Chapter 16
Functional Safety of Automated Driving Systems: Does ISO 26262 Meet the Challenges?

Helmut Martin, Kurt Tschabuschnig, Olof Bridal, and Daniel Watzenig

16.1 Introduction

Science fiction stories about autonomous cars have inspired the imagination for many years. In the early 1980s, the television series *Knight Rider* presented the self-driving and artificial intelligent car named KITT,[1] and the slogan went, 'Knight Rider—A shadowy flight into the dangerous world of a man who does not exist'. Techies of the time were fascinated by the possibility of a technology and imagined that it would be possible to drive or simply travel in cars of the kind in the near future. Today, some decades later, that vision is starting to be made a reality, which will change and further influence the common understanding of the existing human road mobility system. For the last 30 years, the main innovations of vehicle technologies have been achieved by E/E systems in the automotive industry [1], e.g. anti-lock braking system (ABS) in 1978, electronic stability program (ESP) in 1995 and up to collision avoidance systems in 2010 (see Fig. 16.1).

New generations of the advanced driving assistance systems (ADAS) are more complex than ever before in two aspects: firstly from a technical point of view in

[1] Knight Industries, 2000.

H. Martin (✉)
Virtual Vehicle Research Center, Graz, Austria
e-mail: helmut.martin@v2c2.at

K. Tschabuschnig
Magna Steyr Engineering AG & Co KG, Graz, Austria

O. Bridal
VOLVO Group Trucks Technology, Gothenburg, Sweden

D. Watzenig
Virtual Vehicle Research Center and Graz University of Technology, Institute of Electrical Measurement and Measurement Signal Processing, Graz, Austria

D. Watzenig, M. Horn (eds.), *Automated Driving*,
DOI 10.1007/978-3-319-31895-0_16

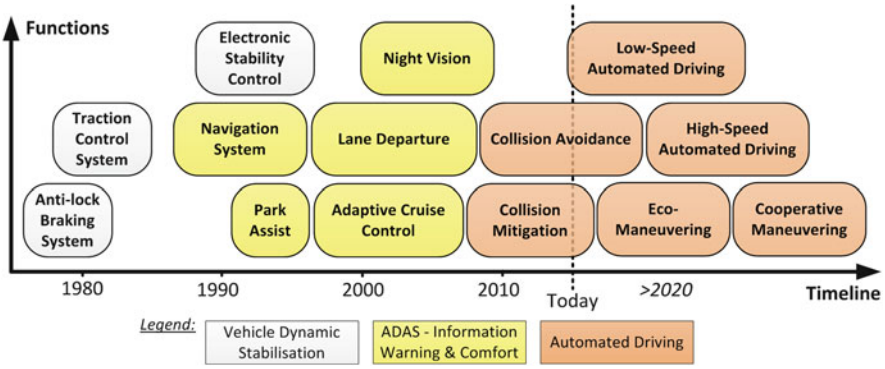

Fig. 16.1 Evolution of advanced vehicle functions

the context of the introduction of new technologies for implementing the functions required and secondly from an organisational point of view concerning the whole supply chain including the suppliers involved for a different kinds of services and products during the lifecycle of an automotive vehicle. In this chapter, we will focus on the technical aspect as well as on the discussion about the challenges of automated driving functions and of how to apply the existing version of the ISO 26262 [2] standard concerning automotive functional safety.

16.1.1 From Driver Assistance to Highly Automated Driving Systems

Today, almost every car in the market provides driver assistance systems (e.g. electronic stability control—ESC). For safety reasons, high-class vehicles are equipped with various additional ADAS functions (e.g. adaptive cruise control—ACC). The introduction of such systems has helped to reduce the number of fatal accidents [3, 4]. However, more than 90 % of accidents still occur as a result of human misbehaviour or mistakes. Thus, it is an important topic for the European Union to reduce the number of human-caused accidents by introducing the next generation of ADAS for our cars, which are referred to as automated driving systems (ADS).

The different definitions of driving automation for on-road vehicles by SAE in the standard J3016 [5] and recommendations provided by BASt[2] and NHTSA[3] are shown and compared with each other in Fig 16.2. The comparison between the

[2]Germany Federal Highway Research Institute (BASt)—http://www.bast.de.

[3]US National Highway Traffic Safety Administration (NHTSA)—http://www.nhtsa.gov/.

