time. Also, ethical issues have to be dealt with, and liability and insurance models have to adapt.

Autonomous driving is based on connectivity, complex software systems, over-the-air updates, and vehicle-to-backend communication. All this implies complex attack surfaces.

Functional safety and cybersecurity are intimately connected. Reliable autonomous cars will only be possible by adopting a “design by security approach” as has been proposed by various researchers (Haas and Möller 2017; Pickhard et al. 2015; Weimerskirch 2016; URL1 2015). Furthermore, cybersecurity for ADAS and autonomous driving requires a holistic approach.

11.8.1 Recommended Reading

Form (2015) outlines different developments in autonomous driving.

Markoff (2016) and Menn (2016) discuss the current interest in artificial intelligence in the context of drones, robots and autonomous cars, a research field that attracts huge fundings.

More information on software architectures for drivetrain control can be found in Orth et al. (2014).

The books (Silberschatz et al. 2010) and (Tanenbaum and Bos 2015) give a good overview of middleware from an operating system perspective.

Vahid and Givargis (2001) discuss the HW implementation of image processing in a case study of a digital camera.

Steinmüller (2008) is a compact introduction to the mathematics of image processing and image analysis.

Köncke and Buehler (2015) discuss the impact of cyberattacks on the industry in general.

The huge investments into autonomous driving can only be handled by a few OEMs (Schaal 2017), most of them need help from partners and strike new alliances. Eckl-Dorna (2016) looks at the cooperation between FCA and Google. More information on the Apple Titan project can be found in Sorge (2017).

Hunt et al. (1996) present a control algorithm for the lateral control of autonomous vehicles based on a network of local model (operating regime)-based controllers. A detailed overview of modeling, simulation, and control of vehicle driveline is given in Kiencke and Nielsen (2005).

For more information on digital control and adaptive control refer to Astrom and Wittenmark (1996), Ogata (2004), and Dorf and Bishop (2010).

Prostinetz and Schimansky show how vulnerable we have become to the availability of the cloud discussing the partial shutdown of the AWS infrastructure in March 2017. This is also important for ADAS functions and autonomous driving, as some algorithms might run in the cloud.

Reuss et al. (2015) discuss the synergies between electro-mobility and autonomous driving.

Paar (2015) emphasizes the importance of cryptography to protect automotive E/E architectures. URL5 (2017) refers to the Spy Act which has triggered a lot of activities in automotive cybersecurity in the US. URL1 (2015) and URL 4 (2015),

(URL7 2015), and (URL16 2016) provide a good introduction to the field of automotive cybersecurity.

11.9 Exercises

What ADAS functions are you aware of, and how would you categorize them?

What are the different levels of driver assistance?

What is the difference between conditional automated and highly automated driving?

What research initiatives in autonomous driving are you aware of?

Who are the leading suppliers in ADAS systems from your viewpoint?

Who will cooperate with whom (OEM, suppliers, service providers, etc.)?

What are the biggest hurdles for autonomous driving from a technology viewpoint?

What sensors are important in ADAS?

What is the relationship between autonomous/piloted driving and automated valet parking? What are the commonalities and what are the differences?

What role do maps play in autonomous driving? Who will provide these maps? What are the collaboration models?

Which R&D processes and safety guidelines (e.g., ISO 26262, ASIL) are being used in ADAS development?

What associations, bodies, and committees are dealing with the regulatory and standardization framework of ADAS and autonomous driving?

What are the biggest obstacles for autonomous driving from a legal viewpoint?

What legal framework is applicable for autonomous driving? What modifications are needed in the future?

What are the current state laws and legislative activities? What federal or state liability legislation is needed?

What is the Vienna Convention on Road Traffic?

What are the recent modifications to this framework which will lead the way to highly automated driving?

What liability questions arise in autonomous driving?

What are the biggest safety concerns regarding ADAS and autonomous driving? How will safety be handled?

What safety standards are applicable? How does ADAS safety differ from safety for piloted driving?

What are the social implications of autonomous driving?

Will people still want to drive themselves if autonomous cars are widely available?

In which countries will autonomous driving be introduced first?

In what of the following areas will autonomous driving be deployed first (logistics, public transport, individual transport)?

What role will autonomous driving play for carsharing, ridesharing, electric cars?

Compare the different strategies for autonomous driving in high-tech companies like Google, mobility service providers like Uber, and classical automotive OEMs.

Read about the AUTPLES project and write a short report about it.

What is the relationship and what are the synergies between autonomous driving and electro-mobility?

Comment on Tesla's approach to ADAS and autonomous driving.

Comment on the importance of cybersecurity in autonomous driving?

What is the relationship between functional safety and cybersecurity?

What are the biggest challenges in autonomous driving?

What role does V2X communication play for autonomous driving?

What are typical mobility use cases for autonomous driving?

Comment on the costs of autonomous driving. Will it be affordable?

Please compare OEMs strategies and announcements in autonomous driving.

How do Indian automakers position themselves?

What are the synergies between robotics and autonomous driving? What can we learn from other industries, e.g., aircrafts?

Please write a short report on potential cyberattacks on autonomous cars.

What are the most critical cybersecurity vulnerabilities of autonomous cars?

What solutions are available to make autonomous cars cybersecure?

Review the recent reports about cyberattacks on cars. What is the relevance for cybersecurity of autonomous cars?

What impact on traffic accidents will autonomous cars have?

Please comment on the complexity of autonomous driving from a HW and SW perspective.

What are the major challenges for the engineers of autonomous cars?

What impact will autonomous cars have on jobs in the transport business?

How do insurances prepare for autonomous driving?

What happens if autonomous cars move from one country to another?

Describe the ethical dilemma situation with driverless cars.

Please comment on the reliability of sensors and computational elements, aging, and wear and tear in ADAS systems.

What does the term robotification mean?

What impact does autonomous driving have on the interior design of a car?

References and Further Readings

(Abraham et al. 2016) Abraham, B., Brugger, D., Strehlke, S., Runge, W.: Autonomous Driving – Only a Trojan horse of Digital Companies?, ATZ elektronik, 01/2016

(Alheeti et al. 2015a) Alheeti, K. M. A., Gruebler, A., McDonald-Maier, K. D.: An intrusion detection system against malicious attacks on the communication network of driverless cars. In: 12th Annual IEEE Consumer Communications and Networking Conference (CCNC), Pages 916–921, 2015

(Alheeti et al. 2015b) Alheeti, K. M. A., Gruebler, A., McDonald-Maier, K. D.: An Intrusion Detection System against Black Hole Attacks on the Communication Network of Self-Driving Cars. In: 6th International Conference on Emerging Security Technologies (EST), Pages 86–91, 2015

(Arndt 2015) Arndt, C.: Developments of Automotive Ethernet Technologies - Introduction to the BroadR-Reach Technology and beyond, Continental & VDI Wissensforum, 06/2015

- (Astroem and Wittenmark 1996) Astrom, K., Wittenmark, B.: Computer Controlled Systems. Prentice Hall, Information and Systems Series, 1996
- (Balani 2015) Balani, N.: Enterprise IoT – A Definite Handbook, self-published, Kindle Edition, 2015
- (Balzert 2011) Balzert, H.: Textbook of Software Engineering: Design, Implementation, Deployment and Operation (in German), Springer Spektrum Publ., 2011
- (Beck 2016) Beck, T.: Do we need Autonomous Driving? (in German) elektronik.net, 01/2016, pp. 48–49. 01/2016
- (Becker 2016) Becker, J. Autopilot of Tesla – in a Tesla the risk is a standard feature (in German). Süddeutsche Online. November 17th 2016. Available from: <http://www.sueddeutsche.de/auto/autopilot-von-tesla-bei-tesla-ist-das-risiko-serienmaessig-1.3252192>
- (Bekey 2005) Bekey, G. A.: Autonomous Robots, Massachusetts Institute of Technology, 2005
- (Besenbruch 2014) Besenbruch, D.: Electronic Systems – Protection of manipulation, ATZ elektronik, 2014
- (Beynon et al. 2003) Beynon, M., Hook, D., Seibert, M., Peacock, A., Dudgeon, D.: Detecting Abandoned Packages in a Multi-camera Video Surveillance System. IEEE International Conference on Advanced Video and Signal-Based Surveillance, 2003
- (Bhattarcharjee 2013) Bhattarcharjee, S.: Efficient Algorithm for Crossing Alert in Camera-based Advanced Driver Assistance Systems, Master of Technology Thesis, International Institute of Information Technology Bengaluru (IIIT-B), 2013
- (Bose 2004) Bose, T.: Digital Signal and Image Processing, John Wiley and Sons, 2004
- (Bridges 2015) Bridges, R.: Driverless Car Revolution Buy Mobility – Not Metal. Self-published, 2015
- (Brisbourne 2014) Brisbourne, A.: Tesla's Over-the-Air Fix: Best Example Yet of the Internet of Things? Wired online. February 2014. Available from: <http://www.wired.com/insights/2014/02/teslas-air-fix-best-example-yet-internet-things/>
- (Chucholowski and Lienkamp 2014) Chucholowski, F., Lienkamp, M.: Teleoperated Driving – Secure and Robust Data Connections, ATZ elektronik, 01/2014
- (Corke 2011) Corke, P.: Robotics, Vision, and Control, Springer Publ., 2011
- (Currie 2015) Currie, R.: Developments in Car Hacking. <https://www.sans.org/reading-room/whitepapers/internet/developments-car-hacking-36607>, 2015
- (Davies 2012) Davies, E. R.: Computer and Machine Vision: Theory, Algorithms, and Practicalities, Elsevier Publ., 2012
- (Dorf and Bishop 2010) Dorf, R.C., Bishop, R.H.: Modern Control Systems, Pearson Education, 2010
- (Dudenhöffer 2016) Dudenhöffer, F.: Who will be put in the fast lane (in German). Campus Publ., 2016
- (Eckert 2016) Eckert, D.: Robots will destroy millions of jobs. Welt online. 27th August 2016. Available from: <http://www.welt.de/wirtschaft/article157872907/Roboter-werden-Millionen-Jobs vernichten.html?config14print#08.2016>
- (Eckl-Dorna 2016) Eckl-Dorna, W.: Savior instead of aggressor: Fiat Chrysler courts Google (in German). Manager Magazin online, April 29th 2016. Available from: <http://www.managermagazin.de/unternehmen/autodesk/roboterauto-allianz-warum-fiat-chrysler-mit-google-kooperieren-will-a-1090052.html>
- (Elgammal et al. 2000) Elgammal A., Harwood D., Davis L.: Non-parametric Model for Background Subtraction. In: Proceedings of the 6th European Conf. on CompVision-Part II, Pages 751–767, 2000
- (Elgammal et al. 2003) Elgammal A., Duraiswami R., Davis, L.S.: Efficient Kernel Density Estimation Using the Fast Gauss Transform with Applications to Color Modeling and Tracking, In: IEEE Transactions on Pattern Analysis and Machine Intelligence; Vol. 25 No. 11, 1499–1504, 2003
- (Ernst 2016) Ernst, R.: Automotive Ethernet – Opportunities and Pitfalls, Institut für Datentechnik und Kommunikationsnetze, ETFA 09/2016, Berlin, 2016

- (Fallstrand and Lindstrom 2015) Fallstrand, D., Lindstrom, V.: Automotive IDPS: Applicability analysis of intrusion detection and prevention in automotive systems. Master' Thesis. Chalmers University of Technology. Available from: <http://publications.lib.chalmers.se/records/fulltext/219075/219075.pdf>
- (Form 2015) Form, T.: Autonomous Driving – Quo vadis?, ATZ elektronik, special edition, 07/2015
- (Forster 2014) Forster, F.: Development Embedded Systems, ATZ elektronik, Vol 9, 01/2014, Pages14–18, Springer Vieweg, Springer 2014
- (Freitag 2016) Freitag, M.: Robotic cars – German manufacturers in pole position (in German). Manager Magazin online. July 26th 2016. Available from: <http://www.manager-magazin.de/unternehmen/autoindustrie/robotautos-deutsche-autobauer-fuehren-a-1104783.html>
- (Fürst 2016) Fürst, S.: AUTOSAR Adaptive Platform for Connected and Autonomous Vehicles, In: EUROFORUM Elektronik-Systeme im Automobil, 02/2016
- (Gaonkar et al. 2011) Gaonkar, P., Nanthini, S., Manoj, S., Mamilla, S.: Lane Departure Warning System, Class Paper, Car IT and Cybersecurity class, IIIT-B, 2011
- (Giachetti et al. 1994) Giachetti, A., Campani, M., Torre, V.: The use of optical flow for the autonomous navigation, In: Proc. 4th Euro. Conf. Comput. Vision, 1994
- (Grünweg 2016b) Grünweg, T.: Ford strategy – Autonomy for All (in German). Spiegel online. October 11th 2016. Available from: <http://www.spiegel.de/auto/aktuell/ford-plant-roboter-taxiflotte-wie-uber-a-1114025.html>
- (Gonzalez et al. 2008) Gonzalez R. C., Woods, R. E., Eddins, S. L.: Digital Image Processing Using MATLAB, Pearson Education, New Delhi, India, 2008
- (Gonzalez and Woods 2008) Gonzalez, R. C., Woods, R. E.: Digital Image Processing, 3rd Edition. Pearson/Prentice Hall Publ., 2008
- (Haas 2014) Haas, R.: Socio-Economic Impact of Autonomous Driving in Emerging Countries, India as example, European Radar Conference, EuRad, Rome, 2014
- (Haas and Möller 2017) Haas, R. E., Möller, D. P. F.: Automotive Connectivity, Cyber Attack Scenarios and Automotive Cyber Security. Proceed. IEEE/EIT 2017, pp. 635-639, ISBN: 978-1-5090-4767-3/17
- (Hanselman and Littlefeld 2008) Hanselman, D., Littlefield, B.: Mastering MATLAB 7, Pearson Education, India, 2008
- (Haykin 2009) Simon Haykin, Neural Network and Learning Machines, 3rd Ed., Pearson Education, Upper Saddle River, NJ, 2009
- (Herold et al. 2016) Herold, N., Posselt, S.-A., Hanka, O., Carle, G.: Anomaly Detection for SOME/IP using Complex Event Processing, Chair of Network Architectures and Services, Technical University Munich (TUM), Department of Computer Science, 2016
- (Hoffmann 2008) Hoffmann, D.: Software-Quality, Springer Publ., 2008
- (Horn and Schunk 1981) Horn K.P., Schunck, B.G.: Determining Optical Flow, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139, U.S.A, Pages 185-203, 1981
- (Hertzberg et al. 2012) Hertzberg, J., Lingemann K., Nüchter, A.: Mobile Robotics – An introduction from a computer science perspective (in German), Springer Vieweg Publisher, Berlin Heideberg, 2012
- (Hudelmaier and Schmidt 2013) Hudelmaier, P., Schmidt, K.: Chip Solutions For Driver Assistance Systems, ATZ elektronik 03/2013, Pages 48–52
- (Hunt et al. 1996) Hunt, K. J., Haas, R., Kalkkuhl, J.: Local Controller Network for autonomous vehicle steering, Control Engineering Practice, 1996
- (Jain 2000) Jain, A. K.: Fundamentals of Digital Image Processing, Prentice Hall Publ., 2000
- (Javed et al. 2002) Javed, O., Shah, M.: Tracking and object classification for automated surveillance, In: Proc. of ECCV, Pages 343–357, 2002
- (Johri 2016) Johri, S.: Attack Surfaces in Connected Cars, Class paper, Car IT and Cybersecurity class, IIIT-B, 2016
- (Johanning and Mildner 2015) Johanning, V., Mildner, R.: Car IT Compact, Springer Publ., 2015

- (Joshi 2009) Joshi, M. A.: Digital Image Processing - An Algorithmic Approach, PHI Learning, New Delhi, 2009
- (Jung and Kalmar 2015) Jung, C., Kalmar, R.: Re-interpret Data Security – the Data Gold and Business Models, ATZ elektronik, 04/2015
- (Kaplan 2016) Kaplan, J.: Artificial Intelligence, Oxford University Press, 2016
- (Karmann et al.1990) Karmann K.-P., Brandt A.: Moving object Recognition using and adaptive background memory. In: Time-Varying Image Processing and Moving Object Recognition, Pages 289–307. V. Cappellini, Ed: Elsevier Science Publishers, 1990
- (Kern 2012) Kern, A.: Ethernet and IP for Automotive E/E-Architectures – Technology Analysis, Migration Concepts and Infrastructure. Ph.D. Thesis, University of Erlangen-Nürnberg, Available from: <https://pdfs.semanticscholar.org/8106/455c487acc052cc701f48615f1172029a057.pdf>, Erlangen, 2012
- (Kiencke and Nielsen 2005) Kiencke, U., Nielsen, L.: Automotive Control Systems: For Engine, Driveline, and Vehicle. Springer Publ., 2005
- (Klauda et al. 2015) Klauda, M., Schaffert, M., Logospiris, A., Piel, G., Kappel, S., Ihle, M., Setting the Course for 2020 – change of paradigms in E/E architecture. ATZ elektronik, Pages 17–22, Springer Vieweg Publ., 02/2015
- (Köncke and Buehler 2015) Köncke, F. C., Buehler, B. O.: Cyber Attacks – Underestimated Risk for the German Industry (in German). Wirtschaftswoche online. November 9th 2015. Available from: <https://www.wiwo.de/technologie/digitale-welt/cyber-angriffe-unterschaetzes-risiko-fuer-die-deutsche-industrie/12539606.html>
- (Lang 2015) Lang, M.: High Degree of Integration of ADAS Functions into One Central Platform Controller, ATZ elektronik, Vol 10, 04/2015, Pages 40–43, Springer Vieweg Publ., 2015
- (Lamparth and Bähren 2014) Lamparth, O., Bähren, F.: From The Connected To The Autonomous Car. ATZ elektronik, Vol 9, 05/2014, Pages 36–39, Springer Vieweg Publ., 2014
- (Lu and Zhang 2007) Lu, S., Zhang, J.: Detecting unattended packages through human activity recognition and object association, PR Vol. 40, No. 8, Pages 2173–2184, 08/2007
- (Mahaffey 2015a) Mahaffey, K.: The New Assembly Line: 3 Best Practices for Building (secure) Connected Cars. Lookout Blog. August 6th 2015. Available from: <https://blog.lookout.com/tesla-research>
- (Mahaffey 2015b) Mahaffey, K.: Here Is How To Address Car Hacking Threats. TechCrunch. September 13th 2015. Available from: <https://techcrunch.com/2015/09/12/to-protect-cars-from-cyber-attacks-a-call-for-action/>
- (Markey 2015) Markey, E.J.: Tracking and Hacking: Security and Privacy Gaps Put American Drivers at Risk. 2015. Available from: https://www.markey.senate.gov/imo/media/doc/2015-02-06_MarkeyReport-Tracking_Hacking_CarSecurity%202.pdf
- (Markoff 2016) Markoff, J.: Artificial Intelligence Swarms Silicon Valley on Wings and Wheels, The New York Times online. July 17th 2016. Available from: <http://nyti.ms/2a0Awys>
- (Matheus and Königseder 2015), Matheus, K., Königseder, T.: Automotive Ethernet, Cambridge University Press, 2015
- (Maurer et al. 2015) Maurer, M., Gerdes, C. J., Lenz, B., Winner, H. (Ed): Autonomous driving, technical, legal and social aspects, Springer Vieweg Publ., 2015
- (Menn 2016) Menn, A.: Nvidia founder Huang: Artificial Intelligence triggers next Industrial Revolution (in German). Wirtschaftswoche online. 9.12.2016. Available from: <https://www.wiwo.de/technologie/digitale-welt/nvidia-gruender-huang-kuenstliche-intelligenz-loest-naechste-industrielle-revolution-aus/14951562.html>
- (Miller and Valasek 2014) Miller C., Valasek C.: A Survey of Remote Automotive Attack Surfaces. IOActive 2014. Available from: https://www.ioactive.com/pdfs/IOActive_Remote_Attack_Surfaces.pdf
- (Miller and Valasek 2015) Miller, C., Valasek, C.: Remote exploitation of an unaltered passenger vehicle. August 10th 2015. Available from: <http://illmatics.com/Remote%20Car%20Hacking.pdf>

- (Müller and Haas 2014) Müller, M. and Haas, R.: Study on Automotive Electronics, Magility GmbH, 2014
- (Nause and Höwing 2016) Nause, M., Höwing, F.: Functional Security as a Model for Software Development in Automotive Security, ATZ elektronik, 03/2014
- (Navet and Simonot-Lion 2009) Navet, N., Simonot-Lion, F.: Automotive Embedded Systems Handbook. CRC Press, 2009
- (Ogata 2004) Ogata, K.: Discrete-Time Control Systems, Pearson Education, 2004
- (Orth et al. 2014) Orth, P., Jentges, M., Sternberg, P., Richenhagen, J.: Software Architecture and Development Tool Chain for the Drive Train, ATZ elektronik, 01/2014
- (Paar 2015) Paar, C.: The future lies in a better encryption, Interview with C. Paar, ATZ elektronik, Vol. 3, Pages 22–24, Springer Vieweg Publ., 2015
- (Pickhard et al. 2015) Pickhard, F., Emele, M., Burton, S., Wollinger, T.: New thinking for safely networked vehicles (in German). ATZ elektronik, 7/2015
- (Pickhard 2016) Pickhard, F.: Measuring everything – Big Data in Automotive Engineering (in German). ATZ elektronik, 02/2016, Volume 11, Issue 1, pp 66–66
- (Postinett 2017) AWS server down – employee shut down internet with a typo. Handelsblatt online. 2nd March 2017. Available from: <https://www.handelsblatt.com/unternehmen/handel-konsumgueter/aws-serverausfall-amazon-mitarbeiter-legte-mit-tippfehler-teile-des-internets-lahm/19468246.html?ticket=ST-36653-ejHoJkYxHfBvZhgrUsfa-ap2>
- (Pratap 2006) Pratap, R.: Getting Started With Matlab. 7- A Quick Introduction for Scientists and Engineers. Oxford University Press, 2006
- (Proakis and Manolakis 2007) Proakis, J. G., Manolakis, D. G.: Digital Signal Processing, Prentice-Hall, Inc., 2007
- (Reif 2014) Reif, K. (Ed.): Driving stabilization systems and driver assistance systems. Springer-Vieweg Publ., 2016
- (Rembor et al. 2009) Rembor F., Kopp, T., Herzog, S., Guguenen, S.: Flexray – a Beginners' Guideline, ATZ elektronik, Vol 4, 03/2009, Pages 16–21, Springer Vieweg Publ., 2009
- (Reuss et al. 2015) Reuss, H.-C., Meyer, G., Meurer, M.: Roadmap 2030 Synergies of Electromobility and Automated Driving, ATZ elektronik, 2015
- (Reuter 2015) Reuter, A.: Data security is a must for functional security (in German), ATZ elektronik, 02/2015
- (Rich and Knight 1991) Rich, E., Knight, K.: Artificial Intelligence. Mc GrawHill Publ. 1991
- (Ridder et al. 1995) Ridder, C., Munkelt, O., Kirchner, H.: Adaptive Background Estimation and Foreground Detection using Kalman-Filtering. In: Process of Int. Conf. on recent Advances in Mechatronics. ICRAM'95, UNESCO Chair on Mechatronics, Pages 193–199, 1995
- (Russell and Norvig 2016) Russel, S., Norvig, P.: Artificial Intelligence: A Modern Approach. Pearson Education, 3rd edition, 2016
- (Schaal 2012) Schaal H.-W.: IP and Ethernet in Motor Vehicles. Vector Informatik GmbH, Available from: https://assets.vector.com/cms/content/know-how/_technical-articles/Ethernet_IP_ElektronikAutomotive_201204_PressArticle_EN.pdf, Pages 1–6, 04/2012
- (Schaal and Schwedt 2013) Schaal, H-W, Schwedt, M.: New Perspectives on Remaining Bus Simulation for Networks with SOME/IP. https://assets.vector.com/cms/content/know-how/_technical-articles/IP_SomeIP_AEL_201308_PressArticle_DE.pdf
- (Schaal 2017) Schaal, S.: Auto Trends at the CES – Only four Car Manufacturers are capable of developing everything on their own. Wirtschaftswoche online. January 4th 2017. Available from: <https://www.wiwo.de/unternehmen/auto/auto-trends-auf-der-ces-nur-vier-autobauer-koennen-alles-selbst-entwickeln/19203074.html>
- (Schäfer 2010) Schäfer, W.: Software Development - Introduction for the Most Demanding (in German). Addison-Wesley Publ., 2010
- (Schill and Springer 2012) Schill, A., Springer, T.: Distributed Systems – Fundamentals and core technologies. Springer Publ., 2012