Level	Name	Narrative definition	Execution of steering and acceleration/ deceleration	Monitoring of driving environment	Fallback performance of *dynamic driving task*	System capability (*driving modes*)	BASt level	NHTSA level
***Human driver* monitors the driving environment**								
0	**No Automation**	the full-time performance by the *human driver* of all aspects of the *dynamic driving task*, even when enhanced by warning or intervention systems	Human driver	Human driver	Human driver	n/a	Driver only	0
1	**Driver Assistance**	the *driving mode*-specific execution by a driver assistance system of either steering or acceleration/deceleration using information about the driving environment and with the expectation that the *human driver* perform all remaining aspects of the *dynamic driving task*	Human driver and system	Human driver	Human driver	Some driving modes	Assisted	1
2	**Partial Automation**	the *driving mode*-specific execution by one or more driver assistance systems of both steering and acceleration/deceleration using information about the driving environment and with the expectation that the *human driver* perform all remaining aspects of the *dynamic driving task*	**System**	Human driver	Human driver	Some driving modes	Partially automated	2
***Automated driving system* ("system") monitors the driving environment**								
3	**Conditional Automation**	the *driving mode*-specific performance by an *automated driving system* of all aspects of the *dynamic driving task* with the expectation that the *human driver* will respond appropriately to a *request to intervene*	System	**System**	Human driver	Some driving modes	Highly automated	3
4	**High Automation**	the *driving mode*-specific performance by an *automated driving system* of all aspects of the *dynamic driving task*, even if a *human driver* does not respond appropriately to a *request to intervene*	System	System	**System**	Some driving modes	Fully automated	3/4
5	**Full Automation**	the full-time performance by an *automated driving system* of all aspects of the *dynamic driving task* under all roadway and environmental conditions that can be managed by a *human driver*	System	System	System	**All driving modes**	-	

Fig. 16.2 Definition of SAE driving automation levels for on-road vehicles and comparison with BASt and NHTSA [36]

levels proposed by the various standards/recommendations is possible up to the BASt Level 4 'fully automated' (see blue line in Fig 16.2).

Evolution of driving systems (based on the definition by BASt/Lx . . . Level x):

L0. *Driver only*—Driver assistance comfort system (e.g. speed limiter).

Responsibility: Driver.
Safe State: Driver always has the control of the vehicle.

L1. *Assisted*—Advanced driver assistance provides safety improvement; ADAS supports the driver (e.g. EBA,[4] ACC, LKA[5]).

Responsibility: Driver.
Safe State: Driver takes over full control of the vehicle.

L2. *Partly automated*—Driving system controls laterally and longitudinally for a certain time in few situations (e.g. motorway assistant).

Responsibility: Driver.
Safe State: Driver takes over full control of the vehicle.

L3. *Highly automated*—Driving system controls lateral and longitudinal movement for a certain time in specific situations (e.g. motorway chauffeur).

Responsibility: Driver *or* system.

[4]Emergency Brake Assist.
[5]Lane Keeping Assist.

Safe State: Driver takes over full control of the vehicle within a specific timeframe *or* system has to control the vehicle in defined driving situations, if the driver did not take over full control.

L4. *Fully automated*—Driving system has complete control of lateral and longitudinal movement within a specified situation of the application (e.g. motorway pilot).

Responsibility: System.
Safe State: System controls the vehicle in some driving situations.

In SAE J3016, the highest level is 'Full Automation', which means from our perspective an autonomous vehicle that is able to drive without a driver. This level is not reached in this chapter because this scenario is too far away from today's technical practice.

The role of the driver will continue to be important for the introduction of automation functions in vehicles over the next few years. For high levels of automation, the driver should not be required to cope with any critical driving situation. In such cases, the ADS should be able to handle any kind of driving situation autonomously—but this is still a future perspective expected that is expected to become reality around the years 2025–2035.