- (Seeck 2015) Seeck, A. Don't expect too much (in German), Discussion at the 1st International ATZ Conference in Frankfurt - From Driver Assistance systems to Autonomous Driving, ATZ elektronik 3/2015
- (Serio and Wollschläger 2015) Serio, G., Wollschläger, D.: Networked Automotive Defense Strategies in the Fight against Cyberattacks (in German). ATZ elektronik, 06/2015
- (Siciliano et al. 2010) Siciliano, B., Sciacicco, L., Villan, L., Oriolo, G.: Robotics – Modelling, Planning and Control, Springer Publ., 2010
- (Siebenpfeiffer 2014) Siebenpfeiffer, W. (Ed.): Networked Automobile – Safety, Car IT, Concepts (in German). Springer Publ., 2014
- (Silberschatz et al. 2010) Silberschatz, A., Galvin, P., Gagne, G.: Applied Operating System Concepts, Wiley Publ., 2010
- (Singer and Friedman 2014) Singer P.W., Friedman, A.: Cybersecurity and Cyberwar: What Everyone Needs to Know, Oxford University Press, 2014
- (Soja 2015) Soja, R.: Security and the Connected Car: Secure Networks for V2X, NXP Community Online. Available from: <https://community.nxp.com/docs/DOC-105879>, 2015
- (Solon 2015) Solon, O.: From Car-Jacking to Car-Hacking: How Vehicles Became Targets For Cybercriminals. Bloomberg online. August 4th 2015. Available from: <https://www.bloomberg.com/news/articles/2015-08-04/hackers-force-carmakers-to-boost-security-for-driverless-era>
- (Sorge 2017) Sorge, N.-V.: Top Engineer from Apple should put Tesla's AutoPilot on track. Manager Magazin online. 11th January 2017. Available from: <http://www.manager-magazin.de/unternehmen/autoindustrie/tesla-apple-topingenieur-soll-autopilot-retten-a-1129485.html>
- (Steinmüller 2008) Steinmüller, J: Image analysis – From image processing to spatial interpretation of images, Springer Publ., 2008
- (Streichert and Traub 2012) Streichert, T., Traub, M.: Electric/Electronics Architectures in Automobiles (in German). Springer, Publ., 2012
- (Sushravya 2016) Sushravya, G.M.: Cybersecurity risks in Advanced Driver Assistance Systems, Class paper, Car IT and Cybersecurity class, IIIT-B, 2016
- (Thiele et al. 2013) Thiele, D., Ernst, R., Diemer, J., Richter, K.: Cooperating On Real-Time Capable Ethernet Architecture. In: Vehicles, ATZ elektronik 05/2013, Pages 40–44, Springer Publ., 2013
- (Tanenbaum and Bos 2015) Tanenbaum, A. S., Bos, H.: Modern Operating Systems. 4th edition, Pearson Publ., 2015
- (Tanenbaum and Van Steen 2017) Tanenbaum, A. S., Van Steen, M.: Distributed Systems Principles and Paradigms. 3rd edition, Pearson Publ., 2017
- (Vahid and Givargis 2001) Vahid, F., Givargis, T.: Embedded System Design, A Unified Hardware/ Software Introduction, Wiley and Sons Publ., 2003
- (Vembo 2016) Vembo, D.: Connected Cars – Architecture, Challenges and Way Forward. Whitepaper Sasken Communication Technologies Pvt. Ltd. 2016. Available from: https://www.sasken.com/sites/default/files/files/white_paper/Sasken-Whitepaper-Connected%20Cars%20Challenges.pdf
- (Vivekanandan et al. 2013) Vivekanandan, B., Bavishi, H., Paranjpe, K.: Preventing malfunctions. In: E/E systems, ATZ extra, Pages 72–74, 10/2013
- (Wagner 2015) Wagner M.A.: An adaptive Software and System Architecture for Driver Assistance Systems applied to truck and trailer combinations. Ph.D. thesis, University of Koblenz-Landau, Available from: https://www.researchgate.net/profile/Marco_Wagner2/publication/279528442_An_adaptive_software_and_system_architecture_for_driver_assistance_systems_applied_to_truck_and_trailer_combinations/links/55a4f8eb08aef604aa04123f/An-adaptive-software-and-system-architecture-for-driver-assistance-systems-applied-to-truck-and-trailer-combinations.pdf, 2015
- (Weber 2013) Weber, M.: AUTOSAR learns Ethernet. Vector Informatik GmbH. Available from: https://assets.vector.com/cms/content/know-how/_technical-articles/IP_AUTOSAR_Hanser_Automotive_201311_PressArticle_EN.pdf

- (Weber 2015) Weber, M.: New Communication Paradigms in Automotive Networking. Vector Informatik GmbH. Available from: https://assets.vector.com/cms/content/know-how/_technical-articles/Ethernet_CANFD_AutomobilElektronik_201508_PressArticle_long_EN.pdf
- (Weiβ et al. 2016) Weiβ, G., Schleiß, P., Drabek, C.: Fail-operational E/E Architecture for Highly-automated Driving Functions. ATZ elektronik, Vol 11, 03/2016, Pages 16–21, Springer Vieweg Publ., 2016
- (Weimerskirch 2016) Weimerskirch, A.: Cybersecurity for Networked and Automated Vehicles (in German). ATZ elektronik, 03/2016
- (Winner et al. 2009) Winner, H., Hakuli, S., Lotz, F., Singer, C. (Eds.): Handbook Driver Assistance Systems (in German). Springer Vieweg Publ., 2015
- (Wolfsthal and Serio 2015) Wolfsthal, Y., Serio, G.: Made in IBM Labs: Solution for Detecting Cyber Intrusion to Connected Vehicles, Part I. Available from: <https://securityintelligence.com/made-in-ibm-labs-solution-for-detecting-cyber-intrusions-to-connected-vehicles-part-i/>
- (Wolf et al. 2015) Wolf, J., Metzker, E., Happel, A.: Ethernet-Security – example SOME/IP. Vector Informatik GmbH (in German). Available from: https://assets.vector.com/cms/content/know-how/automotive-cyber-security/Ethernet-Security_SOMEIP_Lecture_VDI_2015.pdf, 2015
- (Zetter 2015) Zetter, K.: Researchers Hacked A Model S, But Tesla's Already Released A Patch. Wired online. August 6th 2015. Available from: <https://www.wired.com/2015/08/researchers-hacked-model-s-teslas-already/>

Links

2014

- (URL1 2014) Me, my car, my life, KPMG Automotive, 2014, <http://www.kpmg.com/Ca/en/IssuesAndInsights/ArticlesPmy-life-my-car.pdf>

2015

- (URL1 2015) <https://www.mcafee.com/de/resources/white-papers/wpautomotive-security.pdf>
- (URL2 2015) http://www.wiwo.de/unternehmen/auto/emobility/digitalisierung-der-aucht-man-das-lenkrad-nicht-mehr/v_detail_tab_print/11602152.html, 07.04.2015
- (URL3 2015) <https://www.bosch-presse.de/pressportal/de/en/bosch-and-daimler-automate-parking-mercedes-with-built-in-vault-42989.html>
- (URL4 2015) <https://www.symantec.com/content/dam/symantec/docs/white-papers/building-comprehensive-security-into-cars-en.pdf>
- (URL5 2015) <https://www.congress.gov/bill/114th-congress/senate-bill/1806/all-info>
- (URL6 2015) <https://www.digitaltrends.com/cars/bmw-automated-parking-technology-ces-2015/>
- (URL7 2015) <https://www.theiet.org/sectors/transport/documents/automotive-cs.cfm>
- (URL8 2015) https://delivering-tomorrow.de/wp-content/uploads/2015/08/dhl_self_driving_vehicles.pdf
- (URL9 2015) https://vector.com/portal/medien/solutions_for/Security/Ethernet-Security_SOMEIP_Lecture_VDI_2015.pdf
- (URL10 2015) <https://roscon.ros.org/2015/presentations/ROSCon-Automated-Driving.pdf>
- (URL11 2015) https://www.iaa.de/fileadmin/user_upload/2015/english/downloads/press/Automation_from_Driver_Assistance_Systems_to_Automated_Driving.pdf
- (URL12 2015) <http://www.team-bhp.com/forum/car-entertainment/159729-bmw-idrive-connected-drive-bmw-apps-review-faq-thread.html>

- (URL13 2015) http://www.euro.who.int/_data/assets/pdf_file/0006/293082/European-facts-Global-Status-Report-road-safety-en.pdf?ua=1
- (URL14 2015) <https://www.elektrobit.com/newsroom/webinar-automotive-ethernet-new-generation-ecu-communication/>

2016

- (URL1 2016) https://en.wikipedia.org/wiki/Advanced_driver_assistance_systems
- (URL2 2016) https://en.wikipedia.org/wiki/Safety_integrity_level
- (URL3 2016) <https://www.dhs.gov/science-and-technology/cyber-security-division>
- (URL4 2016) https://en.wikipedia.org/wiki/Functional_safety
- (URL5 2016) <http://www.exida.com/Resources/Term/Automotive-Safety-Integrity-Level-ASIL>
- (URL6 2016) https://en.wikipedia.org/wiki/Failure_mode_and_effects_anal
- (URL7 2016) <http://ec.europa.eu/programmes/horizon2020/>
- (URL8 2016) <https://www.abiresearch.com/market-research/product/1022093-connected-vehicle-cloud-platforms/>
- (URL9 2016) <http://www.acea.be/publications/article/strategy-paper-on-connectivity>
- (URL10 2016) <https://www.popsci.com/googles-cars-will-be-treated-like-human-drivers>
- (URL11 2016) <http://some-ip.com>
- (URL12 2016) <https://www.elektroniknet.de/fit-for-the-turning-point-in-the-automotive-industry-127725.html>
- (URL13 2016) <https://techcrunch.com/2016/08/25/2017-audi-a4-driver-assistance/>
- (URL14 2016) https://www.itskritis.de/_uploads/5/8/4/584ab66449001/idsposter.pdf
- (URL15 2016) <http://www.handelsblatt.com/unternehmen/industrie/elektroautopionier-tesla-ruestet-autos-zum-selbstfahren-auf/14713452.html>
- (URL16 2016) https://d23rjziej2pu9i.cloudfront.net/wp-content/uploads/2016/02/19120114/Secure_Ethernet_Communication_for_Autonomous_Driving.pdf

2017

- (URL1 2017) www.mobileye.com/technology/applications
- (URL2 2017) www.mathworks.com
- (URL3 2017) www.google.com
- (URL4 2017) <http://www.ficosa.com>
- (URL5 2017) https://www.bcgperspectives.com/content/articles/automotive-consumer-insight-revolution-drivers-seat-road-autonomous-vehicles/?chapter=4#chapter4_section4
- (URL6 2017) https://www.bcgperspectives.com/content/articles/automotive-consumer-insight-revolution-drivers-seat-road-autonomous-vehicles/?chapter=4#chapter4_section2
- (URL7 2017) <https://www.intel.com/content/dam/www/public/us/en/documents/corporate-information/policy-iot-automotive-transportation.pdf>
- (URL8 2017) <https://www.tomtom.com/>
- (URL9 2017) <https://www.bosch.com/>
- (URL10 2017) <https://www.bosch-iot-suite.com/>
- (URL11 2017) <https://archiv2017.iaa.de>
- (URL12 2017) <https://www.iaa.de>
- (URL13 2017) <http://www.emobil-sw.de/en/activities-en/current-projects/project-details/autopilot-automated-parking-and-charging-of-electric-vehicle-systems.html>
- (URL14 2017) [https://en.wikipedia.org/wiki/Here_\(company\)](https://en.wikipedia.org/wiki/Here_(company))
- (URL15 2017) <https://www.bmvi.de/DE/Themen/Digitales/Digitale-Testfelder/Digitale-Testfelder.html>

- (URL16 2017) <https://www.vda.de/en/topics/innovation-and-technology/network/networked-mobility.html>
- (URL17 2017) https://en.wikipedia.org/wiki/Vienna_Convention_on_Road_Traffic
- (URL18 2017) https://www.nhtsa.gov/sites/nhtsa.dot.gov/files/documents/12837-workshop_on_governance_of_automated_vehicles_03062017_final_version-tag.pdf
- (URL19 2017) <http://adtf.omg.org>
- (URL20 2017) <https://www.rti.com/products/dds>
- (URL21 2017) <https://www.vda.de/en/topics/innovation-and-technology/automated-driving/automated-driving.html>
- (URL22 2017) <https://www.conti-engineering.com/CMSPages/GetFile.aspx?guid=c3af2186-8330-4c66-bdbd-c082502ca609>
- (URL23 2017) <https://www.kpit.com/resources/downloads/kpit-autosar-handbook.pdf>
- (URL24 2017) <https://info.glass.com/mercedes-using-adas/>
- (URL25 2017) <https://www.mercedes-benz.com/en/mercedes-benz/innovation/mercedes-benz-intelligent-drive/>
- (URL26 2017) https://vector.com/vi_security_solutions_en.html
- (URL27 2017) [https://en.wikipedia.org/wiki/Dilation_\(morphology\)](https://en.wikipedia.org/wiki/Dilation_(morphology))
- (URL28 2017) [https://en.wikipedia.org/wiki/Erosion_\(morphology\)](https://en.wikipedia.org/wiki/Erosion_(morphology))
- (URL29 2017) <http://www.pedbikeinfo.org>
- (URL30 2017) <http://www.npr.org/2017/03/30/522085503/2016-saw-a-record-increase-in-pedestrian-deaths>
- (URL31 2017) <https://www.mathworks.com/help/images/functionlist.html>
- (URL32 2017) <https://de.mathworks.com/help/vision/object-tracking-1.html>



Summary, Final Remarks, Outlook, and Further Reading

12

This chapter summarizes the investigation conducted by the authors of this book and gives an outlook on future trends, technologies, innovations, and applications.

12.1 Summary

Chapter 1 set the stage for the investigation into connectivity and automotive cybersecurity, giving a bird's-eye perspective of the connected car, the essential technologies, and its most relevant applications and business models.

Chapter 2 took a close look at the automotive industry from a business perspective, presenting facts and figures for the passenger and the commercial vehicle market, for example, sales figures, market segments, registration numbers, passenger and commercial cars, supply chain, and others. A few OEMs, like Toyota and VW, command a market share of 10%, while some of the main suppliers even reach a 20% market share. The industry has undergone various waves of mergers and acquisitions. The main megatrends that are shaping the industry today are autonomous driving, connectivity, e-mobility, shared mobility, and the digital transformation of the business as a whole. With such a fast pace of change, the industry does not only rely on its own innovation capabilities but has started to work very closely with startups, for example, Daimler's Startup Autobahn.

Chapter 3 gave an overview of Automotive R&D and virtual product creation. A car is a very complex product where thousands of employees have to work together on the development of a new model, often distributed over different continents. Stage-gate controlled processes help to organize and streamline the activities. Most engineering tasks are computer-based. CAD systems, CAE tools, digital mock ups, and sophisticated simulation environments, help to create a virtual product of growing complexity and maturity. Whereas traditionally, automotive engineering focused on the geometry of metal and plastics parts, today, electronics,

mechatronics, and software have become increasingly important. Modern Product Lifecycle Management (PLM) systems are capable of integrating all these domains (electronics, mechanics, hardware, and software) seamlessly.

Chapter 4 provided an overview of automotive electronics. This area has become the dominant source for innovation and accounts for a rising part of the overall cost and value add of a modern car. The chapter looked at different domains and application areas, gave an overview of automotive bus systems and briefly touched upon design and architecture issues like the AUTOSAR software stack as well as hardware-in-the-loop (HIL) testing of electronic control units.

Chapters 5 and 6 dealt with the core topics of the book: connected cars and automotive cybersecurity. Connectivity is one of the main drivers for innovation and new business models. The connected car exchanges information with its environment. It is a complex cyber-physical system. The chapter gave an overview of different communication technologies and business models, discussed the technologies in the context of the Internet of Things (IoT), and gave an overview of different architectures, finally presenting the approaches of Mercedes and BMW as case studies, the *Mercedes Command Online* and *BMW's Connected Drive*.

Chapter 6 provided a detailed introduction to the topic of automotive cybersecurity. Cybersecurity encompasses technologies, processes, and management practices to protect vehicles from attacks, damage, or unauthorized access. The papers from Miller and Valasek in 2014 and 2015 (Miller and Valasek 2014, 2015), the article of Greenberg in the *Wired Magazine* in 2015 (Greenberg 2015), and the report in 2015 by Senator Markey and his team (Markey 2015) have drawn attention to this topic (URL1 2015). Cybersecurity issues become even more important with higher degree of connectivity and new mobility concepts like carsharing, ridesharing, and autonomous driving at the horizon. Security is based on authenticity, integrity, confidentiality, availability and non-repudiation. Therefore, the chapter presents a theoretical framework for cybersecurity, discusses its application to automotive systems, and gave insight into some important case studies. Topics included vulnerabilities, cyberattack surfaces, attack vectors, intrusion detection, intrusion prevention, and solutions for automotive cybersecurity.

Chapter 7 discussed mobile apps for automotive applications. Starting with an overview of information management in automotive companies, the chapter analyzed the challenges of the digital transformation which has led to the term two-speed IT and has popularized agile software methodologies. The leading platforms for mobile apps are Android and iOS. Both platforms are briefly introduced and discussed, and a student project from IIIT-B's 2016 class on Car IT and cybersecurity gave a detailed case study insight into the analysis, design, and implementation of a ridesharing app for students.

Chapters 8, 9, and 10 presented important practical applications: carsharing, ride-hailing/ridesharing, and connected parking as well as automated valet parking. These applications rely on connectivity and expose cyberattack surfaces that have to be carefully dealt with.

Finally, Chap. 11 focused on advanced driver assistance systems (ADAS) and autonomous driving. The chapter gave examples of ADAS functionalities like lane

departure warning and lane keeping assistance, looked at the underlying image processing and image analysis technologies, and discussed legal aspects of higher automation and autonomous driving.

Autonomous cars need flexible E/E and software architectures. The authors showed how middleware can handle the huge traffic of sensor data and discussed the deployment of service-oriented architecture patterns with publish-subscribe functionality. They presented a framework for functional safety and cybersecurity. The field of cybersecurity for autonomous driving is an active and very innovative and essential field of research. The book outlined some direction of this research like security of SOME/IP and encouraged the reader to review the growing research literature in this field.

12.2 Final Remarks: Wind of Change

The automotive industry is facing an unprecedented time of change, both in terms of the speed of this transformation as well as in terms of the breadth and depth (Broy 2015; URL2 2016). While acceleration, top speed, horsepower, and design were the most important criteria for buying a car in the past, electronics and software innovations are the defining criteria of tomorrow (Dunker and Bretting 2016).

Figure 12.1 shows the engine of Volkswagen's Bugatti Veyron. This engine is, no doubt, a marvel of automotive engineering; however, connectivity, artificial intelligence, advances in semiconductors, battery capacity, ease of charging, and the best integration of the car into the digital ecosystem will be the competitive differentiators in the market of tomorrow.

The four drivers of this change:

- Autonomy
- Connectivity
- Electro-mobility
- Shared mobility

are seen as fundamental game changers.

Daimler's response in this regard is the creation of a new organization starting with the letters C, A, S, and E, in short CASE, which stand for the strategic future areas of networking (connected), autonomous driving (autonomous), flexible use (shared), and electric drives (electric).

12.2.1 Frugal Engineering

When Tata launched the Nano, many believed it is impossible to create a Rs. 1 lakh car (the equivalent of USD 2000). However, India has a lot of inexpensive entry-level cars starting at 3000 USD, and Tata was able to introduce the Nano for a very competitive price tag, also thanks to a local, efficient, and inexpensive supply

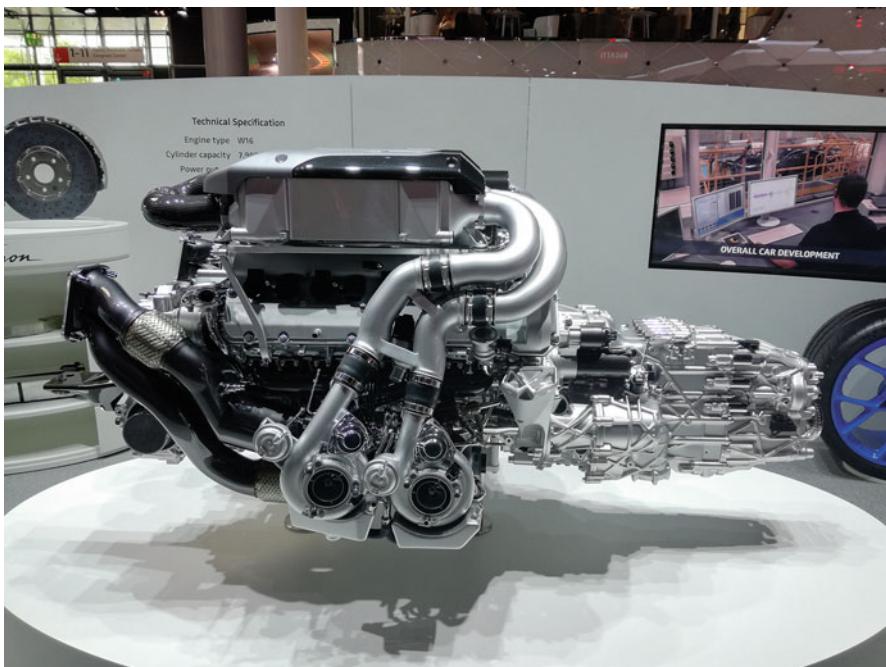


Fig. 12.1 A 16-cylinder, 1500 hp engine of VW's Bugatti Veyron shown at the IAA 2017

chain. The key is frugal engineering, an approach which is simple but effective, sometimes improvised but innovative, always keeping a keen eye on cost and a sharp focus on the need of the mass market. Lots of interesting ideas come from emerging markets, and many of these are also useful for mature markets.

12.2.2 Rise of Asian Markets

The rise of China to the top of the automotive market to become the biggest in the world has shown the potential of emerging markets. The Indian automotive market is growing rapidly too, and it will rise to become the second largest market in the world. It is already the second largest market for busses and one of the major markets for commercial vehicles. Asian mega cities are growing, and the rapid urbanization drives more and more people into the cities with rising demand for individual and public transport. This demand however, cannot be satisfied with technologies based on the internal combustion engine. It needs a mix of internal combustion engine powertrain (ICE) and electric drive technologies, and in the long run pure battery electric cars with zero emission in the crowded metropolitan areas. The diesel engine scandal is accelerating this development. Already, China and India have passed ambitious goals for electric cars as have many European countries and the state of California.

12.2.3 E-Mobility

It is clear that the motorization rate can only grow in a sustainable way with electric cars. This has sparked a lot of research in the underlying technologies like batteries, battery management, power electronics, charging infrastructure, and so forth. Nearly all automotive OEMs have announced a plethora of new battery electric vehicles (BEV). Daimler's compact city car, the Smart, will soon only be available with an electric powertrain as shown in Fig. 12.3. New competitors are coming up like *StreetScooter* (URL6 2017). Much to the dismay of the German automotive OEMs, the German postal service, Deutsche Post DHL Group, does not order their electric vans at Volkswagen, Daimler, or BMW but produces these vehicles in its own subsidiary called StreetScooter, a former e-mobility startup and spin-off from the Technical University of Aachen. Big high-tech companies like Apple are also moving into the lucrative space of e-mobility building on their knowledge of batteries, electronics, software, and the Apple customer ecosystem. Chinese startups spring up like mushrooms, often backed by rich Chinese investors or Chinese IT giants like Alibaba and Baidu.

12.2.4 Fuel Cells

Besides the electric powertrain, hydrogen is another option. Hydrogen can be combusted in fuel cells together with oxygen to form water and generate electricity for an electric powertrain. The technology is not new, but now it is so well developed that it can play its advantages against a pure electric powertrain. The most important advantage is that fuel cell vehicles have their fuel on board, the batteries are considerably smaller, and the range is even higher. The ix35 fuel cell from Hyundai has a range up to 590 km with one tank filling. The refueling process itself is completed in a few minutes, in contrast to pure electric vehicles. Besides, Toyota, Hyundai also offers hydrogen vehicles in series in Germany. Daimler presented a first series model at the IAA 2017 as well.