In the past, vehicle manufacturers realised their particular ADAS functions independently on a do it alone basis and using different OEM[6]-specific trade names (e.g. Adaptive Cruise Control (ACC), Active Cruise Control (ACC), Cooperative Adaptive Cruise (CACC), Distronic Plus). The function itself as well as the handling and the user interaction typically slightly differed from each other to guarantee OEM-specific originality. The levels of automation have to be harmonised for the introduction of ADS functions; otherwise, the driver will not be able to operate different systems in the required way without training or a special extended driving licence for automated vehicles as recommended by NHTSA [6]. One important aspect for handling the challenges is the standardisation and harmonisation of ADS functions of all OEMs on the market. The standardisation must include not only the vehicle itself but also the overall aspects concerning the ecosystem that are required to realise ADS functions like infrastructure (e.g. map data) or environment (e.g. secure C2X[7] communication). In aviation, the rulemaking advisory committee ARAC[8] harmonises all the aviation-specific standards (e.g. for system failures, underdetermined air traffic situation and human factor faults). The awareness of the need for such a rulemaking advisory committee for road vehicles is also given in the automotive industry as an automated vehicle will not be a closed system as was the situation in the past.

[6]Original equipment manufacturer.

[7]Car-to-x means a communication between the car and any other external system, e.g. other cars C2C or the infrastructure C2I.

[8]Aviation Rulemaking Advisory Committee—http://avstop.com/legal/2.htm.

Fig. 16.3 Overview of different safety levels

If we compare the situation of aviation with the road mobility standards concerning safety, ISO 26262 today covers only a subset of those system safety regulations. As an example, we wish to mention the interaction of ADS with the driver in aspects such as warning of the driver, supporting the driver so that an appropriate reaction can occur and feedback to the driver concerning his/her reaction. Only if the reaction of the vehicle is clearly defined and the driver knows which actions are carried out by the vehicle on its own, the right decision or reaction can be expected from the driver within a specific driving situation when needed.

16.1.2 Functional Safety According to ISO 26262

Safety is one of the key issues of road vehicle development. New innovative vehicle functionalities are not only introduced as driver assistance functions. Concerning propulsion, vehicle dynamics control and active and passive safety systems increasingly enter the domain of system safety engineering. Development and integration of these functionalities will enforce the need for a serious consideration of safety within the system development and the need to provide evidence that all reasonable system safety objectives are reached [5].

There are different levels of safety (LoS) (see Fig. 16.3):

LoS1. *Safety with Respect to Product Liability*[9] where safety aspects of any kind must be covered in order to achieve the permission for the launch of a

[9]For example, Austrian Federal Act—Governing the Liability for Defective Product/Product Liability [7]: §5. (1) A product §5. (1) A product shall be deemed defective if it does not provide the safety which, taking all circumstances into account, may be reasonably expected, in particular with respect to: (1) the presentation of the product, (2) the use to which it can reasonably be expected that the product would be put and (3) the time when the product was put into circulation.

product on a specific customer market (e.g. electrical safety of high-voltage systems)

LoS2. *Functional Safety* with a cross-divisional view of any type of malfunction in mechatronic systems (e.g. failure of a mechanical part that could lead to a hazardous event)

LoS3. *Functional Safety* with emphasis on any kind of malfunction of electrical and/or electronic (E/E) systems (e.g. failure within the hardware which must be monitored and handled to achieve the safe state of a system). This means for the automotive industry, the ISO 26262 standard has to be applied.

ISO 26262 'Road Vehicles—Functional Safety' is an automotive industry-specific derivation of the generic industrial functional safety standard IEC 61508 [8]. ISO 26262 was released in November 2011 as the state of the art international standard for E/E systems in passenger cars. It provides a structured and generic approach for the complete safety lifecycle of an automotive E/E system, including design, development, production, service processes and decommissioning. ISO 26262 defines the Automotive Safety Integrity Level (ASIL) as a risk classification parameter for the safety-critical hazardous situation of an item.[10] This is an important parameter for all subsequent safety activities in the safety lifecycle. The ASIL can be seen as a parameter that indicates a reduction of risk requirement in order to achieve a tolerable risk level.

The overall systems engineering must cover all kinds of system properties such as reliability, availability, maintainability, security and (functional) safety. Reliability engineering is closely related to safety engineering and to system safety. Both use common methods for their analyses and may require inputs from each other. Reliability engineering typically focuses on costs through failure caused by system downtime, cost of spares, repair equipment, personnel and the cost of warranty claims. Safety engineering normally does not emphasise costs but rather the preservation of life and nature. Therefore, it deals only with particular safety-critical and dangerous system failure modes [9]. Safety and reliability are different properties. A system can be reliable and unsafe while it can also be unsafe and reliable (see Fig. 16.4). Furthermore, in some cases, these properties even come into conflict with each other. Leveson discusses this problem with very interesting examples from the military as well as the avionic and chemical industries [10].