12.2.5 Connected Cars

At the heart of many innovations lies connectivity. Connected cars communicate with each other, the infrastructure, the cloud, and so forth. They are location-aware, can process much more information than humans, and generate valuable data that could be used for various new services, like emergency services, remote diagnostics, location-aware retail solutions, and usage-based insurances. This is similar to the ecosystem around the smartphone; however, the potential amount of data and the breadth and depth of the data are much higher. Startups like *otonomo* have recognized this and are currently building up a market for car data (URL1 2017). However, there are a couple of key issues and questions to deal with: who owns the data, how does one make sure that privacy is protected, how does everybody get a

fair share, and how does one prevent the OEMs to monopolize this data? These issues have to be handled carefully (Hammerschmidt 2017; URL1 2016).

BMW has already made a bold step with *CarData*, a connected car data service, where a customer can tick off what data they want to share with whom through a portal that is run by the company (URL7 2017). Also, customers can download interesting data on a regular basis.

12.2.6 Shared Mobility

New means of mobility are on the rise. Carsharing, ridesharing, and ridehailing use the smartphone platform for service offering and billing. In India and China, this has led to a boom of people moving into this business and disrupted the market for public transport completely. The efficient inter- and intra-modal transport and the interfaces between different transportation platforms, for example, car to train to bus and others, are still an area of ongoing innovation.

Car ownership is being augmented or replaced by access to mobility and new business models for transportation with flexible utilization (Bridges 2015; URL4 2017). Platforms and digital ecosystem become more important with a seamless integration of billing, discount schemes, and many opportunities of cross-selling. Digital platforms tend to quickly become very large, where the winner takes most as it can be seen from Google, Airbnb, [Booking.com](#), Uber, and others. Also, the way cars are sold will change fundamentally. Tesla has shown this in an impressive way: a few show rooms in prime locations, booking, delivery, and services managed through the Internet.

12.2.7 Autonomous Driving

Adaptive cruise control, emergency breaking, blind spot detection, lane departure warning, and lane keeping assist, as well as remote and automatic parking are just a few of many ADAS functions that make a car safer and easier to drive. ADAS systems have various nonfunctional requirements regarding flexibility, modularization, responsiveness, reliability, testability, security, and others.

These higher levels of automation and autonomous driving have a huge impact on the E/E and software architecture and need flexible bus systems with high bandwidth to carry the huge amount of sensor data, powerful processing platforms for sophisticated image processing, and innovations in artificial intelligence (AI) and machine learning.

The legal framework for autonomous driving has still to evolve, but this might happen faster than in former times because no one wants to stay behind in such an innovative domain. The key is the modification of the *Vienna Convention for Road Traffic* which requires a human to be in charge of vehicle control at any time. Also, ethical issues have to be dealt with and liability and insurance models have to adapt.

Already, suppliers like ZF/TRW talk about their vision zero: no accidents anymore. High automation and autonomous driving will play an important role here. Consider, for example, the number of accidents caused by trucks crashing into the end of a traffic jam—emergency breaks and conditional automation could prevent this.

In Fig. 12.2, BMW's concept car for autonomous driving, which was presented at the IAA 2017 auto show in Frankfurt (URL2 2017), is shown. Daimler's *Smart Vision EQ For Two*, shown in Fig. 12.3, will be an autonomous two-seater which could already be added to the company's growing Car2Go fleet by 2020.

The huge investments into autonomous driving can only be handled by a few OEMs, most of them need help from partners and strike new alliances (Schaal 2017; Freitag 2016).

12.2.8 Automotive Cybersecurity

The digital transformation has a dark side: cybercrime is on the rise. Car manufacturers do not fear so much that an individual car is hacked but the potential of being blackmailed on a larger scale.

Autonomous driving is based on connectivity, complex software systems, over-the-air updates, and car-to-cloud communication. All this creates broad and complex attack surfaces. Cybersecurity for ADAS and autonomous driving requires a holistic approach. The first steps in this direction can be seen but much more work is necessary (Weimerskirch 2016).



Fig. 12.2 BMW's vision of an autonomous car presented at the IAA 2017



Fig. 12.3 Daimler's concept car Smart Vision EQ ForTwo presented at the IAA 2017

Functional safety and cybersecurity are closely connected. Reliable autonomous cars will only be possible by adopting a “design by security approach” with well-designed software stacks.

12.3 Outlook and Further Reading

12.3.1 Outlook

The future will witness an accelerated convergence of technologies. The automotive industry will change substantially, established players will be bought over, others will merge, and some companies will vanish forever. On the other hand, a lot of new ventures will come up, especially in the e-mobility space, and interesting new collaboration models will emerge. Dudenhöffer (2016) discusses some of these scenarios and speculates on who will still be around in the future.

Will we drive a car in the future? There will be people who love to drive, and they will enjoy the unique experience, but many will just rely on self-driving and enjoy the freedom of not having to drive on their own. The autonomous car will be safer and more efficient. The transportation of goods will be dominated by self-driving systems, and also a majority of public transport will be handled by autonomous people movers.

When one of the authors entered the automotive industry right after his studies in the 1990s, it seemed to be an El Dorado for technologies as the car is arguably the most complex mass-produced consumer product.

Again, it is an exciting time to be part of this industry witnessing a change process that has been unprecedented in history. Where will it lead us?

12.3.1.1 GAFA

Never before, the automotive industry had to deal with so many cutting edge innovations at once: the electric vehicle paradigm, the connected and autonomous vehicles paradigm, the new mobility paradigm, and the Industry 4.0 paradigms (Möller 2016). Forecasts speculate that in the very near future vehicles as known today, for example, personally driven and privately acquired and owned vehicles, powered by an ICE, and manufactured by automakers and their suppliers, cease to exist. The forecasts announce that we face a replacement by electric, autonomous, connected mobility services produced in highly automated and flexible factories which are based on the Industry 4.0 paradigm. Furthermore, several reports suggest that other actors, beside automakers and suppliers, represented by the four American Internet companies Google, Apple, Facebook, and Amazon (GAFA) will take control of this new digital value chain. The four giant Internet companies have a combined market value of more than 2,5 trillion US \$ (May 2018). For example, the Google self-driving car project, now named Waymo, stands for a new way forward in mobility with the mission to make driving safe and easy for people and things to move around. Beside this Tesla, a pioneer in electric vehicles and founder Elon Musk building up the production of battery packs the so-called Gigafactory. Tesla's Gigafactory 1, an operational lithium-ion battery at the Tahoe Reno Industrial Center (TRIC) in Storey County, near Clark, in Nevada, USA, began mass production of cells in January 2017. Tesla's Gigafactory 2 refers to the SolarCity Gigafactory at Buffalo, New York, USA, in February 2017. The locations of Gigafactory 3, 4, and 5, introduced as European Gigafactories, will be announced in 2017 too (URL8 2017).

In 2015, the automotive industry has produced worldwide more than 85 million vehicles, an all time record (see Chap. 2). Never before in history were so many ICE-powered and privately owned and driven vehicles produced by the traditional automakers and their suppliers. But now new technologies and services are coming up with disruptive innovations that are supposed to be small, before becoming dominant. But how to know in advance that we are dealing with true disruptive innovations or just short term trends? Companies like Tesla, Uber, and others (see Chap. 9) do appear as successful disruptive players, pushing forward fully electric cars, new mobility services, and autonomous cars, but the question is still what will be the impact, how long can they survive if losses grow much faster than revenues, and will the new business models sustain these disruptive transformations? It is not only the production and sale of electric vehicles but also the required charging infrastructure, the provision of batteries and electricity in the required amount, and at an attractive price to support a fast growing and large market volume. However, only the companies with the best innovations can take over the economic leadership. Are these, beside Tesla and others, the GAFA companies which have the power and

success for continually developing themselves? Meanwhile they have emerged in the past decades from big technology companies to global giants, as can be seen from the Nasdaq Index. Nowadays these companies stand for stability, which continues to evolve and thus turn creative destruction into a mission statement.

12.3.2 Further Reading

Dudenhöffer (2016) and the studies conducted by KPMG (URL1 2014) and VDA (URL4 2017, URL5 2017) discuss many aspects of the digital transformation and show how the industry is shaken up by the new trends of electro-mobility, shared mobility, and autonomous driving. Alexander et al. (2017) analyze the impact of these disruptive effects on the supply chain of automotive electronics.

Further material on automotive electronics can be found in Streichert and Traub (2012), Reif (2014), Borgeest (2014), Krüger (2014), and Navet and Simonot-Lion (2009). An up-to-date overview of automotive software engineering is given in Schäuffele and Zurawka (2016). A detailed overview of modeling, simulation, and control of vehicle driveline is given in Kiencke and Nielsen (2005).

For an in-depth discussion of cybersecurity, refer to Eckert (2014). An overview of car hacking is given in Smith (2016) and Polchow (2016). The DEFCON conference in Las Vegas now regularly organizes a special session on car hacking which is called the “car hacking village” (URL3 2017).

A good overview of the challenges and trends in cybercrime is given in Singer and Friedman (2014), Köncke and Buehler (2015), Loukas (2015), Mitnick and Simon (2005), as well as Goodman (2015).

For more information on electro-mobility, the authors refer to Kampker et al. (2013), Kampker (2014), Hinderer et al. (2016), and Schöttle (2017). Reuss et al. (2015) discuss the synergies between electro-mobility and autonomous driving.

Good summaries of the current research activities in ADAS and autonomous driving are Winner et al. (2009), Siebenpfeiffer (2014), Maurer et al. (2015), Bernhart (2017), and Beck (2016). Becker (2016) discusses the issues that arise if technology is not mature yet.

Ross (2014) gives a detailed overview of functional safety in automotive engineering.

A good introduction to artificial intelligence and machine learning can be found in Russell and Norvig (2016), Kaplan (2016), and Kurzweil (2012). Haykin (2009) gives an in-depth overview of neural networks as an interesting paradigm of nonlinear pattern recognition and machine learning with a multitude of applications in cybersecurity, intelligent mobility, and autonomous driving. AI currently attracts a lot of funding, which is discussed in Markoff (2016) and Menn (2016).

Valuable information on big data and the connected car can be found in the reports by PwC (Viereckl et al. 2016), Cisco (URL2 2015), McKinsey (URL2 2014), and Lamparth et al. (2014).

A compact introduction to Car IT is given in Johanning and Mildner (2015).

A recent overview of the digital transformation in the automotive industry can be found in Koehler and Wollschlaeger (2014).

References and Further Readings

- (Alexander et al. 2017) Alexander, M., Bernhart, W., Zinn, J.: Tier-1 under Pressure, Direction and Role Change in the supply chain (in German). ATZ elektronik, pp. 31–35. 03/2017
- (Becker 2016) Becker, J. Autopilot of Tesla – in a Tesla the risk is a standard feature (in German). Süddeutsche Online. November 17th 2016. Available from: <http://www.sueddeutsche.de/auto/autopilot-von-tesla-bei-tesla-ist-das-risiko-serienmaessig-1.3252192>
- (Beck 2016) Beck, T., Do we need Autonomous Driving? (in German) elektronik.net , 01/2016, pp. 48–49. 01/2016
- (Bernhart 2017) Bernhart, W.: Autonomous Driving, Markets, Drivers and Business Models (in German), ATZ elektronik, pp. 36–41. 02/2016
- (Borgeest 2014) Borgeest, K.: Electronics in vehicle technology – Hardware, Software, Systems, and project management (in German). Springer Vieweg Publ., 3rd edition, Wiesbaden, 2014
- (Bridges 2015) Bridges, R.: Driverless Car Revolution Buy Mobility – Not Metal, Self-published, 2015
- (Broy 2015) Broy, M.: The Danger is in the Speed with which things are changing (in German), ATZ elektronik, Springer Vieweg Publ., 8/2015
- (Dietz et al 2016) Dietz, W., Reindl, S., Bracht, H. (Eds): Basic Principles of the Automotive Business (in German). Springer Automotive Media, 2016
- (Dudenhöffer 2016) Dudenhöffer, F.: Who will be put in the fast lane (in German). Campus Publ., 2016
- (Dunker and Bretting 2016) Dunker, H., Bretting, R.: Times are Changing (in German). Automotive IT, pp. 46–49. 01/02 2016
- (Eckert 2014) Eckert C., IT Security – Concepts, Methods and Protocols, De Gruyter Oldenbourg (in German), 9th edition, Munich, 2014
- (Freitag 2016) Freitag, M.: Robotic cars – German manufacturers in pole position (in German). Manager Magazin online. July 26th 2016. Available from: <http://www.manager-magazin.de/unternehmen/autoindustrie/roboterautos-deutsche-autobauer-fuehrena-1104783.html>
- (Goodman 2015) Goodman, M.: Future Crimes. Doubleday Publ., 2015
- (Greenberg 2015) Greenberg, A.: Hackers Remotely Kill a Jeep on the Highway with me in it. Wired online. July 21st 2015: Available from: <https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/>
- (Haykin 2009) Haykin, S.: Neural Network and Learning Machines. 3rd edition. Pearson Education, 2009
- (Hammerschmidt 2017) Hammerschmidt, C.: Shut Off (in German). carIT, special IAA edition, pp. 32–34. 01.2017
- (Hinderer et al. 2016) Hinderer, H., Pflugfelder, T., Kehle, F.: Electromobility (in German). Springer Automotive Media, 2016
- (Johanning and Mildner 2015) Johanning, V., Mildner, R.: Car IT compact – The Car of the Future – Driving Connected and Autonomously (in German). Springer Vieweg Publ., Wiesbaden, 2015
- (Kampker et al. 2013) Kampker, A., Vallee, D., Schnettler, S.: Electromobility – Foundations of a technology for the future (in German). Springer Vieweg Publ., Wiesbaden, 2013
- (Kampker 2014) Kampker, A.: Production of electromobiles, Springer Vieweg, Publ., 2014
- (Kaplan 2016) Kaplan, J.: Artificial Intelligence, Oxford University Press, 2016
- (Kiencke and Nielsen 2005) Kiencke, U., Nielsen, L.: Automotive Control Systems: For Engine, Driveline, and Vehicle. Springer Publ., 2005
- (Koehler and Wollschlaeger 2014) Koehler, T. R., Wollschlaeger, D.: The digital transformation of the automobile – five megatrends which are changing the industry, Media-Manufaktur GmbH, 2014
- (Köncke and Buehler 2015) Köncke, F. C., Buehler, B. O.: Cyber Attacks – Underestimated Risk for the German Industry (in German). Wirtschaftswoche online. November 9th 2015. Available from: <https://www.wiwo.de/technologie/digitale-welt/cyber-angriffe-unterschaetzes-risiko-fuer-die-deutsche-industrie/12539606.html>

- (Krüger 2014) Krüger, M.: Foundations of automotive electronics: circuits (in German). 3rd edition, Hanser Publ., Munich, 2014
- (Kurzweil 2012) Kurzweil, R.: How to create a mind: The Secret of Human Thought revealed. Viking Penguin Publ., 2012
- (Lamparth et al 2014) Lamparth, O., Bähren, F.: From the Connected to the Autonomous Car. ATZ elektronik, Vol 9, 05/2014, pp. 36–39, Springer Vieweg, Publ., 2014
- (Loukas 2015) Loukas, G.: Cyber-Physical Attacks – A growing invisible Threat. Elsevier Publ., 2015
- (Maurer et al. 2015) Maurer, M., Gerdes, C. J., Lenz, B., Winner, H. (Eds.): Autonomous Driving, Technical, legal and social aspects (in German). Springer Vieweg Publ., 2015
- (Markey 2015) Markey, E.J.: Tracking and Hacking: Security and Privacy Gaps Put American Drivers at Risk. 2015. Available from: https://www.markey.senate.gov/imo/media/doc/2015-02-06_MarkeyReport-Tracking_Hacking_CarSecurity%202.pdf
- (Markoff 2016) Markoff, J.: Artificial Intelligence Swarms Silicon Valley on Wings and Wheels, The New York Times, July 2016, URL <http://nyti.ms/2a0Awys>
- (Menn 2016) Menn, A.: Nvidia founder Huang: Artificial Intelligence triggers next Industrial Revolution (in German). Wirtschaftswoche Online. 9.12.2016. <https://www.wiwo.de/technologie/digitale-welt/nvidia-gruender-huang-kuenstliche-intelligenz-loest-naechste-industrielle-revolution-aus/14951562.html>
- (Miller and Valasek 2014) Miller C., Valasek C.: A Survey of Remote Automotive Attack Surfaces. IOActive 2014. Available from: https://www.ioactive.com/pdfs/IOActive_Remote_Attack_Surfaces.pdf
- (Miller and Valasek 2015) Miller, C., Valasek, C.: Remote exploitation of an unaltered passenger vehicle. August 10th 2015. Available from: <http://illmatics.com/Remote%20Car%20Hacking.pdf>
- (Mitnick and Simon 2005) Mitnick, K. D., Simon, L.: The art of Intrusion, Wiley Publ., Hoboken, NJ, 2005
- (Möller 2016) Möller, D. P. F.: Guide to Computing Fundamentals in Cyber-Physical Systems – Concepts, Design Methods, and Applications. Springer Publ., 2016
- (Navet and Simonot-Lion 2009) Navet, N., Simonot-Lion, F.: Automotive Embedded Systems Handbook. CRC Press, 2009
- (Polchow 2016) Polchow, Y.: Hacker on the Fast Lane (in German). automotive IT, pp.18–21. 01/02/2016
- (Reif 2014) Reif, K: Automotive Electronics: An Introduction for Engineers (in German). Springer Vieweg Publ., 2014
- (Reuss et al. 2015) Reuss, H.-C., Meyer, G. and Meurer, M.: Roadmap 2030 Synergies between electromobility and autonomous driving (in German). ATZ elektronik, pp 54–57, 2015
- (Ross 2014) Ross, H.-L.: Functional Safety in Automobiles (in German), Hanser Publisher, Munich, 2014
- (Russell and Norvig 2016) Russel, S., Norvig, P.: Artificial Intelligence: A Modern Approach. Pearson Education, 3rd edition, 2016
- (Schaal 2017) Schaal, S.: Auto Trends at the CES – Only four Car Manufacturers are capable of developing everything on their own. Wirtschaftswoche online. January 4th 2017. Available from: <https://www.wiwo.de/unternehmen/auto/auto-trends-auf-der-ces-nur-vier-autobauer-koennen-alles-selbst-entwickeln/19203074.html>
- (Schäuffele and Zurawka 2016) Schäuffele, J., Zurawka, T.: Automotive Software Engineering – Basics, Processes, Efficient Deployment of Methods and Tools. Springer Vieweg Publ., 6th edition, Wiesbaden, 2016
- (Siebenpfeiffer 2014) Siebenpfeiffer, W. (Ed.): Connected Cars – Security, Car IT, Concepts (in German). Springer Publ., 2014
- (Singer and Friedman 2014) Singer, P., Friedman, A.: Cybersecurity and Cyberwar: What Everyone Needs to Know, Oxford University Press, Oxford, UK, 2014
- (Schöttle 2017) Schöttle, M.: New roles of Electromobility (in German). ATZ elektronik, pp. 8–15, 03/2017

- (Smith 2016) Smith, C.: The Car Hacker's Handbook - A Guide for Penetration Tester, no starch press, San Francisco, 2016
- (Streichert and Traub 2012) Streichert, T., Traub, M.: Electric/Electronics Architectures in Automobiles (in German). Springer, Publ., 2012
- (Viereckl et al. 2016) Viereckl, R., Ahlemann, D., Koster, A., Hirsh, E., Kuhnert, F., Mohs, J., Fischer, M., Gerling, W., Gnanasekaran, K., Kusber, J., Stephan, J., Crusius, D., Kerstan, H., Warneke, T., Schulte, M., Seyfferth, J., Baker, E. H.: Connected car report 2016: Opportunities, risk, and turmoil on the road to autonomous vehicles. September 28th 2016. Available from: <https://www.strategyand.pwc.com/report/connected-car-2016-study>
- (Weimerskirch 2016) Weimerskirch, A.: Cybersecurity for Networked and Automated Vehicles (in German). ATZ elektronik, 03/2016
- (Winner et al. 2009) Winner, H., Hakuli, S., Lotz, F., Singer, C. (Eds.): Handbook Driver Assistance Systems (in German), Springer Vieweg Publ., 2015

Links

2014

- (URL1 2014) <https://assets.kpmg.com/content/dam/kpmg/pdf/2014/11/me-my-life-my-car.pdf>
- (URL2 2014) <https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/setting-the-framework-for-car-connectivity-and-user-experience> (2014–2018)

2015

- (URL1 2015) <https://www.congress.gov/bill/114th-congress/senate-bill/1806/all-info>
- (URL2 2015) https://cci.car-it.com/download/CCI_2015_Web.pdf

2016

- (URL1 2016) <http://www.acea.be/publications/article/strategy-paper-on-connectivity>
- (URL2 2016) <https://www.elektroniknet.de/fit-for-the-turning-point-in-the-automotive-industry-127725.html>

2017

- (URL1 2017) <http://otonomo.io>
- (URL2 2017) <https://www.iaa.de/>
- (URL3 2017) <https://defcon.org>
- (URL4 2017) <https://www.vda.de/de/themen/innovation-und-technik/vernetzung/vernetzte-mobilitaet.html>
- (URL5 2017) <https://www.vda.de/en/topics/innovation-and-technology/automated-driving/automated-driving.html>
- (URL6 2017) <https://www.streetsooter.eu>
- (URL7 2017) <http://www.bmw.de/de/topics/faszination-bmw/connecteddrive/digital-services/bmw-cardata.html>
- (URL8 2017) https://en.wikipedia.org/wiki/Gigafactory_1

Glossary

A

ABS Anti-lock braking system is a vehicle safety system that allows the vehicle wheels to maintain tractive contact with the road surface according to driver inputs while braking, preventing the wheels from locking up and avoiding uncontrolled skidding

AC Alternating current is an electric current which periodically reverses direction, in contrast to direct current (DC) which flows only in one direction

ACC Adaptive cruise control is an optional control system for vehicles that automatically adjusts the vehicle speed to maintain a safe distance from vehicles ahead

ACEA European Automobile Manufacturers Association represents Europe's car, van, truck, and bus manufacturers

ACM Association for Computing Machinery is the world's largest educational and scientific computing society and delivers resources that advance computing as a science and a profession

ACPS Automotive cyber-physical systems are sophisticated systems which embed electronic components and control systems to improve performance and safety

ACS Airbag control system detects and evaluates a crash before triggering the appropriate restraint systems according to the type of collision and its severity

ACSS Automotive cloud service system based on SOA for the next-generation automotive software platform

AD Autonomous driving or self-driving refers to the capability of a vehicle to sense its environment and to navigate without human input

ADC Analog-to-digital converter translates analog electrical signals into digital signals for data processing purposes

ADAS Advanced driver assistance systems are systems to help the driver in the driving process by increasing car safety, road safety, and better driving

ADTF Automotive Data and Time-Triggered Framework is a framework, which supports automotive software development. It has the advantage of a stable measurement framework, which is used in ADAS and can adopt typical bus data, for example, CAN, FlexRAY, Ethernet, and others, as well as raw data from any sources

- AEB** Autonomous emergency braking is a system that acts independently of the driver and will intervene only in a critical situation to avoid or mitigate an accident by applying the brakes
- AEMP** Association of Equipment Management Professionals is the premier organization serving those who manage and maintain heavy, off-road fleets
- AES** Advanced Encryption Standard is a specification for the encryption of electronic data established by the US NIST
- AG** Stock corporation is a legal form for companies
- AGORA** Framework for the development of intelligent transportation system applications
- AHA** Adaptive high beam assist makes driving in the dark safer and helps to reduce the strain on the driver
- AI** Artificial intelligence is the simulation of human intelligence processed by machines, especially computer systems
- AIC** Availability, integrity, and confidentiality, also known as CIA triad, is a model designed to guide policies for information security within an organization
- AID** Anomaly intrusion detection is a method for detecting both network and computer system intrusions and misuse by monitoring system activity and classifying it as either normal or anomalous
- ANN** Artificial neural networks is a computational model based on the structure and functions of biological neural networks
- ANSI** American National Standards Institute is a private nonprofit organization that oversees the development of voluntary consensus standards for products, services, processes, systems, and personnel in the USA
- AP** Automatic parking is an autonomous vehicle-maneuvering system that moves a vehicle from a traffic lane into a parking spot to perform parallel, perpendicular, or angle parking
- API** Application programming interface is a set of routine, protocols, and tools for building software applications
- APRANET** Advanced Research Projects Agency Network was an early packet-switching network and the first network to implement the protocol suite TCP/IP
- AR** Augmented reality is a method that superimposes a computer-generated image on a user's view of the real world, thus providing a composite view
- ARP** Address Resolution Protocol maps an Internet Protocol address to a physical machine address that is recognized in the local network
- ASD** Agile software development is a set of methods and practices where solutions evolve through collaboration between self-organizing, cross-functional teams
- ASAM** Association for Standardization of Automation and Manufacturing is an incorporated association under German law which members are primarily international automakers, suppliers, and engineering service providers from the automotive industry
- ASIL** Automotive Safety Integrity Level is a risk classification scheme defined by the ISO 26262—Functional Safety for Road Vehicles Standard

ATM Air traffic management is the process by which aircrafts are safely separated in the sky as they fly and at the airports where they land and take off again

AUP Agile Unified Process is a simplified version of the RUP

AUTOSAR AUTomotive Open System ARchitecture is a worldwide development partnership of automotive-interested partners which pursues the objective of creating and establishing an open and standardized software architecture for automotive ECUs excluding infotainment

AV Autonomous vehicle

AVOIDIT Attack Vector, Operational Impact, Defense, Information Impact, and Target efficiently classifies blended attacks by using five major classifiers to characterize the nature of an attack: classification by attack vector, classification by operational impact, classification by defense, classification by informational impact, and classification by attack target

AVP Automatic vehicle parking detects and evaluates a space for parking vehicles in a car park area

B

BAC Blood alcohol concentration refers to the alcohol intoxication of a person usually expressed as a percentage of ethanol in the blood in units of mass of alcohol per volume of blood

BCM Body control module is a generic term for an ECU responsible for monitoring and controlling various electronic accessories in a vehicle's body

BCU Brake control unit is responsible for system control, wheel slide protection, and diagnostics

BDA Big Data Analytics represents a new era in data exploration and utilization

BE Best effort denotes a quality of service in speech and data networks

BEV Battery electric vehicles use electricity stored in a battery pack to power an electric motor and turn the wheels

BFA Brute force attack is a form of attack in which hackers try to crack passwords or decrypt data using raw force (brute force), which means more or less indiscriminate testing

BMVI Ministry for Transport and Digital Infrastructure is a federal government agency of Germany with headquarters in Berlin and Bonn

BNR Business needs and requirements describe the business solution w.r.t. the capabilities required to meet the business needs which describe the business goals, objectives, and problems that the business is trying to solve

BPaaS Business Process as a Service is a form of business process outsourcing (BPO) that employs a cloud computing service model

BSD Blind Spot Detection is a technique that provides 360 degrees of electronic coverage around a vehicle

B2B Business-to-business refers to business that is conducted between companies, rather than between a company and individual clients

B2B2C Business-to-business-to-customer is a model where online, or e-commerce, businesses and portals reach new markets and customers by partnering with consumer-oriented product and service businesses

BMW Bavarian motor manufacturer is the parent company of the BMW Group, a globally operating German automobile and motorcycle manufacturer based in Munich, the capital of the state of Bavaria

BOM Bill of material is a list of raw materials, subassemblies, intermediate assemblies, subcomponents, parts, and the quantities of each needed to manufacture an end product

BVDW Bundesverband für die Digitale Wirtschaft (Federal Association of the Digital Economy) is the central interest representation for companies that operate digital business models and are active in the area of digital value creation

BYD BYD Auto Company is a car manufacturer in Shenzhen, Guangdong Province, in the People's Republic of China and a subsidiary of BYD Company Ltd. The company is one of China's largest automobile manufacturers

C

CaaP Car-as-a-platform is a model for third party development and applications related to in-car connected platforms, offering a selection of features in connected vehicles with a special focus on entertainment apps and safety-management features

CAD Computer-aided design is the use of computer systems supporting analysis, creation, modification, or optimization of a design to increase the productivity of the designer, and improving the design quality and the communications required for documentation, and to create a database for the manufacturing process

CAE Computer-aided engineering is the process of solving engineering problems through the use of sophisticated, interactive graphical software in a factory-based environment

CAESS Center for Automotive Embedded Systems Security is collaboration between researchers at the UC San Diego and the University of Washington with the research mission ensuring security, privacy, and safety of future automotive embedded systems

CAM Computer-aided manufacturing is an application technique using computer software and machinery to facilitate and automate manufacturing processes

CAN Controller area network is a serial bus network that connects devices, sensors, and actuators in a system or subsystem for real-time control application

Car2Go Car2Go is a carsharing provider of the German automaker Daimler

CAS Collision Avoidance (precrash) System is an automobile safety system designed to avoid accidents or at least reduce the severity of an accident

CASE Stands for the strategic future areas of networking (connected), autonomous driving (autonomous), flexible use (shared), and electric drives (electric), which Mercedes-Benz Cars consistently drives forward and intelligently connects

CATIA Computer-Aided Three-Dimensional Interactive Application is a multi-platform software suite for CAD, CAM, CAE, PLM, and 3D

CBS Cloud-based server is a logical server that is built, hosted, and delivered through a cloud computing platform over the Internet

- CC** Cloud computing is a computing-infrastructure and software model for enabling ubiquitous access to shared pools of configurable resources such as applications and services, computer networks, servers, and storage devices
- CCaaDP** Connected-car-as-a-digital-platform is a model of automakers offering a selection of features in their connected vehicles, with a special focus on entertainment apps and safety-management features
- CCD** Charge coupled device is an integrated circuit etched into a silicon surface forming light-sensitive elements, called pixels
- CCG** Connected car gateway connects the vehicle to the outside world, using multiple wireless technologies
- CCRP** Connected Car Reference Platform is a powerful connectivity platform designed to support a wide range of innovative applications and experiences
- CCS** Combined Charging System is a quick charging method for battery electric vehicles delivering high-voltage direct current via a special electrical connector
- CCU** Central control unit is a powerful microprocessor-based control device
- CDO** Chief Digital Officer
- CE** Concurrent engineering is a work methodology emphasizing the parallelization of tasks which is sometimes synonymously called simultaneous engineering (SE) or integrated product development (IPD)
- CEC** Common engineering client is a project for the development, maintenance, and expansion of the engineering client (EC) as a uniform system-wide user interface for the Daimler product development process and the use in downstream processes
- CED** Canny edge detection is a popular edge detection algorithm developed by John F. Canny
- CHAdeMO** Charge De Move is the brand name of a quick charging method for battery electric vehicles delivering up to 62.5 kW of direct current (500V, 125A) via a special connector
- CIA** Confidentiality, integrity, and availability; see AIC
- CORBA** Common Object Request Broker Architecture is an architecture and specification for creating, distributing, and managing distributed program objects in a network.
- CO_x** Carbon oxide whereby x indicates the index number representing the three oxides of carbon: carbon dioxide, carbon monoxide, carbon sub-oxide
- CPE** Common Platform Enumeration is a standardized method of describing and identifying classes of applications, operating systems, and hardware devices present among an enterprise's computing assets
- CPS** Cyber-physical systems are systems where information and software are connected to physical components, with data transfer and exchange, as well as real-time control over an infrastructure such as the Internet
- CPSEF** Cyber-physical systems engineering framework promises higher productivity and shorter time to market (than non-framework-based approaches) through design and code reuse
- CRC** Cyclic redundancy check is an error-detecting code commonly used in digital networks and storage devices to detect accidental changes to raw data

CRM Customer relationship management describes a strategy for the systematic design of all relationships and interactions of a company with existing and potential customers

CS Crosswind stabilization belongs to ADAS compensating strong crosswinds

CS&C Office of Cyber Security and Communications is responsible for enhancing the security, resilience, and reliability of cyber and communications infrastructure in the USA

CSMA/CA Carrier Sense Multiple Accesses with Collision Avoidance is a protocol for carrier transmission in 802.11 networks preventing collisions before they happen

C2C Car-to-car is used in the English language literature as vehicle-to-vehicle (V2V) communication. V2V communication is only a sub-point of V2X communication

C2I Car-to-infrastructure is a communication model that allows vehicles to share information with the components that support a country's highway system

CVE Common Vulnerabilities and Exposures Database is a dictionary of common names for publicly known cybersecurity vulnerabilities

CVW Closing vehicle warning is defined as a function that detects closing vehicles in one or more of the rear zones and warns the vehicle driver

CWS Collision warning system discriminates between objects which pose a threat of collision from those which do not by measuring the relative sight line rate of the object

D

DAC Digital-to-analog converters translate digital signals into analog electrical signals for signal processing purposes

DAPRA Defense Advanced Research Projects Agency is an agency of the US DoD which carries out research projects for the US forces

DAS Driver assistance systems monitor the vehicle surroundings and the driving behavior to detect potentially dangerous situations at an early stage

DAT German Automotive Trust is the information center for the automotive industry

DAX German Stock Exchange Index is a measure which indicates the average price of the thirty most important German shares

D-Bus Diagnostic Bus is a specialized internal communication network that interconnects components inside a vehicle. Protocols include CAN, LIN, and others

DBN Deep belief network is a graphical model which learns to extract a deep hierarchical representation of the training data

DC Direct current is the unidirectional flow of electric charge

DADSS Driver Alcohol Detection System for Safety automatically detects when a driver is intoxicated with a BAC at or above 0.08 and prevents the car from moving

DCU Door Control Unit is a generic term for an embedded system that controls a number of electrical systems associated with a vehicle

DDD Driver drowsiness detection is a safety technique which helps prevent accidents caused by the driver getting drowsy

DDoS Distributed denial-of-service attack is an attack in which multiple compromised computer systems attack a target, such as a server, website, or other network resource, and cause a DoS for users of the targeted resource

DDS Data Distribution Service is an OMG machine-to-machine standard that aims to enable scalable, real-time, dependable, high-performance, and interoperable data exchanges using a publish-subscribe pattern

DHS U.S. Department of Homeland Security has a vital mission to secure the nation from the many threats she faces

DL Deep learning is part of machine learning methods based on learning data representations as opposed to task-specific algorithms, whereby learning can be supervised, partially supervised, or unsupervised

DMU Digital mock-up is a computer-generated trial model used to replace expensive real product/system testing by computer simulation

DNN Deep neural networks are distinguished from the more commonplace single-hidden-layer neural networks by their depth in the number of node layers through which data passes in a multistep process of pattern recognition which uses each layer of node trains on a distinct set of features based on the previous layer's output.

DNS Domain name system means that Internet domain names are located and translated into Internet Protocol (IP) addresses

DoS Denial of service attacks typically flood servers, systems, or networks with traffic in order to overwhelm the attacked resources and make it difficult or impossible for legitimate users to use them

DP Digital prototyping allows conceptual design, engineering, manufacturing, sales, and marketing departments the ability to virtually explore a complete product before it is built

DPA Differential power analysis is a cryptoanalysis with which the encryption, e.g., of smartcards or other encryption components can be determined and the secret key can be derived

DSDM Dynamic systems development method is one of the leading agile approaches, bringing together the agility and flexibility necessary for successful organizations within a framework of the appropriate level of project governance

DSRC Dedicated short range communication

E

EA Emergency assist monitors the activity of the driver, such as accelerator pedal, brake, and steering, and helps to avoid accidents within the system limits and to reduce possible accident sequences

EBA Emergency brake assistant is a vehicle braking technique that increases braking pressure in an emergency case

EBCM Electronic brake control module is a control system used to operate brakes simultaneously, provided that the functions and the operating mode of the brake systems are identical

- EBD** Electronic brake force distribution is a vehicle braking technique that automatically varies the amount of force applied to each of a vehicle's wheels, based on road conditions, speed, and loading
- EBOM** Engineering bills of material is a type of BOM reflecting the product as designed by engineering, referred to as the as-designed bill of materials
- e-Call** Emergency call is an automatic emergency call system for vehicles planned by the EU to be compulsorily incorporated into all new models of passenger cars and light commercial vehicles from March 31, 2018
- ECM** Engine control module controls a series of actuators on a vehicle's combustion engine to ensure an optimal engine performance
- ECU** Electronic control unit is any embedded system that controls an electrical system or subsystem in a vehicle
- E/E** Electrical and electronic systems refer to different components of the on-board network. Electrical components are capacitors, inductors, relays, resistors, switches, and others. Electronic components are application-specific integrated circuits (ASICs), integrated circuits (ICs), field programmable gate arrays (FPGAs), microcontrollers (μ C), microprocessors (μ P), and others
- E/E/PE** Electrical, electronic, and programmable electronic refers to complex systems using computer-based technology
- EEPROM** Electrically erasable programmable read-only memory is a PROM that can be erased and reprogrammed using an electrical charge
- EGNOS** European Geostationary Navigation Overlay Service is a satellite-based augmentation system (SBAS) developed by ESA and EUROCONTROL on behalf of the EU, supplementing GPS, GLONASS, and Galileo satellite navigation system
- EGR** Exhaust gas recirculation is an effective strategy to control NOx emissions from diesel engines reducing NOx through lowering the oxygen concentration in the combustion chamber, as well as through heat absorption
- ELP** Electronic license plate is an identification sign mounted on vehicles that emits wireless signals used for tracking and digital monitoring services
- EMC** Electromagnetic compatibility is the interaction of electrical and electronic equipment with its electromagnetic environment and with other equipment
- EMD** Electromagnetic discharge refers to removing a static field/load from equipment
- EMI** Electromagnetic interference is the disruption of operation of an electric device when it is in vicinity of an electromagnetic field in the RF spectrum caused by another electronic device
- EMNS** Entry media and navigation system is a common infotainment and navigation system
- E911** Emergency call is a system used in North America that links emergency callers with the appropriate public resources
- EOC** End of conversion is the time required to convert an analog or a digital signal into a digital or an analog signal

EPROM Erasable programmable read-only memory is a type of memory chip that retains its data when its power supply is switched off

EPS Electronic power steering assists the driver of a vehicle, unlike traditional systems that act on hydraulic pressure provided via a pump driven by the vehicle's engine, whereby this pump is constantly running, whether the steering wheel is being turned or not

ERP Enterprise resource planning is the integrated management of core business processes, often in realtime

ESA European Space Agency is an intergovernmental organization dedicated to the exploration of space

ESC Electronic stability control is a computerized technology that improves vehicle's stability by detecting and reducing loss of traction. ESC is also referred to as ESP

ESP Electronic stability program is one of the most important safety systems on vehicles which improve lateral dynamics, thus ensuring stable driving in all directions

EssUP Essential unified process focuses on the essentials to provide eight light-weight, easy-to-use practices that can be mixed and matched and used in different circumstances, all of them compatible with agile values and thinking

ETA Estimated time of arrival is the time when a vehicle or emergency service is expected to arrive at a certain place

EU The European Union is a political and economic union of 28 member states located primarily in the continent of Europe, holding approx. 40 percent of the area of continental Europe

EVWS Electric vehicle warning sounds for hybrids and electric vehicles are a series of sounds designed to alert pedestrians to the presence of electric drive vehicles as well as hybrid electric vehicles

EWSV Emergency Warning System for Vehicles is a telematics concept developed particularly for international harmonization and standardization of V2V, R2V, and V2R real-time dedicated short-range communication

F

FAA Federal Aviation Administration's mission is to provide the safest, most efficient aerospace system in the world

FCA Fiat Chrysler Automobiles is an Italian-controlled multinational corporation incorporated in the Netherlands

FDD Feature-driven development is a client-centric, architecture-centric, and pragmatic software process

FDIS Final Draft International Standard refers to ISO 9001:2015

FEM Finite element method is a numerical method for solving problems of engineering and mathematical physics

3G Third-generation wireless mobile radio standard. 3G is understood as Universal Mobile Telecommunications System (UMTS), High-Speed Downlink Packet

Access (HSPA), as well as HSPA +. Surfing speeds achieved with HSPA+ is up to 28 Mbit/s; surfing speed available with HSPA is up to 5.5 Mbit/s

4G Fourth-generation wireless mobile radio standard. 4G enables significantly higher surfing speeds than the 3G standard allowing theoretically downloads up to 300 Mbit/s

5G Fifth-generation wireless mobile radio standard is the proposed next telecommunication standard beyond the current 4G/IMT advanced standards

FIU Fault insertion units are designed to insert fault conditions between automated test equipment and devices under test

FM Frequency modulation is widely used for a variety of radio communications applications, and it is especially useful for mobile radio communications, being used in taxis and many other forms of vehicles

FMEA Failure mode and effects analysis is a qualitative and systematic tool, usually created within a spreadsheet, to help practitioners anticipate what might go wrong with a product or process.

FMECA Failure, mode and effects, and critical analysis is a step-by-step approach for identifying all possible failures in a design, a manufacturing or assembly process, or a product or service

FMI Functional mock-up interface is a tool-independent standard to support both model exchange and co-simulation of dynamic models using a combination of XML files and compiled C-code

FPGA Field programmable gate array is an integrated circuit that can be programmed in the field after manufacturing

G

GA Genetic algorithms are methods for solving both constrained and unconstrained optimization problems which are based on natural selection

GCN Global communication networks are filtering trading information provided via online GCN, providing information about financial investments, and trading execution via online GCN

GDI Gross domestic income is the sum of all income earned while producing goods and services within a nation's borders

GENIVI GENIVI® is a nonprofit industry alliance committed to driving the broad adoption of open source, IVI software and providing open technology for the connected car

GHz Gigahertz is a measure of frequency equivalent to 10^9 cycles per second

GIS Geographical information system is an information system for the collection, processing, organization, analysis, and presentation of spatial data

GLONASS Global Navigation Satellite System using GPS, GLONASS, Galileo, or the Chinese satellite system BeiDou in many applications such as local awareness

GM General Motors is an American multinational corporation that designs, manufactures, markets, and distributes vehicles and parts and sells financial services

GMRF Gaussian Markov Random Field models are most widely used in spatial statistics

GND Ground is the reference point for all signals or a common path in an electrical circuit where all of the voltages can be measured from

GNSS Global Navigation Satellite System is a system that uses provided autonomous geo-spatial positioning

GNUGPL GNU General Public License is a widely used free software license which guarantees end users the freedom to run, study, share, and modify the software

GPRS General Packet Radio Service is a packet-based wireless communication service that promises data rates from 56 up to 114 Kbps and continuous connection to the Internet for mobile phone and computer users

GPS Global Positioning System is a US space-based radio navigation system operated by the US Air Force that helps pinpoint a 3D position to about a meter of accuracy w.r.t. latitude, longitude, and altitude

GPU Graphic processing unit is a specialized and optimized processor for computers, game consoles, and smartphones

GSA Greenbone Security Assistant is a web application that connects to the OpenVAS Manager and OpenVAS Administrator to provide for a full-featured user interface for vulnerability management

GSM Global system for mobile communication is a standard for fully digital mobile networks, which is mainly used for telephony but also for circuit-switches and packet-switches, data transmission, as well as short messages

GUI Graphical user interface software that works at the point of contact between a computer and its user

H

HCD Head-coupled displays share common elements with immersive VR systems with the goal presenting a realistic, stable computer-generated scene

HD High density is a technology which has a considerably higher power density than the standard technology

HDC Hill descent control is a vehicle safety feature to facilitate safe travel down steep grades

HDTRI High-definition traffic real-time information is used to broadcast real-time information to vehicles

HFCPI Hands-Free Cell Phone Interfaces connect headset hands-free and voice dialing hands-free cell phone devices for telecommunication in vehicles

HIL Hardware-in-the-loop is a technique that is used in the development and test of complex real-time embedded systems

HMD Head-mounted display is a device worn on the head or as part of a helmet that has a small display optic in front of one or each eye

HMI Human machine interface is the user interface that connects an operator to the controller for an industrial system

HP Hewlett-Packard is an American multinational information technology company headquartered in Palo Alto, California

HRTS Hard Real-Time Systems is hardware or software that must operate within the constraints of a stringent time limit

HTML5 Hypertext Markup Language is a markup language used for structuring and presenting content on the World Wide Web

HTTPS Hypertext Transfer Protocol Secure is a communication protocol on the World Wide Web for transferring data securely

HVAC Heating, ventilation, and air control is the technology of indoor and vehicular environmental comfort

HW Hardware is the key term referring to the mechanical and electronic equipment of a data processing system

I

IA Intersection assistance is an ADAS that monitors cross-traffic in an intersection/road junction. If the IA detects a hazardous cross-traffic situation, it prompts the driver to start emergency braking by activating visual and acoustic warnings and automatically engaging brakes

IAA International motor show located in Frankfurt/Main (passenger cars) or Hannover (commercial vehicles) shows the latest trends in cars and mobility

IaaS Infrastructure as a Service is a form of cloud computing that provides virtualized computing resources over the Internet

IAM Identity and Access Management is becoming more and more important through the decentralization of systems, the increased use of mobile devices, and the global access to cloud infrastructures to know which user needs which rights for systems and applications and how they use these rights on which device avoiding getting a problem sooner or later with unauthorized access or data usage

IB Interface builder is software within the Xcode IDE that makes it simple to design a full user interface without writing any code

IBM International Business Machines is an American multinational technology company

ICC International color consortium provides color management systems with the information necessary to convert color data between native device color spaces and device independent color spaces, called the Profile Connection Space (PCS)

ICE Internal combustion engine powertrains are dominating the vehicle market today due to their reliability and drivability

ICT Information and communication technology encompasses the infrastructure and components which enable modern computing

IDE Integrated development environment is a software application that provides a programming environment to streamline developing and debugging software

IDS Intrusion detection system is a device or software application that monitors a network or systems for malicious activity or violations

IDPS Intrusion detection and prevention systems are two different solutions in that one is a passive detection monitoring system and the other is an active prevention system

- IEC** International Electrotechnical Commission is the world's leading organization that prepares and publishes international standards for all electrical, electronic, and related technologies
- IEEE** Institute of Electrical and Electronic Engineers is the world's largest technical professional organization dedicated to advancing technology for the benefit of humanity
- IHC** Intelligent headlight control uses a video camera to measure the ambient brightness and to estimate the distance from vehicles in front and oncoming traffic
- IoDaS** Internet of Data and Services interconnecting data, services, and people through the Internet and improving data analysis, boosts productivity, enhances reliability, and generates new revenue opportunities through innovative business models
- IoE** Internet of Everything can be defined as the intelligent connection of people, processes, data, services, and things
- iOS** Operating system of the company Apple for mobile devices developed for the iPhone, iPad, iPad mini, and iPod touch
- IoT** Internet of Things means networking of objects/things with the Internet, so that these objects/things can communicate independently over the Internet doing different tasks for the owner
- IP** Internet Protocol
- IPSA** Intrusion prevention system architecture determines what assets to protect, the sensitivity of those assets, and the confidentiality, integrity, and availability requirements of the identified assets
- IFT** Image Processing ToolboxTM provides a comprehensive set of reference-standard algorithms and workflow apps for image processing, analysis, visualization, and algorithm development
- IPv4** Internet Protocol, version 4 is the 4th revision of the Internet Protocol and a widely used protocol in data communication over different kinds of networks
- IPv6** Internet Protocol, version 6, is a set of specifications from the Internet Engineering Task Force (IETF) that's essentially an upgrade of IPv4
- IS** Intersection support provides a way to asynchronously observe changes in the intersection of a target element with an ancestor element
- ISO** International Standard Organization is an international standard-setting body composed of representatives from various national standards organizations
- IT** Information technology refers to anything related to computing technology, such as hardware, networking, software, the Internet, or people that work with these technologies
- ITIS** Intelligent transportation information systems use advanced communication, information, and electronics technology to solve transportation problems such as traffic congestion, safety, transport efficiency, and environmental conservation
- ITU** International Telecommunication Union is an agency of the United Nations (UN) whose purpose is to coordinate telecommunication operations and services throughout the world

- IV** Initialization vector is a fixed-size input to a cryptographic primitive that is typically required to be random or pseudorandom
- IVI** In-vehicle infotainment is a collection of hardware and software in vehicles that provides audio or video entertainment

K

km Kilometer is an abbreviation of distance

KPIT An IT service and consulting company based in Pune, India which offers solutions for medium and large companies in the fields of automotive electronics, industrial automation and chip design, business IT, as well as IT services for banks and insurance companies

kWh kWh is an abbreviation of kilowatt-hour and is the amount of energy that is converted at an output of one kilowatt (1kW) within 1 hour

L

LAN Local area network encompasses computers and peripherals connected to a server within a distinct geographic area such as an office or a commercial establishment

LAS Lateral acceleration sensors used to measure the lateral acceleration acting on the vehicle and calculate the vehicle's actual position

LCA Lane Change Assistant uses two mid-range radar sensors that are concealed in the rear bumper, one on the left and one on the right, which monitor the area alongside and behind the car, whereby control software collates the sensor information to create a complete picture of all traffic in the area behind the vehicle

LDW Lane departure warning alerts the driver if his vehicle is drifting out of the lane using visual, vibration, or sound warnings

LED Light emitting diode is a two-lead semiconductor light source, emitting light when activated

LHW Local hazard warning is some kind of cooperative awareness messages only in that they are transmitted by roadside

LiDAR Light detection and ranging is a remote sensing method that uses light in the form of a pulsed laser to measure ranges, variable distances

LIN Local interconnect network is a serial network protocol used for communication between components in vehicles