The ISO 26262 standard states 'ISO 26262 does not address the nominal performance of E/E systems, even if dedicated functional performance standards exist for these systems (e.g. active and passive safety systems, brake systems, Adaptive Cruise Control)'. ASIL is not a nominal performance metric for other system properties (e.g. maintainability, reliability, availability) of ADS functions. Specific metrics for other concerns need to be examined in certain analyses of the particular scope (e.g. mean time to repair (MTTR) for maintainable systems).

[10]An item is a system or array of systems for implementing a function at vehicle level, to which ISO 26262 is applied.

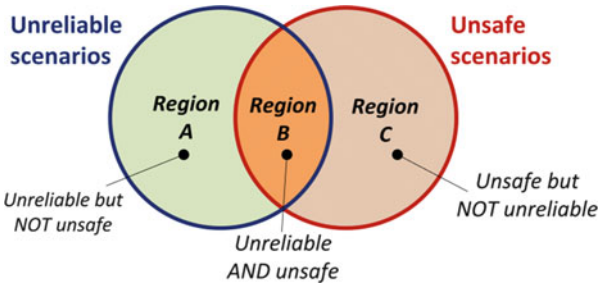

Fig. 16.4 Relation of unreliable and unsafe scenarios

The ISO 26262 standard provides guidance by introducing requirements and recommendations to reduce the risk of systematic development failures and to handle the complexity of E/E systems. Nevertheless, compliance with the standard presents a significant challenge for companies, because ISO 26262 sets requirements and recommendations but does not explicitly define how they should be implemented in an efficient way in the context of a particular application. To implement the requirements and recommendations of the ISO 26262 in a particular application, expert knowledge in functional safety must create a thoughtfully argued and documented interpretation of the ISO 26262 for the particular application.

ISO 26262 provides a systematic top-down engineering approach based on the V-model.[11] A specification starts from the system-of-systems (SoS) level down to the subsystem and component level and subsequently to the implementation level of hardware (HW) and software (SW) modules. After the implementation and verification of HW and SW, the integration a bottom-up approach follows on at the right side of the V-model: integration of HW and SW modules in components (e.g. HW+SW in ECU), components in subsystems (e.g. ECU in HV battery), subsystems to system (e.g. HV battery in powertrain) and system in SoS (e.g. powertrain in vehicle).

16.2 General Challenges of ADS

Some challenges are particularly relevant for automated systems in general terms (compared to 'classic' automotive electronic systems) and are related to complexity, availability and reliability. This section provides an overview of different kind of challenges that must be investigated for the development of safety-critical aspects of ADS.

[11] See definition at http://v-modell.iabg.de/v-modell-xt-html-english/index.html.

16.2.1 Increasing Complexity of ADS

A system can be described as an aggregation of elements or components concerning their cooperation and interaction with others to function properly. Interactions in a system are exchange processes between components realised by flow of material, energy and information (component relationships). In the event of failure, the system should be able to react in a fault-tolerant manner, which means that the system is able to trap a fault—'the system and its intended functions are able to survive' [11].

Safety is a system property intended to avoid system faults or malfunctions from causing any substantial damage (e.g. injuries to people or damage to the environment), which requires precise error detection. If an error is detected, the system must switch into a passive safe state with the consequence that the system is no longer available or reliable, but it is safe (failure integrity). The influence of system attributes such as availability, reliability, safety and security[12] must be harmonised, and a kind of trade-off is required, because the ADS can be safe but that does not mean that the system is available or secure.

If a system is required to guarantee high availability and fail-operational characteristics, the system architecture is expected to have higher complexity of implemented functions. This means that the system grows in terms of the number of components and the interactions between them. The effort involved for the additional system safety causes increasing complexity. In addition unexpected effects arise when repetitive interactions are effected by increasing non-linear functions between the components. The most important attributes [12] of complex systems are:

- *Nontransparency*—state, interconnection and behaviour of a system and its components are only partly known.
- *Sensitivity*—interference of results in the case of unexpected input changes.
- *Instability*—smallest disturbances cause unknown, unwanted behaviour of the system.
- *Internal dynamics*—continuous change of the system's state by the system itself without any external influence.

The mentioned attributes promote the appearance of additional faults and complicates their identification. Despite simplest components and interactions, the whole system generates forms, patterns and behaviour dynamics that could not be derived from particular components. This property is referred to as emergence,[13] which arises from various signal feedbacks of the system components.