LKA Lane keeping assistant combines the functions of BSW and CVW

LKS Lane keeping system uses a camera that can identify lane divisions and works proactively to keep the vehicle within a detected lane

LLVM Low-Level Virtual Machine is a collection of libraries and tools that make it easy to build compilers, optimizers, Just-In-Time code generators, and many other compiler-related programs

LOC Lines-of-code referring to non-commentary lines, meaning pure white space and lines containing only comments, are not included in the metric

LOD Level of detail increases the efficiency of rendering by decreasing the workload on graphics pipeline stages

LSE Large-scale engineering is the process of integrating several components or system into an engineered device

LTE Long-Term Evolution is a 4G wireless broadband technology developed by the 3G Partnership Project (3GPP) of an industry trade group

M

MaaS Mobility-as-a-Service provides an alternative way to move more people and goods in a way that is faster, cleaner, and less expensive than current options

MAC Message authentication code is a cryptographic checksum on data that uses a session key to detect both accidental and intentional modifications of the data

MAN Maschinenfabrik Augsburg-Nuremberg is a vehicle and mechanical engineering group called MAN SE (Maschinenfabrik Augsburg-Nuremberg Societas Europaea)

MANET Mobile ad hoc network is a network that can change locations and configure itself on the fly

MITM Man-in-the middle attack is a general term for when a perpetrator positions himself in a conversation between a user and an application—either to eavesdrop or to impersonate one of the parties, making it appear as if a normal exchange of information is underway

MBOM Manufacturing bill of material, also referred to as the manufacturing BOM, contains all the parts and assemblies required to build a complete and shippable product

Mbps Megabit per second is used to measure data transfer speeds of high-bandwidth connections, such as Ethernet and cable modems

MEMS Microelectromechanical systems is a technology that can be defined as miniaturized mechanical and electromechanical elements like devices and structures that are made using the techniques of microfabrication

MIL Model in the loop is a technique used to abstract the behavior of a system or sub-system in a way that this model can be used to test, simulate and verify that model

MID Misuse Intrusion Detection actively works to detect potential intrusion threats to vulnerable data

MISRA Motor Industry Software Reliability Association is a programming standard from the automotive industry

MITRE A not-for-profit company that operates multiple federally funded research and development centers created by the split from the Massachusetts Institute of Technology (MIT)

MMS Multimedia Messaging Service offers the possibility to send multimedia messages to other mobile devices or to normal e-mail addresses with a mobile phone

MOM Message-oriented middleware is a specific class of middleware that supports the exchange of general-purpose messages in a distributed application environment

MOST Media-oriented systems transport is the de facto standard for multimedia and infotainment networking in the automotive industry

MPV Multipurpose vehicle is a type of vehicle favored by families due to a more practical interior than a regular vehicle

MRF Markov Random Field is a graphical model of a joint probability distribution

MRP Materials requirements planning is a production planning, scheduling, and inventory control system used to manage manufacturing processes

MTBF Mean time between failures is the predicted elapsed time between inherent failures of a system, during normal system operation, and is calculated as the arithmetic mean (average) time between failures of a system

M2M Machine-to-machine communication is often used for remote monitoring

MTTF_d The abbreviation for the mean time to a dangerous failure

MVC Model view controller is a software architectural pattern for implementing user interfaces on computers

MySQL My Standard Query Language is one of the most used relational database management systems, available as open source software as well as commercial enterprise version for various operating systems, and forms the basis for many dynamic websites

N

NAFTA North American Free Trade Agreement is one of the world's largest free trade zones for strong economic growth and rising prosperity for Canada, the USA, and Mexico

NCS Network control system is a control system wherein the control loops are closed through a communication network

NEC Nippon Electric Company offering digital display solutions

NECS Networked embedded computing systems typically consist of multiple computers that are connected by a wireless or wired network

NFCW Near field collision warning represents a significant leap in vehicle safety technology by attempting to actively warn vehicle drivers of an impending collision event, thereby allowing the vehicle driver adequate time to take appropriate corrective actions in order to mitigate or completely avoid the event

NHTSA National Highway Traffic Safety Administration is responsible for keeping people safe on America's roadways

NI National Instruments is an American multinational company with international operation which produces automated test equipment and virtual instrumentation software

NICB National Insurance Crime Bureau is a not-for-profit organization that receives support from nearly 1,100 property and casualty insurance companies and self-insured organizations

NIST National Institute of Standards and Technology is one of the USA's oldest physical science laboratories

NoSQL Non-SQL provides a mechanism for storage and retrieval of data that is modeled in means other than the tabular relations used in relational databases

NOx Oxides of nitrogen, abbreviated with NOx, since there are several nitrogen-oxygen compounds due to the many oxidation states of the nitrogen

NS Navigation system is a part of the vehicle control used to find a direction in a vehicle which can be achieved by calculating the route

NSF National Science Foundation's mission is to advance the progress of science, a mission accomplished by funding proposals for research and education made by scientists and engineers

NTG New Telematics Generation is equipped with the advanced, more user-friendly generation of telematics technology

NURBS Nonuniform rational B-splines are mathematically defined curves or surfaces which are used in the computer graphics area for modeling shapes or forms

NVD National Vulnerability Database is the US government repository of standards based on vulnerability management data represented using the Security Content Automation Protocol (SCAP)

NVP Night vision plus can alert the vehicle driver to the potential danger posed by pedestrians or animals in unlit areas in front of the vehicle by automatically switching from the speedometer to a crystal-clear night view image and highlighting the sources of danger, whereby a spotlight function is able to flash any pedestrians detected ahead

NVT Network vulnerability tests are a software testing technique performed to evaluate the quantum of risks involved in the system in order to reduce the probability of the event

NXP Next eXPerience (formerly Philips Semiconductors) is driving innovation in the secure connected vehicle. End-to-end security and privacy and smart connected solutions

O

OAIT Open Artificial Intelligence Technologies mission is to build safe artificial general intelligence (AGI) and ensure AGI benefits are as widely and evenly distributed as possible

OBD Onboard diagnostics is a computer-based system originally designed to reduce emissions by monitoring the performance of major engine components

O-D Origin-destination is a model created to understand travelers' true origins and destinations for any specific trip

OSW Obstacle and collision warning alerts vehicle drivers that the vehicle is in immediate danger colliding with an obstacle

OEM Original Equipment Manufacturer is a company that produces parts and equipment that may be marketed by another manufacturer

OMG Object Management Group is a consortium founded in 1989, which deals with the development of standards for vendor-independent system-wide object-oriented programming

OOA Object-oriented analysis in an object-oriented variant of the analysis process and the design process as part of the development of a software system

OPENSIG One-Pair Ethernet Alliance Special Interest Group is an interest group that promotes the introduction of BroadR-Reach, an Ethernet based communication technology in motor vehicles

OpenUP Open Unified Process is a lean unified process that applies iterative and incremental approaches within a structured life cycle embracing a pragmatic, agile philosophy that focuses on the collaborative nature of software development

OpenVAS Open Vulnerability Assessment System is a framework of multiple services and tools that together provide a comprehensive and powerful solution for vulnerability scanning and vulnerability management

OS Operating system is the most important program that runs on a computer

OSI Open systems interconnection is a reference model for how applications can communicate over a network

OSS Operating system scheduler is an essential part of a multiprogramming OS

OTA Over the air refers to various methods of distributing new software, configuration settings, and even updating encryption keys to devices and systems

P

PaaS Platform as a Service is a service that provides a computer platform for developers of web applications in the cloud

PAM Process assessment model holds all details to determine process maturity and relates to one or more PRMs

PAYD Pay as you drive is a type of motor vehicle liability insurance in which the premium is calculated from the quantity and type of vehicle use

PC Personal computer is a multipurpose computer operated directly by the end user, which size, capabilities, and price make it feasible for individual use

PCAST U.S. President's Council of Advisors on Science and Technology is a council, chartered in each administration, with a broad mandate to advise the President on science and technology

PCB Printed circuit board is used to mechanically support and electrically connect electronic components using conductive pathways, tracks, or signal traces etched from copper sheets laminated onto a nonconductive substrate

PCCMS Precrash collision and mitigation system makes use of unique data fusion algorithms that combine the input from radar and vision sensors to enhance safety system functionality by warning vehicle drivers if it estimates a high risk for collision when the equipped vehicle approaches a pedestrian or another vehicle

PCM Powertrain control module is an automotive control component used in vehicles, consisting of the engine control unit and the transmission control unit

PDM Product data management is the holistic company-wide management and control approach of all product data and processes of the entire life cycle from development and production through sales and maintenance

PFD Probability of failure on demand can be done by means of a so-called Markov model

PHYD Pay how you drive is a type of car liability insurance in which the premium is calculated from the quantity and type of vehicle use

- PID** Proportional-integral-differential controller is a control loop feedback device widely used in industrial control systems
- PIN** Personal identification number is a number known only to one or a few persons, with which they can authenticate themselves against a machine
- PL** Performance level is defined according to standard EN 13849 as discrete level that specifies the ability of safety-related parts of a control to perform a safety function under predictable conditions which means it is a measure of the reliability of a security function
- PLC** Product life cycle is the process of managing the entire life cycle of a product from inception, through engineering design and manufacture, to service and disposal of manufactured products
- PLM** Product life cycle management takes into consideration the entire vision of effectively managing and connecting all information related to the process and production data needed to design, produce, validate, support, maintain, and ultimately dispose of manufactured goods
- PMM** Power Management Module is a device feature that allows controlling the amount of electrical power consumed by underlying devices, with minimal impact on performance
- PNT** Positioning, navigation, and timing services is used in combination with map data and other information like weather or traffic data and is the most popular modern navigation system better known as GPS
- POI** Point of interest is a specific point location that someone may find useful or interesting
- PPP** Public private partnership represents the involvement of private economic entities in the execution of public tasks
- PRM** Process reference model describes for a certain application domain a set of processes, whereby each process is described by its purpose and the associated process outcomes. It is always related to a PAM which holds all details to determine the maturity of the processes of the reference model
- PROM** Programmable read-only memory is a read-only memory that can be modified once by a user allowing him to tailor a microcode program using a special machine called PROM programmer
- PS** Parking sensors are embedded to alert the driver to obstacles while parking by measuring distances to nearby objects
- PSAP** Public safety access points are selected in a wireless network for E911 calls
- PSI5** Peripheral sensor interface 5 is an open standard for automotive sensor applications based on existing sensor interfaces, e.g., for peripheral airbag sensors
- PSS** Passive safety system refers to a system that protects vehicle drivers and passengers during a crash, primarily airbags and seatbelts
- PUC** Pollution under control is a valid certification that's granted to a vehicle that has passed the PUC test saying that the vehicular emissions are under control and in accordance with the pollution norms
- PwC** PricewaterhouseCoopers is a multinational professional services network and is one of the big four auditors, along with Deloitte, EY, and KPMG

PWDC Power window and door control are operated by an electric motor and has different types of settings in a vehicle enabling a vehicle driver to open or close any vehicle windows of interest as well as activate the door locks when the ignition key is turned on and the vehicle starts moving. Another setting in PWDC is the automatic window roll-up when the ignition key is turned off and the driver's door is opened or the electric locks are set to lock the vehicle

R

RAM Random access memory is a computer data storage device which stores frequently used program instructions to increase the general speed of a system

R&D Research and development refers to the investigative activities a business conducts to improve existing products and procedures or to lead to the development of new products and procedures

RC4 Rivest Cipher 4 is a stream cipher which is a symmetric key cipher where plaintext digits are combined with pseudorandom cipher digit stream

RDA Rural drive assistance is applied for analysis of rural road drive and alternatives

RDS Radio data system is a protocol for transmitting additional information to a radio transmission

RDP Road departure protection is a special ADAS that can be used to prevent a deviation from the road

RF Radio frequency refers to the rate of oscillation of electromagnetic radio waves in the range of 3 kHz to 300 GHz

RFID Radio frequency identification is a technology that incorporates the use of electromagnetic or electrostatic coupling in the radio frequency (RF) portion of the electromagnetic spectrum to uniquely identify objects or things

RKE Remote keyless entry is an electronic lock that controls access to a vehicle without using a traditional mechanical key

RKI Remote keyless ignition system refers to an electronic remote control as a key which is activated by pushing a button for a keyless start

RMS Root mean square refers to the mathematical method defining the effective voltage or current of an AC wave

RoI Region of interest is a portion of an image that users want to filter or perform some other operation on

ROI Return on investment is a metric which can be used as a rudimentary gauge of investment's profitability

ROS Robot operating system is a software framework for personal robots; the development began in 2007 at the Stanford Artificial Intelligence Laboratory within the Stanford AI Robot project (STAIR)

RPC Remote procedure call is a protocol that one program can use to request a service from a program located in another computer on a network without having to understand the network's details

RRF Risk reduction factor determines the SIL

RSA Rivest-Shamir-Adleman is a cryptosystem for public key encryption which is widely used for securing sensitive data, particularly when being sent over an insecure network such as the Internet

RSR Road sign recognition is an ADAS technique by which a vehicle is able to recognize the traffic signs on the road

RSU Roadside unit is a computing device located on the roadside that provides connectivity support to passing vehicles

RTE Runtime environment is the execution environment provided to an application or software by the operating system

RTV Roadside-to-vehicle in the connected car context means that there is some kind of data backbone, an internet gateway, which connects the vehicle and roadside sensors

RUP Rational unified process is an iterative software development process framework which is not a single concrete prescriptive process but rather an adaptable process framework, intended to be tailored by the development organizations and software project teams who will select the elements of the process that are appropriate for their needs

RVS Rear view system increases the field of view for the vehicle driver and detects additional information for fusion with other parking systems

S

SaaS Software as a Service is part of cloud computing, whereby the SaaS model is based on the principle that the software and the IT infrastructure are operated by an external IT service provider and are used by the customer as a service

SAN Sensor and actuator network is a network of sensor nodes that can measure data in a system and a network of actuators capable of modifying this system based on the processed sensor data

SAP Systems Applications and Products in Data Processing is a leading provider of enterprise software

SC Seat comfort is a subjective topic because drivers and passengers are all shaped differently and have their own opinion as to which seat is more comfortable. The best seats for sore backs are those that offer adequate lumbar support, leg support, and a high degree of adjustability

SCA Side channel attack makes it possible for an attacker without access to the system itself to deduce how the system works and what data it is processing

SCADA Supervisory control and data acquisition is a system of software and hardware elements to maintain efficiency, process data for smarter decisions, and communicate system issues to help mitigate downtime

SCU Speed control unit is designed to precisely control engine speed with rapid responses to transient load changes

SD Service discovery is the automatic detection of devices and services offered by these devices on a computer network

SDF Sensor data fusion means combining sensor data derived from disparate sensors such that the resulting information has less uncertainty than would be possible when these sensor data were used individually

- SDK** Software development kit is a toolkit for software engineers that offering an easy access to a special operating system or a programming language
- SDP** The service discovery protocol is a network protocol that helps accomplish service discovery
- SE** Simultaneous engineering is a concurrent new product development through employing cross-functional teams to reduce cycle time
- SEND** Secure neighbor discovery protocol is a security extension of the neighbor discovery protocol (NDP) in IPv6
- SENT** Single-edge nibble transmission is a protocol for a point-to-point scheme for transmitting signal values from a sensor to a controller
- SFF** Safe failure fraction is a measure of the proportion of all possible faults in the safe direction
- S/H** Sample and hold is an analog device that samples the voltage of a continuously varying analog signal and holds its value at a constant level for a specified minimum period of time
- SHD** Sunroof is a fixed or operable opening in an automobile roof which allows light and/or fresh air to enter the diver/passenger compartment
- SIL** Safety integrity level is defined as a relative level of risk reduction provided by a safety function or to specify a target level of risk reduction
- SIM** Subscriber identity module is a smart card inside a GSM cell phone that encrypts voice and data transmissions and stores data about the specific user so that the user can be identified and authorized to the network supplying the phone service
- SIR** Susceptible-infected-recovered model is a classical approach for describing the spread of infectious diseases with immunity formation, which is an extension of the SI model, in mathematical epidemiology, a field of theoretical biology
- SIS** Susceptible-Infected-Susceptible is also known as the contact process model
- 6LoWPAN** Internet Protocol v6 and low-power wireless personal area network is the name of a concluded working group in the Internet area of the Internet engineering task force (IETF)
- SMS** Short message system is a way of sending short written messages from one mobile phone to another
- SMW** Smart mirrors and wipers allow improving the visibility if the mirrors had wipers in case of rainy weather conditions
- SNR** Stakeholder needs and requirements represent the views of those at the business or enterprise operations level as a set of requirements for a solution that can provide the services needed by the stakeholders
- SOA** Service-oriented architecture is a software design style where services are provided to other components by application components and by a communication protocol over a network
- SoC** System-on-a-chip is a microchip with all the necessary electronic circuits and parts for a given system, such as a smartphone or wearable computer, on a single integrated circuit

SOC Start of conversion is the time when converting an analog or a digital signal into a digital or an analog signal starts

SOME/IP Scalable service-Oriented Middleware over IP is an automotive middleware solution that can be used for control messages which was designed from beginning on to fit devices of different sizes and different operating systems perfectly

SOS Save our souls is used mostly by the military, but anyone around the world understands what it means

SPL Sound pressure level is a logarithmic quantity for describing the strength of a sound event

SPT Security penetration test is an authorized simulated attack on a vehicle's computer system that looks for security weaknesses, potentially gaining access to the system's features and data

SQL Structural Query Language is a standard computer language for relational database management and data manipulation

SRTS Soft Real-Time Systems devices with weak real-time constraints

STA Station or wireless end point is a device that has the capability to use the 802.11 protocol

STEP Standard for the exchange of product model data is an ISO standard for the exchange of CAD files with a standardized description of product and process data. STEP data are used in computer-assisted technologies such as CAD, CAE, and CAM, as well as PDM

S/N Signal-to-noise ratio is a measure of signal strength relative to background noise, usually measured in decibels (dB) using a signal-to-noise ratio formula

SUV Sports Utility Vehicle is a passenger car with driving comfort and increased off-road capability, classified as a light truck, but operated as a family vehicle

SW Software

SWAS Steering wheel angle sensors recognize how far the steering wheel is turned

SWOT Strengths-Weaknesses-Opportunities-Threats is a structure planning method that evaluates those four elements of a system, project, organization, or business venture

SyRS System requirement specification is a structured collection of information that embodies the requirements of a system

SysML Systems Modeling Language is a general-purpose modeling language for systems engineering applications supporting the specification, analysis, design, and V&V of a broad range of systems

T

TCM Transmission control module is a device that controls electronic automatic transmissions to calculate how and when to change gears in the vehicle for optimum performance

TCO Total cost of ownership is the purchase price of an asset plus the costs of operation

- TCU** Telematics control unit refers to the embedded system onboard of a vehicle that controls tracking of the vehicle
- TCS** Traction control system is a car safety feature that prevents wheels from spinning on low-grip surfaces
- TCP/IP** Transmission Control Protocol/Internet Protocol is a family of network protocols also known as an Internet Protocol family because of its great importance for the Internet
- TelCO** Telematics' Control Unit refers to the embedded system onboard a vehicle that controls the tracking of the vehicle
- TFS** TaxiForSure is a value-based cab company which cab aggregator Ola has shut down
- 3C** Computation, communication, and control represent three categories in ICT
- 3D** Three-dimensional describes an image that provides the perception of depth
- TKIP** Temporal key integrity protocol is a stopgap security protocol used in the IEEE 802.11 wireless networking standard
- TMC** Traffic message channel is a technology for delivering traffic and travel information to motor vehicle drivers
- TMCU** Transmission control unit is similar to an engine control unit, but it is responsible for the proper operation of a modern transaxle or transmission
- TOF** Time of flight describes a variety of methods that measure the time that it takes for an object, particle or acoustic, electromagnetic, or other waves to travel a distance through a medium
- TPS** Tire pressure sensor is a device to monitor the air pressure inside the pneumatic tires of vehicles and report real-time tire pressure information to the driver of the vehicle
- TPM** Trusted platform module is a standard that defines a hardware root of trust (HRoT) widely accepted as more secure than software that can be more easily breached by attackers
- TPMS** Tire pressure monitoring system is a real-time sensor-based pressure measurement device
- TRIC** Tahoe Reno Industrial Center in Storey County is Tesla's Gigafactory 1 for battery pack production
- TRW** Thompson Ramo Wooldridge is a former US automotive part supplier which was taken over by ZF
- TSR** Traffic sign recognition is an image processing technique by which a vehicle is able to recognize the traffic sign on the roads and is part of ADAS. The detection methods can be generally divided into color-based, shape-based, and learning-based methods
- TSS** Daimler TSSS GmbH is a wholly owned subsidiary of Daimler AG acting as a corporate IT service provider serving exclusively customers in the Daimler Group
- TTM** Time to market is the length of time taken in the product development process from the product idea to the finished product

U

UAV Unmanned autonomous vehicle is a machine that can move through the terrain intelligently and autonomously without the need for any human intervention

UBI Usage-based insurances are also known as pay as you drive (PAYD) and pay how you drive (PHYD) and mile-based vehicle insurance, whereby the cost depends upon the type of vehicle used, measured against time, distance, behavior, and place

UDS Unified diagnostic service is a diagnostic communication protocol in the ECU environment within the automotive electronics, which is specified in the ISO 14229-1

UK The United Kingdom is a sovereign state in Northern Europe based on a constitutional monarchy

UML Unified modeling language is a general-purpose modeling language that is intended to provide a standard way to visualize the design of a system

US/USA The United States of America is a country of central and northwest North America with coastlines on the Atlantic and Pacific Oceans including the non-contiguous states of Alaska and Hawaii and various island territories in the Caribbean Sea and Pacific Ocean

USB Universal serial bus is an industry standard that defines cables, connectors, and communication protocols for connection, communication, and power supply between computers and devices

US-CERT US-Computer Emergency Readiness Team strives for a safer, stronger Internet for all Americans by responding to major incidents, analyzing threats, and exchanging critical cybersecurity information with trusted partners around the world

USD US Dollar is the official currency of the USA and its insular territories per the US Constitution

V

VANET Vehicle ad hoc network is a technology that uses moving cars as nodes in a network to create a mobile network which could be treated as a subgroup of MANET

VAS Vehicle audio system is equipment installed in a vehicle to provide in-car entertainment and information for the vehicle users

V&V Verification and Validation are independent procedures that are used together for checking that a system/product meets requirements and specifications and that it fulfills its intended purpose

VBAT Battery voltage is the supplying electric power

VCM Vehicle control module is a configurable, multipurpose controller device developed to meet the requirements of vehicle applications

VCS Vehicular communication systems are networks in which vehicles and RSUs are the communicating nodes, providing each other with information, such as safety warnings and traffic information

VDA German Association of the Automotive Industry is an interest group of the German automotive industry, both automakers and automobile component suppliers

VDI German Association of Engineers is one of the largest technical and scientific associations in Europe. Its role in Germany is comparable to that of the American Society of Civil Engineers (ASCE) in the USA

VE Virtual environment is a computer-generated 3D representation of a setting in which the user perceives them to be and which interaction takes place

VEDS Vehicle emergency data sets provide useful and critical data elements and the schema set needed to facilitate an efficient emergency response to vehicular emergency incidents

VERDICT Validation Exposure Randomness Deallocation Improper Conditions Taxonomy shows that all cyber attacks can be classified under four improper conditions, namely, validation, exposure, randomness, and deallocation

VLC Media player format is usable for almost all files and formats which is made possible by codes that are already implemented and required for different formats

VP Virtual prototyping is a method in the process of product development which involves CAD and CAE software to validate a design before building up a physical prototype

VR Virtual reality is the representation and simultaneous perception of reality and its physical properties in a real-time computer-generated, interactive virtual environment

V2E Vehicle to environment can position the vehicle in the surrounding environment, but the environment can also sense and position the vehicle with intelligent components

V2H Vehicle to home system makes it possible to draw power from electric vehicles (EV's) large capacity batteries through a distribution board to power a home

V2I Vehicle-to-infrastructure is a communication model that allows vehicles to share information with the components that support a country's highway system

V2R Vehicle-to-roadside is supported by V2I protocol and V2R

V2V Vehicle-to-Vehicle is an automotive technology designed to allow vehicles to talk to each other

V2X Vehicle-to-X is the wireless exchange of critical safety and operational data between vehicles and road infrastructure

VW Volkswagen is the regular brand of Volkswagen AG

W

WAVE Wireless access for the vehicle environment is currently considered as the most promising technology for vehicular networks supporting interoperability and robust safety communications in a vehicular environment

WCCPS Wireless cyber-physical surveillance systems combine low-end sensors with cameras for large-scale ad hoc surveillance in unplanned environments

WEP Wired equivalent privacy is a security protocol, specified in the IEEE Wi-Fi standard, 802.11b, that is designed to provide a WLAN with a level of security and privacy comparable to what is usually expected of a wired LAN

Wi-Fi Wireless Fidelity is a type of wireless network technology used for connecting to the Internet

WPA Wi-Fi protected access is a security standard for users of computing devices equipped with wireless Internet connections, or Wi-Fi

WLAN Wireless local area network is a wireless distribution method which links two or more devices using wireless communication within a limited area which provide a connection to the wider Internet

WNC Wireless network connections allow working independently which need a broadband Internet connection and modem and a wireless router

WSAN Wireless sensor and actuator network is a group of sensors that gather information about their environment and actuators, such as servos or motors that interact with them, whereby all elements communicate wirelessly; interaction can be autonomous or human controlled

WSN Wireless sensor network are spatially distributed autonomous sensors to monitor physical or environmental conditions and to cooperatively pass their data through the network to other locations

WSS Wheel speed sensor is a sender device used for reading the speed of a vehicle's wheel rotation

WVSC Wireless Vehicle Safety Communication helps to overcome some of the limitations of autonomous systems and enhance the overall safety system performance

WWDW Wrong-way driving warning is a new ADAS to prevent wrong-way driving

WYSIWYG What you see is what you get editor or program that allows a developer to see what the end result will look like while the interface or document is being created

X

XaaS X-as-a-Service refers to an approach to provide and consume everything (X) as a service

XML Extensible markup language is a simple, very flexible text format derived from SGML (ISO 8879)

XOR Exclusive OR is a logical operation that outputs true only when inputs differ; one is true the other is false

XP Extreme programming is a method which places the task of solving a programming task at the forefront of software development and thereby makes a formalized approach less important

Y

YRS Yaw rate sensor is a gyroscopic device that measures a vehicle's angular velocity around its vertical axis, whereby the angle between vehicle's heading and vehicle's actual movement direction is called slip angle which is related to the yaw rate

Z

ZF Gear Factory is a German car parts maker headquartered in Friedrichshafen

Index

A

Access

- direct physical, 330
- misdirection, 330
- progressive, 330
- unauthorized, 88, 267, 270, 297, 314, 325, 338, 496, 582

Accuracy

- 67, 102, 104, 158, 218, 254, 255, 257, 518, 544, 550
- Activities, 8, 23, 29, 31, 33, 45, 46, 48, 54, 60, 62, 65, 69, 72–76, 97, 132, 173, 174, 176, 178, 185–187, 202, 205, 217, 228, 237, 248, 274, 278, 280, 288, 300, 304, 305, 319, 340, 341, 343, 345, 350, 381, 394, 417, 421, 423, 430, 432, 439, 452, 457, 495, 497, 500, 518, 549, 550, 571, 590

Actros

30

Actuator

- capabilities, 176
- scheduling, 176

Ada

Adaptive

- cruise control (ACC), 96, 100, 142, 152, 157–159, 162, 253, 257, 302, 332, 514, 516, 550, 569, 586

high beam assist (AHA), 96

Address spoofing

351, 371

Ad hoc

- computer network, 223
- mode, 146, 344

Advanced

- driver assistance system (ADAS), 9, 10, 47, 95, 149–151, 166, 498, 513, 514, 516–518, 582

research and technology for embedded intelligence systems (ARTEMIS)

172

research projects agency network (ARPANET)

206

simulation and control engineering tool (ASCET)

131

Advanced driver assistance systems (ADAS)

10, 13, 25, 27, 83, 96, 133, 147, 150–155

Advantage

- competitive, 68, 74, 128, 379, 560
- financial, 62
- time, 62

AEMT telematics standard

219, 220

Aftermarket

in-vehicle connectivity solutions, 228

Aftersales service

70, 129

Agent

theory, 277

Agile

- manifesto, 128, 379, 384, 433
- modeling, 129
- software development (ASD), 127–129, 166, 379, 384

unified process (AUP), 129

Agility

3, 128, 379

Airbag control system (ACS)

93, 100, 323

Airbiquity

240, 245

Aircraft

24, 28, 189, 217, 304, 572

Alcohol blood level

26

Algorithm

4, 28, 50, 52, 54, 69, 84, 106, 113, 130, 131, 152, 174, 175, 178, 179, 183, 205, 206, 243, 246, 254, 257, 267, 275–279, 282, 287, 288, 291, 310, 320, 322, 337, 339, 341, 351,

- 352, 354, 355, 385, 413–415, 462, 466, 472, 478, 492, 495, 502, 513, 516, 524, 530, 531, 534, 538–544, 563, 566, 570
- Analog-digital-converter (ADC), 104, 107–109, 145, 165, 173
- Analytic shapes, 52
- Anatomy of automotive hack, 366
- Android
- auto, 241
 - car, 379, 397
 - software development kit, 379, 395
 - technology stack, 395
- Anomaly
- detection, 288, 289, 300, 369
 - detection techniques, 289, 300
 - intrusion detection, 279, 282, 291, 340–343, 467, 505
- ANSI/EIA 632 model, 72, 73
- Antilock braking system (ABS), 92, 93, 100, 134, 320, 550
- App
- development, 379, 387, 390, 398, 432
 - market, 379, 388
 - mobility, 379
 - store, 233, 334, 383, 384, 409
- Apple
- CarPlay®, 101, 241, 397
 - iCloud ecosystem, 379, 390
 - iOS, 379, 383, 389, 393–395, 397, 404, 405, 409, 415, 432
 - APIs, 392
 - architecture, 390
 - platform, 390
 - simulator, 394
 - iPhone®, 101, 233
 - iWatch, 389
 - MAC OS, 337, 379, 389, 391, 393
 - mobile devices, 389
 - project Titan, 34, 570
- Application programming interface (API), 149, 151, 219, 237, 240, 315, 379, 390, 392, 394, 395, 398, 412–415, 565
- Architecture of a battery management system, 450
- Arena, 35
- Arterial roads, 516
- Artificial
- intelligence (AI), 4, 210, 241, 246, 255, 265, 272, 273, 369, 517, 526, 550, 570, 586, 590
 - neural network (ANN), 273, 274, 500, 506
 - neural network based IDS, 500–503
- Ashok Leyland, 16, 49, 50, 56, 57, 64
- Assembly analysis, 55
- Assistance function, 28, 33, 34, 127, 248, 254, 513, 514, 517, 518, 550, 552, 569
- Association for computing machinery (ACM), 183
- Associative reasoning, 282
- Asymmetric cryptographic algorithms, 355
- Attack
- anatomy, 265
 - on the billing system, 478
 - graphs, 292, 293
 - scenarios, 265, 268, 269, 331, 349, 367, 454, 463, 506
 - surface, 265, 330–338, 340, 346, 362, 364, 365, 370, 439, 453–456, 503, 514, 566, 570, 582, 587
 - surface intrusion points, 332, 333, 335, 336
 - taxonomy, 326, 370
 - value chains, 307–309, 370
 - value chains in vehicles, 308
 - vector, operational impact, defense, information impact, and target taxonomy (AVOIDIT), 328, 329
 - vulnerability, 566
- Audi, 30, 33, 115, 129, 257, 517, 560, 562
- Augmented reality (AR), 58, 77, 241
- Authentication mechanism, 353, 453, 456, 479
- Authorization, 453
- Auto-code generation, 131, 544
- Automaker, 1–3, 5–7, 9, 19, 26, 30, 45, 46, 48, 88, 89, 94, 96, 115, 126, 132, 142, 145, 147–149, 163, 181, 214, 222, 224–226, 228–230, 232, 233, 237, 238, 240, 245–248, 252, 254, 256, 257, 267, 269, 281, 302, 304, 307, 313–316, 322, 335, 336, 345, 348, 362, 365–368, 490, 589
- Automated
- driving, 28, 147, 251, 256, 259, 556, 558, 560, 571
 - valet parking, 10, 452, 493–496, 505, 582
- Automatic
- climate control (ACC), 83, 94, 96, 126
 - cooling (AC), 94
 - headlamps (AH), 94
 - parking (AP), 152, 159, 160, 164, 216, 517, 569, 586
 - test sequences, 133
 - vehicle parking (AVP), 96
 - wipers (AW), 94

- Automotive
aftermarket, 17, 19
attack surfaces and vulnerability, 265
cloud service system (ACSS), 230, 259
cyber-physical systems, 9, 85, 86, 171,
 173–197, 199–206, 209, 257, 265,
 267, 268, 270, 271, 280–283, 294,
 295, 300, 308, 309, 316–324, 326,
 328, 332, 337, 345, 346, 348, 354,
 360, 362, 382
cybersecurity, 7, 9, 265–346, 348–371, 582,
 587
data and time triggered framework (ADTF),
 565
development process, 45–62, 77
E/E systems, 9, 26, 28, 45, 86, 95, 110, 111,
 127, 132
electronics, 10, 83, 86–109, 120, 151, 165,
 221, 248, 269, 307, 405, 582, 590
engineering, 181, 581, 583, 590
industry, 1, 3, 7, 9, 13–38, 45, 83, 88, 111,
 116, 120, 126, 129, 142, 149, 163,
 214, 222, 225, 232, 234, 241, 245,
 247, 248, 252, 256, 268, 269, 302,
 303, 305, 308, 315, 319, 336, 360,
 380, 382, 432, 496, 513, 559, 581,
 583, 588–590
IT, 302–307, 337, 380, 387
manufacturers, 14, 27, 129, 229, 502
mechatronic, 9
mega trends, 13, 19, 37, 38
night vision, 516
open source architecture (AUTOSAR), 127,
 142–145, 147, 166, 311–314, 514,
 566, 582
open source architecture adaptive platform,
 83, 127, 131, 147
protection mechanisms, 26, 359
safety, 120, 122, 132, 360, 566
safety integrity level (ASIL), 120, 122, 123,
 132, 360, 361, 371, 571
security, 338
software development, 127, 128, 132, 138,
 149, 247, 561
software engineering, 83, 126, 590
Spice®, 127, 129
suppliers, 3, 32
transformation, 9
AUTomotive Open Source Architecture
(AUTOSAR)
adaptive platform, 83, 147
application, 145, 146, 148, 311, 314
network, 147
open standardized software architecture for
 automotive ECU, 311
operating system, 145, 147
Autonomous
car, 25, 28, 463, 514, 517, 549, 552, 554,
 557, 558, 566, 569, 571, 572, 583,
 587–589
delivery robot, 37
driving (AD), 26–28, 105, 171, 302, 480,
 493, 513, 581
emergency braking (AEB), 141, 152, 160
mobility, 552
vehicle, 4–6, 28, 34, 95, 101, 167, 171, 214,
 241–247, 254, 255, 259, 505, 513,
 552, 557, 560, 570, 589
Autopilot function, 557
Availability, integrity, and confidentiality (AIC)
 triad, 331
Avert, 266, 439
AVOIDIT architecture, 328, 329
- B**
- Backend system, 25, 432, 439, 446, 453, 456
Background subtraction
 algorithm, 534, 535
 model, 536, 537
Backoffice, 240, 443
Backpropagation algorithm, 4, 281
Backscatter modulation, 212
Bandwidth, 87, 103, 118, 119, 147, 173, 230,
 351, 499, 561, 564, 569, 586
Bangalore, 20, 405, 465, 468
Basic
 functions, 53, 64, 143, 227, 311, 421
 reproduction rate, 284
 spline function, 51
Battery
 capacity, 220, 583
 management system, 22, 139, 449
 prices, 22, 38
Behavior detection, 296
Bellman-ford algorithm, 175, 257
Benchmark, 46, 155, 286, 390
Bernstein polynomial, 50
Bézier
 curves, 53
 net, 51
 point, 51
 representation, 50, 51
 surface, 50, 51, 77
Bharat-Benz, 16, 30, 57
Big data analytics (BDA), 1, 4, 239, 463

- Bills of materials (BOMs), 61
- Black
 hat Asia security conference, 324
 hole attack, 499
- Blending of technology, 67
- Blind spot
 detection (BSD), 152, 153, 158–160, 162, 514, 569, 586
 sensor, 521
- Block algebra equations, 192
- Blood alcohol concentration (BAC), 95
- Bluetooth, 101, 215, 223, 227, 237, 238, 249, 252, 253, 306, 331–333, 365, 366, 395, 397
- BMW
 connectedDrive store, 171, 249, 260
 DriveNow, 440, 452, 455
- Body
 control module (BCM), 89
 control unit (BCUnit), 320
 electronics, 86, 89–92, 118, 119, 139, 165
 network, 88
- Boeing, 55, 70
- Bogus messaging, 499
- Bosch
 car multimedia, 248
- Bot algorithms, 478
- Boundary control points, 54
- Brake
 assistant, 514
 by-wire, 116
 control unit (BCU), 320
- Brightness patterns, 538, 539
- Broadcasting
 serial network protocol, 115
- Brushless, asynchrony machine, 449
- Brute force attack (BFA), 339
- B-spline
 surface, 51, 52
- Bus
 system, 83, 110–112, 114–121, 126, 165, 166, 230, 310, 313, 322, 330, 501, 503, 552, 561, 569, 582, 586
- Business
 as-a-service (BaaS), 238
 case, 3, 47
 model, 1, 5, 7, 8, 10, 19, 24, 35, 147, 149, 171, 183, 214, 220, 225–228, 231–233, 238, 241, 259, 269, 314, 315, 439, 441, 461, 462, 464, 465, 488, 491, 505, 550, 581, 582, 586, 589
 needs and requirements (BNR), 59
- to-business (B2B), 7, 226, 234
to-business-to-customer (B2B2C), 7
- BYD, 33
- Byzantine
 failure, 358
 model, 358, 371
- C**
- C, 103, 130, 131, 184, 276, 337, 379, 391, 394, 398
- C++, 184, 276, 337, 379, 393, 394, 404, 544
- Cab
 aggregator, 461, 472, BNF–467
 services, 409, 461, 463, 465, 466
- Camera
 based sensor, 2, 154, 156, 159, 160, 210, 243, 260, 489, 495, 497, 517, 525, 526, 552, 556, 561, 564, 566, 569
 rigs, 34
- Canny edge detection algorithm, 530
- Car
 as-a-platform (CaaP), 334
 density, 17, 18, 20
 E/E system, 9, 26, 28
 hacking, 28, 237, 266, 362, 371, 587
 hacking village, 26
 hailing and ride sharing, 461
 information technology (IT), 6, 9, 34, 126, 215, 225, 230, 234, 238, 240, 267, 306, 455
 insurance companies, 19, 28, 231
 insurance policy, 19
 in-the cloud, 171, 238–241, 243, 259, 550
 ownership, 13, 17, 20, 23, 440, 586
 park operators, 486
 play, 38, 231, 387, 397, 435, 504, 505, 550, 552, 571, 572
 pooling, 405
 rental, 24, 439, 441, 448, 454, 455
 ride, 24, 404, 461, 486, 503, 571
 sharing, 24, 28, 220, 379, 404, 405, 440, 441, 452, 454, 462, 477, 485, 496, 505, 571
 sharing activities, 439, 452
 sharing model, 220, 379
 sharing services, 439–441, 452, 473
 to-backend (C2B), 25, 439, 453
 to-car (C2C), 25, 201, 550, 566
 to cloud communication, 570, 587
 to-go business model, 439, 441
 toGo concept, 441, 454
 to-infrastructure (C2I), 550

- Cartesian product, 51
CeBit, 35, 247
Center
for automotive embedded systems security (CAESS), 364
experimental security analysis of a modern automobile, 364
of mass, 542
Central
control points, 50
control unit (CCU), 321
gateway, 323, 502
Centroid in a plane, 542
Change
in brand loyalty, 163
in customer demand, 163
management, 64, 65, 67
in mobility, 163
Characteristic polyhedron, 51
Charge De Move (ChaDeMo), 22
Charging
infrastructure, 22, 447, 448, 453, 454, 585, 589
station, 448, 449, 455
Chassis
electronics, 86, 92, 93, 165
system, 92
Check-out phase, 444, 445
China, 3, 14, 16, 17, 20, 27, 33, 86, 388, 441, 462, 464, 477, 478, 584, 586
Choreo platform, 240
Classification
of common cybersecurity risks, 272
Closed-loop control system
block diagram, 191
symbols, 191
Cloud
access, 248, 269
based detection, 296
based infrastructure, 234
based security, 357
based systems, 239
based technologies, 239, 240
computing, 1, 178, 179, 238, 266, 357–360, 371
networks, 357
services, 6, 238, 239, 305, 309, 412
COBOL
code, 382
Cocoa Touch Layer, 391
Code portability, 181
Collaborated
and parallel work, 56
work, 45, 67
Collaboration, communication, control (3C), 85, 177
Collision
avoidance (pre-crash) system (CAS), 517
free motion, 96
warning system (CWS), 517, 521
Color model, 526–528
Combination of parking and charging, 496
Combined charging system (CCS), 22
Combustion
engine, 19, 21, 23, 34, 38, 97, 163, 448, 449, 456, 584
engine vehicles, 19, 22, 97
Comfort electronics (CE), 87, 94, 165
Commercial
of the shelf systems (COTS), 380, 433
vehicle, 2, 7, 13, 16, 17, 30, 141, 474, 581, 584
Common
engineering client, 381
vulnerabilities and exposures database (CVE), 280
Communication and entertainment systems, 86
Competing charging standards, 22
Complementary software (CSW), 145, 313
Complex modeling and simulation, 53
Complexity, 10, 28, 45, 47, 54, 56, 59, 83–85, 88, 110, 111, 113, 116, 126–130, 132, 133, 136, 137, 142, 143, 148, 152, 175, 177, 181–184, 265, 266, 269, 311, 313, 315, 322, 331, 350, 385, 391, 433, 461, 492, 525, 549, 550, 557, 581
Component
analysis, 181, 545
supplier, 46
test, 133, 140
vulnerability, 338
Compound anomaly detection, 342
Compromised
key attacks, 317, 370
privacy, 309, 310, 340
Computer
aided design (CAD), 45, 49–51, 54–57, 64, 65, 67–69, 77, 382, 581
aided engineering (CAE), 55, 57, 65, 77, 581
aided manufacturing (CAM), 65, 67
assisted perception, 58
emergency readiness team (CERT)
taxonomy, 323, 328
format, 1, 45, 101, 130, 145, 146, 219, 256, 313, 320, 334, 345
graphics, 50, 51, 54, 57, 131, 147, 326, 526, 527
modeling and simulation, 66
society, 59, 183

- Computer aided design (CAD), 45, 49–51, 54–57, 64, 65, 67–69, 77, 382, 581
- Computer aided engineering (CAE), 55, 57, 65, 77, 581
- Computer aided manufacturing (CAM), 65, 67
- Computer aided three-dimensional interactive application (CATIA), 49, 50, 64, 382
- Computerized motor management, 35
- Computing performance, 87 technology, 266
- Conceptual ideas, 69
- Conceptualization phase, 45, 46, 73, 77
- Concierge services, 251
- Concurrent engineering (CE), 55, 56
- Conic sections, 52
- Connected car, 5, 7–10, 25, 38, 101, 105, 166, 171–257, 267, 269, 334, 335, 379, 485, 490, 498–500, 503, 504, 506, 550, 569, 581, 582, 585 platform, 233, 241, 259 reference platform, 171, 237, 259 services, 7, 227, 233, 435, 490 car-as-a-digital-platform (CCaaDP), 335 drive, 488, 582 home, 250 parking, 384, 485–507, 582 parking app, 488, 504 services, 225, 232, 310, 384 trucks, 245 world, 8, 363
- Connectivity, 2, 4, 5, 7–10, 13, 19, 25, 29, 85, 87, 95, 101, 126, 136, 147, 208, 225–228, 231–233, 235–241, 247, 252, 257, 268, 269, 282, 305, 306, 330, 336, 355, 365, 367, 395, 439, 446, 453, 455, 456, 485, 496, 503, 550, 570, 581–583, 585, 587
- Constraint, 1, 37, 49, 73, 99, 110, 135, 150, 183, 196, 201, 204, 234, 241, 283, 290, 305, 317, 324, 340, 341, 343, 395, 412, 506, 539, 563
- Conti, 31, 493, 551
- Continental, 3, 153, 154, 159, 256, 517
- Control, 6, 13, 45, 83, 171, 265, 380, 449, 473, 492, 514, 581 derivative, 193–197, 199, 200, 258 integral, 193–197, 258 point, 50, 51, 53, 54 point matrix, 53 proportional, 193–196, 258 proportional integral derivative (PID), 196, 197, 199, 200
- system, 65, 94, 96, 100, 107, 118, 127, 132, 176, 177, 189–197, 199–201, 210, 216, 253, 258, 266, 270, 283, 323, 332, 363, 368, 516, 543
- Controllability, 122, 200, 360
- Controller area network (CAN) bus, 89, 114–121, 237, 265 firewall, 115 output signal, 115 data message structure, 117
- Cookies, 249
- Core animation, 392 graphics, 391 OS layer, 391 process, 59, 390 service layer, 391, 392
- CoRide App data flow diagram, 408 use case diagram, 406
- Correlation process, 293
- Cost driver, 22 efficiency, 115, 126, 383 management, 46 pressures, 19, 49 reduction, 46, 63 structure, 34, 450, 451, 456 of conventional vehicles, 451 of electric vehicles, 451
- Covariance matrix, 342
- C-program, 181, 398
- Cradle[®] software tool, 186, 187
- Crash avoidance system, 302
- Crime incidents in ridesharing, 461, 476
- Crosswind stabilization (CS), 97
- Customer analysis, 46 identification, 46, 221 mobility support, 220 profile, 111, 221 relationship management (CRM), 238 requirements, 46, 62, 379
- Cutting-edge innovation, 1, 2, 6, 7, 28, 589 technology, 4, 28, 273
- Cyber attack, 25, 228, 266, 268, 271, 293, 304, 307, 309, 328, 329, 331, 340, 368, 439, 453–456, 461, 462, 478, 496, 497, 567, 569, 570, 572, 582 surfaces, 331, 332, 453, 454, 582 threats, 496
- components, 172, 177, 206, 257, 271, 295, 318

- crime, 266, 281, 453, 462
criminal attack, xiii–xv, 19, 266, 267, 278, 279, 282, 283, 288, 289, 295, 297, 318, 327, 328, 330, 332, 333, 358
characteristics, 279, 283, 293, 327
classification, 277
taxonomy, 326, 370
physical system (CPS), 9, 10, 29, 85, 86, 125, 164, 171, 173–197, 199–206, 209, 257, 258, 265–269, 271, 280–283, 294, 295, 300, 308, 309, 316–324, 326, 328, 332, 336–338, 345, 354, 360, 362, 582
architecture, 316
concept map, 179, 180
design recommendations, 180, 181, 183, 184, 258
requirements, 176, 184–188
risk, 303, 566
security, 5, 35, 241, 272, 280, 305, 311, 325, 328, 339, 368, 369, 455, 462, 478, 485, 496, 503, 504, 506, 514, 572
security approach, 267, 311, 318, 463
security audit, 294
security risk, 86, 270–272, 293, 294, 328, 330, 332, 369, 454, 478
security solutions, 303, 309, 317, 338, 496, 514
space, 266, 274, 282
threats, 303, 485, 496, 503, 514, 569
weapon, 318
- D**
- Daimler
AG, 46, 441
Benz, 46, 525
Bharat Benz, 16, 30
Car2Go, 439–442, 444–446, 448, 449, 451–457
Damage of functioning, 265, 266
Dashboard
modifications, 305
Data
centric approach, 174, 186, 302
cleanup, 204
distribution service (DDS), 565
flow diagram (DFD), 408
glove, 57
link layer, 115
security, 56, 202, 230, 243, 266, 270, 357–360, 504
tracker, 57
tsunami, 4
- Database
map, 409, 412
system, 62, 382
transactional, 409
DC-DC converter, 450
DDOS attack, 343, 478
Decentralized learning algorithm, 291
Decision-making, 46, 68, 71, 157, 277, 287, 290
Decryption, 352, 353, 355, 356
Deep
belief networks (DBN), 281
learning (DL), 265, 281
neural network (DNN), 265, 272, 281, 504
Defective
operation, 270, 295
Defense advanced research projects agency (DARPA), 365
Deliverables, 47, 48, 65, 66, 148, 149, 315
Delphi, 31, 247, 517
Denial-of-service (DOS)
attack, 340, 354
Denso, 3, 31, 159–161
Dependency graph
method, 293, 294, 301
Depth information, 556
Design
control, 66, 176
release management, 65
Detect, 62, 65, 96, 97, 99, 100, 103, 104, 108, 112, 124, 152, 154, 155, 157, 160, 206, 214, 251, 266, 270, 278, 287, 289, 291, 296, 300, 301, 303, 326, 348, 352, 355, 463, 474, 478, 490, 497, 499, 500, 503, 516, 517, 522–524, 530, 533, 534, 536, 537, 542, 544, 546
Detection of moving objects, 533, 535, 536, 543
Detector, 57, 190, 300
Development, 1, 45–78, 83, 172, 266, 379, 384, 466, 488, 513, 581
Devices
home, 179
mobile, 163, 179, 207, 243, 244, 247, 249, 359, 389
personal, 179
security, 179
wearable, 179