One popular development method is to abstract the reality, which means building a model to simplify or reduce the reality and capture the interesting major behaviour

[12] See also 'dependability'—umbrella term to describe different quality attributes of a system.

[13] Emergent entities (properties or substances) 'arise' out of more fundamental entities and yet are 'novel' or 'irreducible' with respect to them [13].

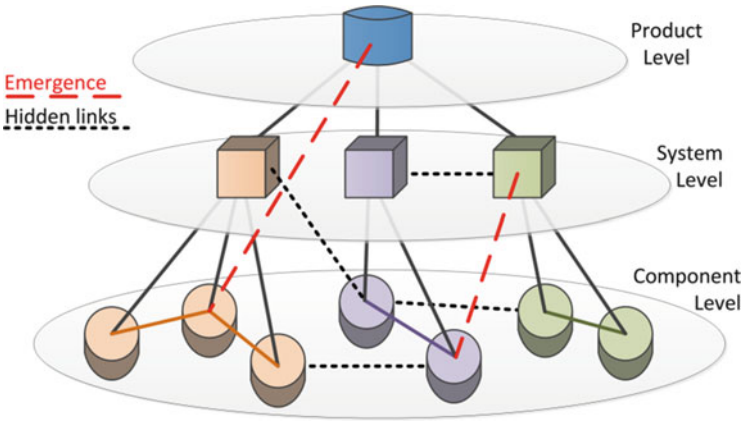

Fig. 16.5 Latent linkages between system components and integration levels

of the system. The state space of a model is always smaller than the state space of the real world because not all parameters such as temperature and friction that affect the components are considered. The synthesis of the component models does not show all operating states or all linking conditions. In particular, undesirable effects and hidden links could occur (see Fig. 16.5).

Unidentified coupling of components and over different integration levels may lead to systematic faults during the modelling of the systems. These nontransparency links and reactions to signals are the cause of unpleasant effects such as emergence (spontaneous system behaviour caused by smallest state changes on lowest level without direct derivation), common cause effects (single fault, cause simultaneously multiple components failure), powered run away (activation of a not provided function and is not designated in the conception or signal flow) and hidden links (unwanted operation states in the system, not identified as failure). In these cases, the system works incorrectly, while a faulty state is not visible. That could cause the loss of all safety reserves in the system. The implemented safety mechanisms are ineffective and cannot be activated because the functional chain is unknown. These nontransparency links must be discovered during the system design.

The mission of mastering complex systems is to control the above-mentioned impacts in time and to prevent injurious effects. This could be done by a safe system design and increasing system transparency. The quality, robustness and fault tolerance of the design depends on prediction potential of the applied development procedure.

16.2.2 Strict Requirements Concerning Availability and Reliability of ADS

A high degree of automation means that many—and potentially all—of the tasks usually carried out by the driver will now be executed by control systems in such a manner that the driver relies fully on the correct operation of these systems. The unavailability of a function—for example, the inability to perform automated braking or automated steering—is more critical when the driver is 'not in the loop' than it would be if the driver is 'in the loop'. Regarding safety, it is generally considered as acceptable that a semi-autonomous function such as conventional cruise control or adaptive cruise control is suddenly deactivated, provided the driver is informed about the deactivation. The deactivation could be caused by a detected error in the system, by the activation of a stability function such as ESP or ASR or by some other triggering condition. The sudden loss of the vehicle's ability to drive autonomously, perhaps after several hours of fully autonomous driving, would typically be considered highly critical concerning safety, even if the driver is forced to take over control of the vehicle. In an extreme case, the vehicle continues to operate fully autonomously and to the extent that the driver does not even have any possibility to take over control.

Thus, the closer we approach towards fully autonomous vehicles, the more important it becomes to ensure that automated functions are fully available. Classical 'fail-safe' design solutions that rely on deactivating a function and informing the driver are no longer sufficient. Instead of fail-safe designs, fault-tolerant designs will be needed so that functions remain operational even when a fault is encountered in the system.

In context of criticality of potential failures of functions for highly automated driving, it is clear that systems providing the functions are able to significantly affect the vehicle behaviour. Potential failures can cause very bad effects, and highly autonomous functions are, therefore, typically associated with strict requirements on safety integrity. However, it should be noted that many conventional systems also require high levels of safety integrity, for example, brake systems and steering systems. So, this aspect is not a *fundamental* difference between ADS functions and other vehicle functions. In general, automated functions tend to be associated with stricter safety requirements.