- Diagnostic
 instruments, 35
 tests, 133
- Didi, 462–466
- Digital
 analog-converter (DAC), 173
 archive, 69
 ecosystem, 583, 586
 factory, 29
 information, 58, 97, 172
 manufacturing, 66
 maps for fully autonomous driving, 254–257
 mock-up (DMU), 54, 55, 57
 model, 58
 platform, 335, 586
 prototyping (DP), 54
 solution, 231, 233
 transformation in vehicles, 222, 303, 304
- Digitization, 1, 2, 58, 163, 245, 267, 269
- Dijkstra algorithm, 175, 257
- Direct physical access, 330
- Discontinuities in flow, 539
- Disrupt
 communication, 265, 266
 impact, 28, 590
- Distraction, 231, 473, 516, 549
- Distributed denial of service (DDOS), 343, 454, 499
- Disturbance, 103, 189–191, 267, 325
- Dongfeng, 16, 31
- Door
 control unit (DUC), 321
 locking mechanism, 84
- Drive
 by-wire, 116, 332
 slip control, 126
 train, 141, 450, 570
- Driver
 alcohol detection system for safety (DADSS), 95
 assistance electronic, 83, 87, 94–98
 assistance system (DAS), 5, 28, 94, 126, 153
 drowsiness detection (3D), 97, 514, 516
 error, 150, 516
 license, 23, 411, 477, 549, 559
- Driving
 conditions, 149
 phase, 443, 445
 profile, 250
 worthiness wizard, 226
- Drones, 2, 6, 570
- Drowsiness, 97, 514, 516
- dSpace
 I/O boards, 139
 SCALEXIO, 140
 simulator, 138–140
- Dynamic
 pricing, 465
 systems development method (DSDM), 129
- Dysfunctional sensor processing, 309
- E**
- E-
- call, 8, 251
 - CarTec, 23
 - mobility, 3, 21, 23, 504, 585, 588
 - plate recognition, 4
- Eavesdropping, 317, 325, 331, 351, 352, 359, 371
- Economy of scale, 63, 74
- Ecosystem, 8, 34, 35, 236, 269, 302–304, 350, 379, 390, 491, 550, 583, 585
- Edge
 detection, 513, 528, 530, 533, 534, 545, 546
 detection algorithm, 546
- E/E
 architecture, 110–114, 126, 132, 142, 143, 311, 502, 561, 562
 multifunctional components, 114
 systems, 9, 26, 28, 45, 86, 89, 95, 109–111, 127, 132
- Effective reproductive rate, 285
- Electric
 brake distribution, 93
 car, 21, 22, 24, 34, 441, 447–450, 452, 453, 455, 496, 584, 585, 589
 components, 450
 drive, 2, 21, 23, 139, 140, 583, 584
 drive train technology, 21, 584
 power management, 449
 power train, 5, 21, 47, 139, 449, 585
 propulsion, 20
 vehicle, 13, 19, 20, 33, 34, 97, 163, 439, 447–451, 495, 504, 585, 589
 vehicle models, 23
 vehicle warning sound (EVWS), 97
- Electromagnetic
 compatibility (EMC), 118, 230, 336
 discharge (EMD), 118
 interference (EMI), 118
 signal, 57

- Electronic
brake control module (EBCM), 320
control unit (ECU), 92, 98, 99, 111, 113,
116, 117, 126, 127, 129, 131–140,
144, 145, 160, 281, 305, 308, 312,
319–323, 333, 362, 453, 561, 563
horizon, 255, 256
license plate (ELP), 307
meeting tools, 68
mirrors, 83
power steering (EPS), 100
seat adjustment with memory (ESAM), 94
solid-state nonvolatile storage medium, 98
stability control (ESC), 93, 100, 127, 154
stability program (ESP), 93, 332, 550
E-mail attachment, 297
Embedded computing
power, 28
systems (ECS), 178, 179
Emergency
assist (EA), 97
phase, 445
warning system, 215
Emission, 2, 3, 13, 19–23, 71, 117, 163, 225,
246, 271, 448, 584
Encryption algorithm, 352
Engine control module (ECM), 98, 99, 321,
323
Engineering staff, 45
Enterprise resource planning (ERP), 62, 66, 67,
238, 382
Entertainment
features, 226, 334
online, 251
Entity-to-entity-oriented IoT applications, 208
Entry media and navigation system (EMNS),
149, 316
Epidemic theory, 265, 272, 284–287
Erasable programmable read-only memory
(EPROM), 98
Error
configuration, 292
covariance, 283, 284
detector, 190, 191
magnitude, 540
ERTICO-ITS Europe, 256
Essential unified process (EssUP), 129
Estimated time to arrival (ETA), 466, 474
Ethernet, 111, 116, 119, 126, 128, 147, 230,
237, 336, 503, 561, 563
Ethical
difficulties, 559, 560
issues, 28, 569, 586
questions, 246
Europe, 14, 16, 17, 20–23, 71, 86, 123, 236,
249, 252, 361, 440, 441, 486, 558
European union (EU), 172, 231, 256, 441
E-vehicles, 163, 228
Everything-as-a-service (EaaS), 238
Evolutionary
algorithm, 113, 275, 276, 279
phase, 45, 73
Exchange standard of product model data
(STEP), 66
Exclusive OR operation (XOR), 352, 353
Exfiltration of data, 270
Exhaust gas recirculation (EGR), 99
Extended markup language (XML), 130, 187,
219, 395, 404, 413, 416–418, 420,
421, 423, 427
Extreme programming (XP), 129
- F**
- Facebook®
client, 252
- Failure
mode and effects analysis (FMEA), 123,
124, 361
mode and effects and criticality analysis
(FMECA), 123, 124, 166, 361, 362
routing unit (FRU), 140, 141
- Fake accounts, 462, 478
- False
negative, 205, 317
positive, 205, 279, 289, 300, 317, 340, 498,
503
- Fatal
road accidents, 27
traffic accidents, 26
- Fault
detection and localization, 301
diagnostics, 301
insertion unit (FIU), 135, 139–141
tolerance, 150, 174
- Feature-driven development (FDD), 129
- Feedback loop, 176, 190, 192
- Fiat Chrysler, 34, 368
- Finite element model (FEM), 50, 67
- First come first serve principle, 405, 406
- 5G-based services, 241
- Fixed grid, 539
- Flash memory chip, 98
- Flat network architecture, 174, 257
- Fleet
dispatching, 202
routing, 202
tracking, 202

- Flexibility, 3, 19, 24, 63, 139, 213, 269, 279, 304, 440, 552, 561, 569, 586
- FlexRay, 111, 115, 118, 119, 126, 128, 310, 336
- Flooding, 173, 175, 272, 280
- Ford, 34, 145, 175, 239, 257, 266, 334, 335, 365, 367
- Forecast, 3, 25, 63, 163, 201, 208, 224, 239, 241, 589
- Forward-looking cameras, 156
- 4G-based services, 241
- Framework for automotive cybersecurity best practices (FACBP), 304
- Fraud, 246, 291, 330, 461, 462, 474, 478
- Free
- floating car sharing, 440
 - form entities, 52
 - ways, 516
- Freightliner, 30
- Frugal engineering, 583
- Fuel
- cells, 585
 - prices, 14
- Functional
- audit, 125, 362
 - capability, 55
 - needs, 59, 185
 - performance test, 348
 - safety, 10, 29, 83, 95, 120–127, 132, 166, 265, 350, 360–362, 371, 386, 514, 563, 566–570, 572, 583, 588, 590
 - measures, 121
 - and security, 303, 350–362
 - security testing, 265, 337
- Functionality
- undisturbed, 126
- Fuso, 30
- Fuzzy
- and penetration testing, 265, 337
 - sets, 276, 279
- G**
- Game
- controller, 392
 - theory, 175, 265, 272, 287, 289, 300, 311, 369
- Gateway, 113, 116, 167, 234, 235, 237, 240, 259, 281, 323, 333, 340, 350, 502, 556
- Gaussian Markov random field (GMRF), 301
- Gaussian random variables, 301
- General motors (GM), 3, 30, 71, 247, 335
- Generalization notation, 185
- Generic cyber-attacks life cycle, 271
- Genetic algorithm (GA), 277, 279
- GENEVI
- alliance, 83, 147, 171, 247–249, 260, 314
- Geo location, 446
- German
- academy for science and engineering (acatech), 172
 - automotive trust report (DAT), 26
 - Federal Ministry of Education and Research (BMBF), 153, 172
- Germany, 13, 14, 17, 19, 21, 26, 33, 203, 230, 247, 252, 382, 440, 441, 486, 558–561, 585
- Global
- footprint, 30
 - navigation satellite system (GNSS), 217, 218, 222, 237
 - positioning system (GPS), 96, 215, 217, 218, 236, 252, 254, 255, 321, 362, 412, 443, 446, 454–456, 462, 465, 475, 506, 552
 - sales, 3
 - vehicle production, 3, 14
- Global automotive market, 13, 14, 30, 37
- Global positioning system (GPS), 215, 217, 218, 236, 252, 254, 255, 321, 362, 412, 443, 446, 454–456, 462, 465, 475, 506, 552
- Global System for Mobile Communications (GSM) connection, 25, 395, 439, 446
- Go/kill decisions, 49
- Google
- autonomous car, 25
 - local search, 252, 253
 - maps, 224, 249, 409, 412, 415, 417
 - street view, 34, 252
- Government
- policies, 461, 477
 - policies for ridesharing, 461, 477
- GrabTaxi, 463
- Graph theory, 265, 272, 291
- Graphic user interface (GUI), 390, 395, 404, 417, 420, 423
- Gyroscopic device, 100
- H**
- Hacked and compromised accounts, 478
- Hackers, 330, 338, 362, 454
- Hacking
- user accounts, 454

- Hand
 free cell phone interface, 215
 over, 29
- Hard-real time system (HRTS), 119, 188
- Hardware-in-the-loop (HIL)
 platform, 137
 test, 83, 133–142, 166, 582
- Hash functions for message authentication code (HMAC), 324
- Hazard, 97, 121, 152, 153, 157, 215–217, 231, 235, 246, 249, 256, 268, 269, 360, 365
- Head mounted display (HMD), 57
- Health-check phase, 445
- Heating, ventilation, air control (HVAC), 87, 94, 236
- Herd immunity, 285
- HERE
 digital maps for fully autonomous driving, 254–257
 hazard warnings, 257
 on-street parking, 257
 real-time traffic, 256
 road signs, 255, 257
- Heterogeneity in CPS design, 181
- Heterogeneous traffic, 29
- Heuristic based detection, 296
- Hewlett-Packard (HP), 35, 584
- High
 definition maps, 550
 fidelity mapping, 156
 precision maps, 254
 speed Ethernet bus system, 561
 tech companies, 7, 28, 571, 585
 throughput, 204, 561
- High-definition (HD) maps, 29, 241, 253, 255, 556
- Highly automated driving (HAD), 28, 147, 251, 256, 259, 558, 571
- Hill descent control (HDC), 93, 97
- Holistic
 approach to security, 355
 cybersecurity solutions, 309
 perspective on security, 272
 security approach, 310
- Homogeneous
 control polygon vertices, 53
 polygonal control vertices, 52
- Honda, 71
- Horizon 2020 program, 172
- Horns optical flow algorithm, 538–542
- Hough transform, 531, 545
- Huawei, 388
- Human-machine interface (HMI), 34, 35, 151, 178, 227, 258, 432, 442, 506, 513
- Hypertext markup language 5 (HTML5), 225, 230, 259, 395
- Hyundai, 16, 30, 468, 585
- I**
- IBM
 DB2 database, 383
 WebSphere application server, 383
- Identifiability, 200
- Identity
 and access management (IAM), 239
 theft, 462
- IEC 61508, 120, 122, 127, 132, 360, 361
- IEC 61508-1:2010, 122
- IEC 61508-2:2010, 122, 123
- IEC 61508-3:2010, 122
- IEEE 802.11i wireless local area network standard (WLAN), 215, 319
- IEEE 802.11 wireless network standards, 319
- Image
 analysis, 523, 525, 526, 544, 583
 detection, 513, 526, 528, 530, 533, 534, 545, 546
 processing, 28, 95, 513, 525, 526, 528, 531, 533–535, 542, 544–546, 552, 569, 583, 586
- Impact analysis, 65
- Impairment test, 348
- In
 build camera, 489
 built functions, 542, 545
 car wifi, 237, 241, 367, 471
- Incentive points, 405–407
- Incubator, 35
- Independent repair shops, 19, 365
- India, 14, 16, 17, 382, 388, 462, 465, 471, 472, 475, 477, 478, 583, 584, 586
- Information
 and communication technology (ICT), 5, 172, 177, 183, 187, 316, 356, 362
 imperfect, 290, 291
 perfect, 290
 technology (IT), 5, 9, 29, 34, 58, 67, 126, 215, 225, 230, 232, 234, 238, 240, 249, 251, 266–268, 281, 302–307, 316–324, 326, 328, 337, 349, 379–383, 387, 405, 431, 466, 485, 486, 550, 560, 582, 585, 590
 technology landscape, 381, 382, 387

- Information technology (IT)
 cloud, 238
 landscape, 381, 382, 387
- Infotainment
 components, 83, 171
 technology, 9, 171, 214, 234, 305, 316, 379, 397
- Infrared camera, 156
- Infrastructure
 as-a-service (IaaS), 238, 239, 360
 mode, 344
- Initial checking, 535, 536
- Injecting malware, 478
- Innovation
 cutting-edge, 1, 2, 6
 evolutionary, 3
 revolutionary, 3
- Insiders, 288, 330, 358
- Institute of electrical and electronic engineers (IEEE), 9, 59, 72, 76, 95, 101, 151, 183, 305, 319, 344, 351
- Insurance, 19, 24, 28, 203, 214, 220, 231, 236, 247, 302, 440, 441, 476, 477, 479, 560, 569, 585, 586
- Integrated
 analysis, 48
 development environment (IDE), 379, 382, 383, 393–395, 413
 telematics system, 252
- Integration
 of car and smart home, 384
 test, 133, 139
- Integrity check value (ICV), 352
- Intelligent
 control, 84
 emergency call (IEC), 251
 headlight control (IHC), 152
 parking assistance, 497, 514
 vehicles, 7
- Intercept, 267, 317, 353
- Interest rate, 14
- Interfaces and data exchange, 66
- Interior design, 47, 552, 555
- International
 electrotechnical commission (IEC), 72, 116
 road traffic regulation, 558
 standard organization (ISO), 59, 66, 72, 73, 75, 76, 95, 114–123, 127, 129, 132, 151, 220, 324, 336, 360, 361
- Internet
 and apps, 252
 browser, 253
 control message protocol for IPv6 (ICMPv6), 314
- enabling technologies, 6, 208–210, 241
 of-everything (IoE), 4, 178, 179, 241
 of-things (IoT), 1, 85, 171, 173, 177, 202, 206–210, 214, 222, 239, 241, 288, 303, 488, 491, 504, 582
 of-things roadmap, 209
 protocol v6, 202
 services, 208, 225, 252
- Interoperability
 semantic, 187
 syntactic, 187
- Intersection
 assistance (IA), 97
 support (IS), 152
- Inter-vehicle network, 10, 332, 333
- Intrusion
 detection, 4, 19, 265, 281, 282, 284, 291, 300, 323, 324, 340–343, 345, 347, 348, 350, 455, 463, 503, 569, 582
 detection and prevention systems (IDPS), 25, 278–280, 305, 342, 343, 345, 502
 architecture, 346, 502
 tasks, 345
 types, 343
 detection systems (IDS), 291, 497, 502, 503
 prevention technologies, 343
- In-vehicle
 comfort, 6
 information systems, 86
 infotainment (IVI), 101, 222, 314
 networking, 95, 237
 software, 148, 315
 types and their detection, 94
- Invested cash flow, 46
- iOS
 architecture, 390
 platform, 390
 programming model, 391
- iPod, 34, 333, 390
- ISO 11898-2:2003, 116
- ISO 26262, 10, 95, 120, 122, 127, 132, 151, 360
- ISO/DIS 26262, 120
- ISO/IEC 12207:2008, 75
- ISO/IEC 15288, 72, 75, 76
- ISO/IEC 15288:2002, 72, 75
- ISO/IEC 15504-2, 75
- ISO/IEC 29148 (ISO/EIT 2011), 59
- ISO/IEC JTC1/SC7, 72
- ISO/IEC TR 24748, 76
- ISO/TS 15143-3:2016, 220

J

Japan, 14, 17, 22, 83
Java
Enterprise application, 383

K

Karma, 33
Kinematic systems, 55
Knot vector, 54
Knowledge
flow, 56
meta, 64
sharing, 58

L

LabVIEW, 131
Lane
change assistant (LCA), 153, 162, 253, 516
departure warning (LDW), 98, 253, 516
keeping assistant (LKA), 153, 162, 516,
518, 520, 522
keeping system (LKS), 586
Laplace transform, 198
Large-scale engineering (LSE), 60
Last mile route, 250
Lateral acceleration sensor (LAS), 100
Layer
localization, 255
physical, 116, 118, 119, 230, 356
Legacy functionality, 132
Legal enforcement, 143, 311
Level
of-abstraction, 181
of-detail (LOD), 46, 54
of-innovation (LOI), 60
Liability, 188, 220, 337, 473, 513, 558–560,
569, 586
Lidar, 156, 157, 159, 517, 525, 552
Life cycle
management, 10, 45, 61, 69, 70, 73, 121,
125, 241, 362
process, 59, 72, 75, 76, 311
Light imaging detection and ranging (LiDaR),
95, 156–159, 162, 243, 244, 254,
255, 556
Likelihood, 121, 265, 267, 268, 293
Lines-of-code (LOC), 127, 148, 315, 330, 385,
386, 417
Link layer protocol, 116
LINUX
container (LXC), 398
container resources, 398
foundation, 248

Livelock, 175

LLVM compiler framework, 404
Local
area network (LAN), 95, 215
hazard warning (LHW), 153, 159, 162
interconnect network (LIN), 89, 90, 111,
115, 118, 119, 126, 128, 322

Locating tracking, 351

Logical database requirements, 409

Long-term profitability, 46

Loss of control, 70, 93, 497, 521, 566

Low-power wireless personal area network
(LoWPAN), 202

Lyft, 462–466, 474, 476

M

Machine
learning, 1, 4, 239, 273, 277, 282, 390, 488,
550, 552, 566, 586
to-machine (M2M), 221, 222, 455
to-machine protocol, 448
Magna, 3
Magneti-marelli, 247
Mahindra, 16
Main software (MSW), 144, 312
Maintenance
cost, 29, 89, 112, 132, 175, 217, 343, 382,
383

Mal
function, 521
ware, 306

Malicious
attempts, 265, 267, 294, 295
cyber-criminal attack, 265–267, 278,
281–283, 288, 294–297, 300, 318,
330, 332, 333, 358
files, 296, 297, 333
file types, 297
software intrusion, 265, 266, 278, 291, 295,
332

Management
change, 60, 64, 65, 67, 121, 360
configuration, 65, 66, 71, 123, 132, 134,
292, 337
of dispersed teams, 68
engineering change, 65
inventory, 62
product structure, 54, 64, 65
program, 65, 71, 360, 379

Man-in-the-middle attack (MITA), 272, 309,
317, 339, 497

Manufacturing, 1, 6, 7, 14, 16, 17, 27–31, 33,
34, 46, 47, 54, 56, 62–67, 70–72, 74,
100, 104, 105, 117, 120, 123, 127,

- 129, 143, 145, 147, 178, 218, 229,
232, 302, 304, 311, 313, 333, 334,
379, 383, 384, 388, 389, 397, 431,
432, 440, 448, 463, 502, 517, 557,
559–561, 587, 589
- Market**
 launch, 34, 56, 62, 70, 73, 74, 164, 231, 315,
379, 389, 452, 471
 penetration, 22, 70, 74, 86, 222, 466
 volume, 74, 227, 589
- Marketing and sales**, 55, 62
- Markov**
 model, 121, 301
 random field, 301
- Maruti Suzuki**, 16
- Mass hacking**, 330
- Material requirements planning (MRP)**, 65
- Mathematical model**, 84, 131, 191, 200
- MATLAB®**
 image processing toolbox, 545, 546
 Simulink®, 131, 200, 513, 543, 544, 552,
569
- Maturity**, 45, 47, 49, 55, 57, 73, 74, 86, 171,
229–231, 581
- Mean**
 shift tracking algorithm, 534, 535
 time between failure (MTBF), 123, 361
 time to dangerous failure (MTTF_d), 123
- Measuring software quality**, 146
- Mechatronic**
 product features, 83
 system, 83–86, 100, 133, 134
- Media**
 and content delivery network, 240
 oriented systems transport (MOST), 322
- Medium access control**, 351
- Mega cities**, 584
- Mercedes**
 benz, 33, 115, 252, 253, 257
 COMAND® online, 171, 252–254, 582
 emergency call center, 252
 intelligent drive system, 253
- Message authentication code (MAC)**, 324
- Metropolitan area network (MAN)**, 119
- Microprocessor (μP)**, 99, 147, 244
- Microsoft**
 azure, 239, 504
 embedded platform, 395, 446
 SQL server, 383
 Windows, 239, 318, 337, 389, 390, 404, 447
- Middleware**, 66, 147, 180, 204, 206, 229, 248,
382, 383, 514, 552, 561–566, 569,
583
- Miller and Valasek**
 physical hack, 365
 remote hack, 367
- Mindmap**, 516, 518, 519, 553
- Minimization problem**, 542
- Minimized resource consumption**, 127
- Misdirection access**, 330
- Mission critical**
 automotive components, 266, 342
 ECU components, 308
- Misuse intrusion detection**, 341
- Mitigation of cyber attacks**, 439, 453, 454
- Mobile**
 ad hoc network (MANET), 95, 303, 351
 device-centric cloud computing, 359
 office, 249, 250
- Mobility**, 269
 as-a-service (MaaS), 8, 241
 on demand, 2, 19
 management, 8, 209, 216, 226, 235, 379,
582, 585
 services, 2, 7, 8, 225, 231–233, 239, 256,
267, 269, 384, 589
- Model**
 based development, 45, 127, 130–132, 146
 based software development, 83
 of intent, 288
 in-the-loop testing (MIL), 131, 151
- Modularity of automotive software**, 143, 312
- Modularization**, 10, 45, 63, 64, 110, 569, 586
- Morphological**
 operation, 526, 531, 532, 534, 546, 547
 operator, 546, 547
- Motorola**
 droid, 240
- Multi**
 brand vehicle model line, 45
 dimensional settings, 61
 hop, 107
 lateration, 307
 layer perceptron (MLP), 500
 period usage, 220
 purpose vehicle (MPVs), 13
 tasking, 391, 395
 touch, 391, 395
- N**
- Nash equilibrium**, 290
- National**
 highway traffic safety administration
 (NHTSA), 171, 242, 558
- instruments fault intrusion unit (FIU)**, 135

- instruments hardware in the loop test device, 133, 134
science foundation (NSF), 172
vulnerability database (NVD), 345
National Instruments (NI), 116, 134, 137
Navigation
satellite, 217, 218, 222, 237
system, 86, 96, 101, 109, 114, 149, 154, 155, 217, 231, 249, 250, 252, 253, 316, 330, 444, 497, 516
Near field collision warning (NFCW), 153, 159, 162
Network
architecture, 174, 303
structure, 174
systems, 95, 140, 354
technology, 95, 116
topology, 106, 173, 175, 355
uncertainties, 176
vulnerability test (NVTs), 338
Networked vehicle, 7, 8, 133
Next
generation engineered systems, 172
Step operating system, 398
Next eXPerience (NXP), 31, 517
Night vision plus (NVP), 153
Nissan
motor company, 71
Node structure of an artificial neural network, 275
Noise removal, 536, 537
Non
maximum suppression, 530
object based detection, 524
rational B-spline basis functions, 52
uniform rational B-splines (NURBS), 52–54
Nvidia, 552
- O**
- Object
based approach, 524
detection, 159, 513, 523, 524, 533–535, 537, 556
management group (OMG), 565
Object oriented analysis (OOA), 408, 421
Objective-C, 379, 393, 394, 398, 399, 403
Observability, 200
Obstacle
and collision warning (OCW), 153
Off street parking, 486, 490, 503
Ola
auto, 471
masked number feature India, 475
micro, 468
mini, 468
price chart, 470
Prime, 468
On-board
diagnostics II port, 306, 332, 335, 365, 503
diagnostics (OBD), 117, 119, 323, 324
sensors, 494, 514, 557
Online transportation network, 461, 463
On-street parking, 256, 257, 486, 487, 489, 503
Ontology
concepts, 184, 185
Open
application layer interfaces, 248
GL ES, 392
location platform, 256
loop control system, 190
network technology, 85
systems interconnection (OSI), 115
unified process (OpenUP), 129
vulnerability assessment system
(OpenVAS), 337, 338
Operating system (OS), 144, 280, 312, 337, 379, 389–391, 393–395, 398, 407, 412
OPPO, 388
Optical flow algorithm, 534–536, 538–542
Optimization, 35, 60, 77, 89, 111, 113, 131, 146, 201, 202, 207, 216, 226, 275, 277, 279, 290, 404, 462, 544, 545
Opto electronic display, 35
ORACLE database server, 383
Original equipment manufacturer (OEM), 5, 6, 13, 25, 28–32, 35
branded workshops, 19
Origin-destination (O-D) vehicle ride, 250
Outsiders, 306, 330, 359, 371
Outside sensors, 494
Overall product quality, 45, 55
Over-the-air (OTA)
attack, 454
update, 241, 309, 335, 566
- P**
- Pair-of-actions, 288
Pandora®, 334
Park
house, 486, 490, 492, 494, 496, 497
house management system, 496
pilot, 514
and ride facilities, 486

- Parking
 app, 210, 488, 489, 504
 assistance system, 485, 492, 494, 503, 505, 514
 information, 252, 490
 maneuver, 96
 sensor (PS), 98
 space, 24, 235, 251, 440, 454, 461, 462, 485–491, 493, 495, 496, 503
- Partition
 hierarchical, 113, 114
- Passenger
 capacity, 47
 car, 13, 16, 30, 202, 236
 manufacturing, 143
 market, 3, 581
- Passive safety measures, 27
- Path planning, 497
- Pedestrian
 detection, 142, 155, 157, 162, 514, 536
 protection system (PPS), 155, 159, 162
- Penetration test, 265, 337, 349, 359, 360
- Peoplesoft, 380
- Performance
 business, 62
 driving practices, 48
- Personal learned destinations, 250
- Peugeot, 63, 233, 247
- Physical
 components, 106, 171–173, 178, 265, 266, 271, 272, 294, 295, 318, 362
 model, 55
 system requirements, 176
- Piaconet, 101, 223, 224
- Platform
 as-a-service (PaaS), 238, 239, 360
 strategy, 63, 229
- Platooning
 trucks, 154, 245
- Plug-in hybrid vehicle, 22
- Pollution load, 20
- Polynomial
 curve, 50, 51
 surface, 50
- Population dynamics, 285
- Porsche, 30, 33, 35, 490
- Positioning, navigation, and timing services (PNT), 97
- Power
 control, 86
 management module (PMM), 89
 seats, 83
 train control module (PCM), 321
- train technology, 19, 21, 47
 window and door control (PWDC), 90, 91
- Precision, 59, 102, 103, 157, 253, 254
- Pre-crash collision and mitigation system (PCCMS), 154, 158, 159, 162
- Predictably dependable computing systems project, 188
- Predictive intelligence, 214
- President's council of advisors on science and technology (PCAST), 172
- Probability of false alarm, 284
- Process
 management, 62, 66, 142
 modeling, 58, 66
- Product
 animation, 55
 data creation, 62
 data management concept, 45
 data management (PDM), 54, 61, 64–67, 69
 data model, 66
 development process, 29, 46, 48, 60, 130
 innovation process, 48
 life-cycle management (PLM), 45, 61, 69
 life-cycle (PLC), 45, 48, 56, 61, 69, 86, 120, 126, 142, 228, 311
 management, 64, 65, 71, 582
 plant, 47, 56, 63
 portfolio, 69
 related cost, 69
 revenue, 69
 structure management, 65
- Progressive access, 330
- Project phase, 46
- Prometheus project, 28
- Protection
 from malicious intrusion, 164
 for pedestrians, 27
- Protocol-oriented programming, 404
- Prototype iterations, 60
- PSA Peugeot Citroen, 63, 247
- Public
 private partnership (PPP), 172
 safety access point (PSAP), 236
 transport, 8, 24, 468, 584, 586, 588
 transportation, 2
- Q**
- Qualcomm
 connected car reference platform (CCRP), 237, 238
 technologies, 237, 238
- Quality management (QM), 361

R

- Radar, 95, 141, 157–158, 163, 243, 244, 254, 255, 335, 517, 525, 552, 553, 560
Radio
 communication, 351
 detection and ranging (Radar), 95, 96, 141, 156–158, 162, 243, 244, 254, 259, 516, 517, 525, 552
 frequency identification (RFID), 104, 171, 202–205, 207, 208, 210–214, 218, 258, 306, 442, 453, 490
 frequency (RF), 157, 205
 telemetric, 35
Radio frequency identification (RFID)
 tag, 202, 204, 212, 306, 453
Random
 access memory (RAM), 212
 hardware failure, 567
 nonnegative variable, 61
 uniform variable, 61
 variables, 61, 290, 301
Rapid prototyping, 131, 141, 148, 166, 315, 513, 552
Rate of infection, 286
Rational B-splines, 51, 52, 77
Real-time
 behavior, 126, 127, 150
 information, 257, 408, 489
 monitoring, 85, 177, 201
 operation, 114, 204, 268
 ride hailing, 465
 traffic alert, 8
 traffic information (RTTI), 251, 256, 488, 490
Rear view system (RVS), 154, 162
Receivers, 57, 117, 118, 159, 209, 218, 225, 228, 352, 354, 516
Recursion formula, 51
Recycling, 56
Reference input, 189–191
Release planning, 387
Reliability, 73, 111, 113, 114, 126, 127, 133, 143, 150, 159, 177, 178, 181, 189, 201, 205, 212, 218, 234, 312, 318, 325, 342–344, 361, 381, 387, 569, 572, 586
Remote
 access tunnel, 307
 car door opening, 34
 code execution, 366
 control, 34, 139, 226, 384
 deactivated critical safety elements, 307
 diagnostics, 236, 334, 335, 384, 585
 keyless entry (RKE), 91
 network access, 271
 parking, 493, 496, 550
 procedure call (RPC), 564, 565
 services, 250, 251, 305
Renault, 23, 30, 63
Rental car company, 454
Repair
 tear, 17
 wear, 17
Repository
 central, 61
Requirement
 analysis and negotiation, 60, 185
 definition, 187
 developing a pragmatic taxonomy, 327
 elicitation, 60, 184, 185
 engineering, 45, 59, 127, 132, 185–187
 identification, 60, 185
 management, 60
 non-functional, 386, 409, 569, 586
 real-time, 114, 116, 137, 177, 188, 189, 305, 561
 specification, 59, 60, 132, 185
 validation, 60, 185
Research and development (R&D)
 budget, 37, 47
Resilience
 test, 34
Resolution, 68, 103, 109, 115, 157, 158, 313, 518, 524, 526, 544, 546
Resource shortness, 163
Retrieval management, 65
Return on investment (ROI), 60, 70
Reusability
 of functions, 144, 312
Reverse engineering, 331, 334, 337
Ride
 comfort, 94, 150
 hailing, 24, 461, 463, 465, 468, 473, 477, 479, 491, 559, 582, 586
 companies, 462, 463, 468, 491, 559
 economy, 462
 share service, 461, 473
 sharing, 379, 404, 407, 411, 462, 469–471, 473, 475, 477–479
 app, 405, 475, 582
 arrangement, 461
 companies, 461, 464, 476, 477, 479
 economy, 241, 472
 platforms, 473, 474, 476
Risk level
 identification, 328

- Rivest
cipher 4 algorithm, 352
Shamir-Adleman encryption/decryption, 356
- Road
departure protection (RDP), 154
safety, 7, 164, 215, 216, 244, 245, 253, 513
side unit (RSU), 2, 98, 235, 302
sign recognition (RSR), 154
transportation, 201, 216
- Robot operating system (ROS), 565
- Rollover, 521
- Routing
algorithm, 174, 175
schemes, 174, 175
- Runtime environment (RTE), 144–146, 148, 312
- Rural drive assistance (RDA), 155
- Rush hour, 21
- S**
- Safe
distance, 96, 100, 516
failure fraction (SFF), 121, 123, 124, 361
- Safety
active, 127, 154, 155, 241, 243, 251, 253, 513, 514, 550
assistants, 253, 513
belt, 26, 100, 154, 514
control, 86, 92, 93, 97, 121, 127, 130, 149, 151, 164, 210, 226, 245, 266, 332, 333, 360, 398, 514, 550
features, 89, 94, 95, 127, 130, 149, 164, 225, 226, 231, 240, 242, 321, 333, 334, 404, 473, 475, 550, 557
function, 88, 89, 120, 127, 130, 150, 151, 226, 398
functional, 10, 29, 83, 95, 120–127, 132, 150, 227, 265, 303, 350–362, 385, 566–570, 583, 588
integrity level (SIL), 120, 122, 360, 566
life-cycle phases, 122
passive, 27, 93, 127, 243, 253, 514
performance, 28, 35, 111, 121, 130, 360, 404
related software, 122
requirements, 83, 120, 122, 126, 127, 130, 132, 142, 143, 150, 249, 311, 324, 325, 361
in ridesharing, 472
risk for ridesharing passengers, 474
standards, 10, 28, 89, 95, 120, 122, 123, 127, 132, 229, 231, 350, 360, 361
- Sales documents, 55
Samsung, 70, 388
Santa clara law review 1145, 246
- Satellite navigation system, 516
- Scalable software architecture, 552
- Scania, 30
- Scoping, 47
- Scrum
development process, 384
master, 384
project, 384, 385
release planning, 387
- SDL suite, 184
- Seat comfort (SC), 91
- Securing automotive mission-critical components, 342
- Security
breaches, 309, 549, 567
challenges, 265, 266, 303, 304, 350
flaws in modern vehicles, 364
layers, 203, 355, 356, 383
measures, 293, 319, 340, 350, 351, 355, 356
objectives and their impacts, 325
of data, 7, 323, 324, 357, 569
platform, 233, 265, 356, 357
procedures, 307
view on security, 289
- Self
driving, 8, 9, 28, 34, 147, 164, 241, 254, 513, 549, 550, 552, 558–560, 566, 588
driving vehicle, 8, 9, 28, 34, 147, 164, 241, 254, 513, 549, 558–560, 566
parking, 242
- Semiconductor platforms, 552
- Send to car, 249
- Sensor
data fusion, 105, 106, 176, 556
network node, 173
network security, 355
node, 105–109, 118, 173, 174, 178, 213, 271, 272, 280, 317, 325, 354, 355
output voltage, 103
technology, 28, 83, 87, 102–105, 156, 216, 222
- Sequence diagram, 409, 410
- Sequential structure, 46
- Service
delivery, 8, 62, 239, 240, 381
discovery, 564
oriented architecture (SOA), 230, 561, 583
robotics, 28
- Shadow removal, 536, 537

- Shared
economy, 8, 220, 233, 472, 487
sensor and actuator networks (SANs), 106,
 176
service centers, 382
- Sharing economy, 8, 220, 233, 241, 472
- Side-channel attack, 309, 339
- Signal
processing, 188, 543, 552
processing algorithms, 28
- Signature-based detection, 296
- Silicon Valley, 34, 35
- Simple rational B-spline surface algorithm, 52
- Simplex, 51
- Simulink®, 131, 184, 200, 513, 543, 544, 552,
 569
- Simultaneous engineering (SE), 56, 567
- Single vehicle roadway departure, 521
- Skidding control, 110
- Smart
car, 46, 439, 449, 454
city, 209, 210
hardware, 266
mirrors and wipers (SMW), 91
mobility, 2, 8, 210, 239, 241
phone, 25, 70, 209, 225, 227, 231, 252,
 462
phone market, 388
road, 244
street lights, 210
ticketing, 221
traffic signals, 245, 303
traffic signs, 245
transit systems, 245
watch, 8, 34, 35
- Smog, 20
- Sobel method, 530
- Social networks, 223, 249, 287, 301, 462
- SoftBank pepper robot, 35, 36
- Soft real-time systems (SRTS), 119, 188
- Software
as-a-service (SaaS), 238
based cryptographic methods, 324
engineering approach, 182
in the loop simulations, 151
market volume, 74, 85
security, 237, 305
sophisticated, 116, 266
- Solutions
embedded, 227, 231, 256, 269, 355, 446
remote, 227, 496
tethered, 227
- Sound pressure level (SPL), 161
- Spare parts, 8, 29
- Spatial orientation, 58
- Spear phishing, 265, 301
- Specific object detection, 537
- Speed
control unit (SCU), 321
limit, 26, 154, 155, 254, 257, 490, 514
- SPLITVIEW, 252, 253
- Spoofing, 306, 309, 310, 313, 340, 353, 454,
 497, 566
- Sport utility vehicle (SUV), 13, 357, 469
- Spreading mechanism, 285, 287
- Sprint
network, 367
- SpriteKit, 392
- SQL Server, 383
- Stage gate®
controlled development process, 45, 48
- Stakeholder
needs and requirements (SNR), 59
- Standardization, 63, 72, 116, 127, 143, 145,
 215, 220, 228, 231, 248, 312, 313,
 566
- Starship delivery robot, 37
- Startup autobahn, 35, 36, 38, 581
- Statemate®, 184
- Static characteristic, 55, 102, 156, 193, 265, 296
- Statistical
analysis, 280, 326, 362, 544, 545
hypothesis testing, 301
- Steer-by-wire, 100, 116
- Steering
assistance, 492
and backing off maneuvers, 492
wheel, 93, 96, 100–102, 152, 153, 243, 249,
 253, 255, 365, 504, 516, 518, 522,
 552, 554, 555
wheel angle sensor (SWAS), 100
- Steering wheel, 96, 100–102, 153, 243, 249,
 253, 255, 365, 504, 516, 518, 522,
 552, 554, 555
- Step response, 194–196, 199
- Stochastic customer arrivals, 220
- Stop-over phase, 444, 446
- Stream cipher, 351–353
- StreetScooter, 585
- Strengths-weaknesses-opportunities-threats
analysis (SWOT), 5
- Stress test, 348
- Structural query language (SQL), 187, 239,
 383, 413
- Subscriber identity modules (SIM), 231, 250,
 252

- Subversion of a node, 355
 Sunroof (SHD), 5, 8, 87, 91, 92
 Supervisory control and data acquisition (SCADA), 107, 177
 Supply chain management, 203, 205, 269
 Surface modeling, 51
 shape, 50–52
 Surge pricing mechanism, 461
 Surreptitiousness, 283, 284
 Susceptible-infected-recovered model (SIR), 286, 287
 Suspension control unit (SPCU), 321
 system, 321
 Swatch, 46
 Swift, 379, 393, 398
 Sybil attack, 499
 System identification, 107, 117, 171, 204, 206, 213, 221, 282, 297, 307, 543
 modeling, 53, 57, 60, 66, 84, 85, 131, 183, 185
 on-a chip (SOC), 236, 308
 requirement and specification (SyRS), 59
 security, 229, 230, 268, 455
 and software engineering, 59, 69, 177, 182, 185
 Systems, applications, products (SAP), 252, 380, 382, 490
- T**
- Tag active, 211–213
 characteristics, 211–213, 293
 passive, 211–213
 reader, 203–206, 211–213, 453
 Target costing, 46, 70
 Task completion time, 176
 Tata motors, 16
 Taxi aggregation model, 468
 app, 468
 ForSure (TFS), 465, 471
 Technological advances, 2, 163, 178, 208, 209, 212, 244, 307, 317, 322, 354, 462
 features, 4–6, 163, 209, 212, 214, 228, 234, 354
 Telco communication, 462
- Telematics components, 171, 214, 215, 218, 222, 332, 446
 control unit (TCU), 237, 502
 market segments, 222, 223
 technologies, 171, 214, 215, 231, 234
 Temporal key integrity protocol (TKIP), 319
 Tesla giga factory, 22, 589
 Testing, 4, 34, 47, 55, 74, 110, 123, 127, 129, 131, 133, 135, 137–139, 151, 171, 183, 235, 242, 247, 255, 265, 301, 308, 322, 336, 337, 343, 348–350, 360, 395, 561, 582
 Texture, 55, 392, 544
 Threats, 4, 5, 7, 19, 24, 25, 237, 239, 265–270, 272–274, 281, 288, 289, 294, 297, 303, 306, 308–310, 318, 324, 330, 331, 340, 342, 343, 345, 350, 358, 364, 453, 473, 474, 485, 496–497, 503, 504, 514, 566, 569
 Threshold technique, 533
 theorem, 286
 Tier1 supplier, 5, 28, 31, 35, 45, 46, 56, 62, 115, 142, 143, 149, 156, 225, 227, 228, 233, 240, 247, 248, 308, 311, 315, 360, 431, 493, 502, 517
 Time based intrusion detection, 282, 348
 of-flight (ToF), 161
 to-market (TTM), 49, 55, 62, 68, 181, 249
 T_I microcontroller, 197
 Timing attack, 318, 500
 Tire pressure sensor (TPS), 115
 Topology bus, 112, 561
 linear bus, 112
 ring, 112
 star, 111
 Total error, 540
 Toyota, 3, 30, 35, 71, 334, 365, 367, 468, 469, 581, 585
 Tracking GPS, 217, 218, 236, 474
 of trailers, 218
 of trucks, 245
 of vehicles, 200–202, 204, 218
 of wagons, 203
 Trade-off parameter, 289

- Traffic
 analytics, 490
 congestion, 20, 210, 216, 224, 251, 468
 jam, 20, 152, 203, 249, 254, 461, 462, 468, 557, 587
 light, 4, 29, 152, 164, 210, 216, 235, 244, 245, 255, 556
 message channel (TMC), 96, 224
 mixed, 246, 254
 regulation, 27, 88, 558, 560
 rules and regulations, 559
 sign, 98, 255, 256
 sign recognition (TSR), 98, 142, 155, 514, 525
- Train, 5, 24, 304, 450, 570, 586
- Transducer, 105, 161
- Transfer
 function, 191–193, 200
- Transferability of functions, 144, 312
- Transmission control
 module (TCM), 99
 protocol/internet protocol (TCP/IP), 206
- Transmission control protocol/internet protocol (TCP/IP), 206, 561, 563, 564
- Transmitter, 57, 118, 212, 225, 228, 306, 516
- Transponder, 211, 213
- Transportation
 freight, 2, 28, 202, 217
 passenger, 6, 28, 201, 203
 platform, 28, 407, 490, 586
 system, 8, 19, 98, 200–202, 204, 209, 216, 217, 241, 303
- Turnover by revenue, 16
- Two-speed IT, 383, 387
- Type 2/mode 3, 22
- U**
- Uber
 driver safety, 474, 475
 Go, 469, 470
 MOTO, 471
 panic button, 475
 price chart, 470
 safety net India, 475
 SUV, 469, 471
 X, 469–471
 XL, 469, 470
- Ubiquitous
 computing, 179, 201, 206, 214
 information network, 178
- Ultimate disposal phase, 45, 72
- Ultra
 precise HD map, 255
 sonic sensor, 160, 161
 sound, 156, 243, 489, 490, 492, 493, 495
 sound warning system, 492
- Unauthorized
 access, xiv, 88, 265, 267, 270, 314, 325, 338, 356, 496, 582
 intrusion of a web-based application, 363
 transmission, 351
- Unified modeling language (UML)
 profiles requirements validation, 185
- Unit step, 194–196
- Universal serial bus (USB)
 media playback, 234
 port, 116, 252, 324, 333, 366, 497
- UNIX, 331, 390
- Unmanned vehicles, 28, 277
- Urbanization, 584
- USA, 558, 559, 589
- Usage-based insurance (UBI), 19, 214, 221, 236, 585
- Use cases and architecture of car2go, 442
- Utilization phase, 45, 73
- V**
- Valet parking functionality, 454, 493
- Validation
 phase, 47
- Value
 chain, 19, 29, 35, 61, 62, 208, 228, 230, 233, 307–310, 450, 517, 589
 engineering, 46, 47
- Vault, 65, 66
- V-diagram, 322
- Vehicle
 ad hoc network (VANET), 95, 303
 audio system, 101
 connection, xii, 19, 101, 225, 227, 231, 240, 245, 305, 324, 336, 365, 566
 control module (VCM), 99
 density, 17, 18, 143, 311
 diagnostics, 9, 129, 222, 332
 disabled remotely via web application, 363
 emission, 19, 20, 71
 hacking, 26, 115, 164, 265, 306, 324, 330, 338, 362, 363, 365–368
 liability insurance, 220, 560
 location, 101, 111, 203, 215, 217, 218, 227, 228, 236, 237, 245, 250, 252, 255, 256, 305, 414, 445, 473, 516

- Vehicle (*cont.*)**
- management, 2, 10, 22, 45, 46, 89, 90, 121, 142, 202, 216–218, 222, 226, 235, 241, 271, 307, 321, 334, 362
 - manufacturer, 5, 6, 16, 17, 28, 30, 31, 46, 117, 143, 147, 229, 308, 311, 333, 334, 559–561
 - model, 2, 23, 45–47, 154, 240, 322
 - security risk, 330, 332
 - self-diagnostic, 119
 - to-cloud data, 256
 - to-home (V2H), 215, 216, 244
 - to-infrastructure (V2I), 10, 95, 151, 209, 235, 244, 303, 496, 497, 503, 550, 557, 566
 - to-mobility (V2M), 303
 - to-road (V2R), 215
 - to-vehicle (V2V), 9, 95, 209, 215, 244, 303, 557
 - tracking system, 201–206
 - utilization, 28, 45, 142
 - vehicle communication (V2V), 306, 309, 323, 332, 335
- Vehicular communication system (VCS), 98, 306
- Velocity tracking, 129
- Venture capital firms, 35
- Verification, 60, 66, 71, 132, 151, 186, 283
- Vertex, 54
- Video
- camera, 152, 154
 - conferencing, 68
- Vienna Convention for road traffic, 495, 558, 586
- Virtual
- product creation, 10, 29, 45, 64–68, 581, 589
 - prototyping (VP), 54
 - reality (VR), 57, 58
 - space, 58
- Virus
- activation, 297, 301
 - surrounding, 298
- Vision system, 159, 516
- Visteon, 247, 517
- Visual collaboration tools, 68
- Visualization
- technique, 57
- Voice control and telephony (hands free), 253
- Volcano automotive group, 115
- Volkswagen
- diesel gate scandal, 71
 - group, 21, 33, 71
- Volvo, 115, 233
- Vulnerability
- instances, 292
- Vulnerable
- access point, 306
 - scanning, 337
- W**
- Warning light, 323
- Wear and tear, 17, 448, 450
- Weather maps, 252
- Web
- browser, 253, 357, 395
 - radio, 253
- Wet road, 522
- Wheel speed sensor (WSS), 92, 93, 100
- Widget, 395, 417, 420, 423, 430
- Wi-Fi
- protected access, 319
 - services, 9, 466
- Wind river, 247
- Window lift, 87, 90, 323
- Windows
- embedded stack, 447
 - phone, 389
- Wired equivalent privacy (WEP), 319, 351–353
- authentication, 319, 352, 353
 - encryption and decryption, 352, 353
- Wireless
- access for the vehicle environment (WAVE), 216, 235
 - communication, 9, 88, 95, 107, 173, 202, 207, 213–216, 223, 231, 234, 243–245, 267, 280, 301, 303, 336, 350, 354, 367
 - local area network (WLAN), 95, 215, 319
 - mesh network, 173, 301
 - sensor-and-actuator network (WSAN), 176
 - sensor network (WSN), 106, 173–175, 354–356
 - vehicle safety communication (WVSC), 215
- Wireless local area network (WLAN), 102, 215, 319, 344, 351, 494
- Work
- flow of repair, 492, 503
 - plan, 46, 48

Worldwide production of cars, 15

Wrong-way driving warning (WWDW), 98

X

Xcode

integrated development environment (IDE),
379, 393, 394

interface builder (IB), 394, 399, 417

programming environment, 393

Y

Yaw rate sensor (YRS), 100

Yelp®, 334

Z

Zero

days, 318, 454, 566

sum, 290

Zombie detection, 394