16.3 Challenges to ADS Concerning Functional Safety

For relatively high levels of automation (i.e. closer to 'autonomous driving' than 'driver warning functions'), a complexity issue must be faced that makes the safety analysis more difficult than that of conventional systems. In a 'classic' vehicle, the driver is responsible for coordinating all the vehicle functions (propulsion, deceleration, steering, headlamps, direction indicators, etc.). In principle, this means

that each independent system function can be investigated separately with respect to functional safety and taking into account the possibilities that exist for the driver to handle a particular malfunction of that vehicle function. But with higher degrees of automation, the driver is no longer the overall coordinator, which means that any malfunction need to be handled by another function. In fact, the limits between these functions become blurred and difficult to define since the interaction between the different functions grows which is now more complex. The ISO 26262 approach of looking at one function (or 'item', which is the real or imagined system that provides the function) at a time is less appropriate when the functions are heavily dependent on each other. In the following section, more safety-related topics will be discussed that must be taken into account for the engineering of ADS.

The innovations of today's vehicles follow a continuing evolutionary approach. The development of future technologies is based on existing automotive engineering best practices and does not only reuse the existing ones. Some of these evolutionary aspects will be discussed in the following.

16.3.1 Vehicle Platform for Basic Driving Functions

Many of the current discussions on ADS are concerned with the functional level to replace the single driver tasks by additional ADS functions. Further important issues that need to be covered are the basic actuation functions, such as accelerating, braking and steering, to implement the required vehicle movement. For these functions, today's vehicles provide function-specific assistance for the human driver through means such as force support in braking systems by a hydraulic or an electromechanic brake. Systems for automated driving functions need to be improved to support the fully required brake force without a human driver. Furthermore, the safety concepts of existing systems must be updated because the ECU (e.g. of the steering system) needs to detect any kind of malfunction and their effects have to be mitigated, because without a driver the system has to monitor, decide and react on its own. The steering system's safety goal can be formulated like, 'Avoid the reversible and irreversible steering request from the steering system affected by any of the involved E/E systems' (e.g. steering angle sensor or ECU) [14]. The 3-level monitoring concept (EGAS concept) provides a possible technical solution, which is a standardised principle for safety designs for vehicle engine controls published by German OEMs [15].

Future vehicle architectures will introduce new safety concepts in the automotive industry (e.g. steer-by-wire systems will change safety concepts in contrast to the systems nowadays). In the event of any fault, a deactivation in a fail-silent mode as a safe state will not be possible (e.g. a fail-operational mode can be realised by redundant system architecture). As a conclusion, it is obvious that the implementation of ADS functions in existing vehicle platforms cannot be seen as only add-ons to existing functions. The overall safety concept of vehicles has to be

updated for upcoming requirements concerning fault-tolerant and fault-operational behaviour of highly automated vehicles.

Issue: *Are existing vehicle platforms ready for ADS*?

16.3.2 From ADAS to ADS Functions

Today, ADAS functions are used as a basis for the realisation of ADS functions. However, these ADAS functions concern specific aspects of specific automotive use regarding:

- *Scenarios*: from simple to complex scenarios (e.g. from keeping a driving distance by ACC on the motorway to city chauffeur at traffic crossing)
- *Vehicle speed*: from low to high speed (e.g. from park steering assist to high-speed motorway chauffeur)
- *Vehicle Safety Risk*: from 'normal' to 'low' risk (e.g. from emergency braking assist to automated driving on the motorway)

The challenge is the combination and interaction of these basic functions. All kinds of interactions between these basic functions need to be analysed and handled in such manner that no unintended interactions concerning timing and value could occur. Any kind of functional and technical interaction must be dealt with during the system design phase.

Issue: *Is reusing of existing ADAS possible*?

16.3.3 Share of Sensors and Actuators

Different vehicle functions share the same sensors and actuators, and all functional and technical condition has to be met. Sensor signals and actuator command signals may not be faulty in the case of feature interaction and synchronisation. In many applications an adequate fusion of sensor data and a voter mechanism for actuator command signals are required. In particular, any kind of unwanted interactions has to be handled so that no hidden links could affect any malfunction behaviour.

Issue: *Is the available technology sufficient and adequate for the required functions*?

16.3.4 From Many ECUs to Host ECUs

Today, more and more functions of vehicles are implemented on existing single-core ECUs. These existing technologies slowly reach their limits (e.g. clock frequency, heat dissipation, size of gates). The following challenge approach is a shift from

single-core to multicore ECUs, which means a shift from distributed functions with many ECUs to a few multicore host ECUs. The latter offer many different functions, but this rather new technology also requires new safety features. For safety-critical applications according to ISO 26262, these multicore ECUs with shared resources have to support specific safety measures in hardware (e.g. use of lockstep core or memory protection). Furthermore, safety measures have to be supported by the software and software engineering constrains. Real-time (e.g. loads of cores), functional (e.g. sequences) and safety (e.g. spatial redundancy) aspects have to be considered by the operating system and the application software. Many new algorithms from different vendors have to be integrated in these platforms, and coordination, configuration and documentation pose a further challenge. All these aspects have to be compliant to ISO 26262 and require safety evidence for the assessment of those applications.

Issue: *Is new technology ready for safety-critical applications*?

16.4 Importance of the Concept Phase

The concept phase defined in ISO 26262 focus on the functional abstraction of a specific item by (1) definition of the item, (2) assessment of the potential risks of that item by performing the hazard analysis and risk assessment (HARA), (3) determination of the ASIL for each hazardous event, (4) definition of high-level functional safety requirements as safety goals and (5) derivation of a functional safety concept (FSC), which covers all relevant safety measures to achieve functional safety for the defined item. In the following, each of these activities is described and relevant steps will be discussed in more detail.

16.4.1 Item Definition

This activity covers the definition of the item, the required functionalities, the intended behaviour, the interaction with other items/systems of the vehicle and the interaction with the external environment of the vehicle. ISO 26262 is intended as an automotive-specific functional safety standard, and it should be usable for any kind of E/E system in a vehicle. This can be slightly different when considered beyond the scope of specific items. For example, if we compare a hybrid powertrain system component such as a high-voltage battery system with an automated driving systems for a motorway assistant (MWA): The MWA contains much more complex and networked functionalities that must to be coordinated with external items (e.g. other vehicles) and environmental systems (e.g. traffic signs) and furthermore with vehicle internal functions related to fundamental vehicle platform functions.

16.4.2 Hazard Analysis and Risk Assessment

In the concept phase, the functional abstraction allows to have an abstract view of the system. Functional safety concerns unintended behaviour of the item. Safety analyses should be carried out in that phase to identify potential hazards of the item (e.g. HAZOP[14] or Concept FMEA[15]) followed by risk assessment.

The following steps describe activities that need to be done during the HARA including some proposed further extensions concerning ADS functions; these are written in bold letters and described in more detail:

Step 1: Elaboration of Hazardous Events
- Step1.1: Driving scenarios by situation analysis
  - Driving situation (e.g. manoeuvre at crossroads)
  - Infrastructure (e.g. communication between car and environment)
  - Environmental condition (e.g. weather)
  - Operating mode of the vehicle (e.g. acceleration)
  - Traffic participants involved (e.g. pedestrian)
  - **Driver presence** (e.g. driver in the loop/or not)
- Step 1.2: Hazard identification (e.g. by HAZOP)
  - From malfunctions
  - To malfunction behaviour
  - To hazard
- Step 1.3: Derivation of hazardous events
  - Combine driving situation with hazards
  - Potential source of harm to specific group of traffic participants at risk

Step 2: Classification of Hazardous Events
- Step 2.1: Severity classification
- Step 2.2: Exposure classification
- Step 2.3: **Controllability classification**

Driver Presence and Controllability Classification Each hazardous event is classified by the risk parameters severity (S), probability of exposure (E) and controllability (C) during the HARA. Parameter C denotes the estimation of controllability of a hazardous event by the driver or other persons potentially at risk. Controllability classes are C0 to C3, where C0 meaning 'controllable in general' and C3 meaning 'difficult to control or uncontrollable'. In the specific context of risk assessment for automated driving functions, the parameters depend on the role of the driver within a specific driving situation, which is why an ASIL should be

[14]Hazard and operability study.

[15]Failure mode and effects analysis.

determined for any potential hazardous event. For ADAS and partially automated functions, the driver must always be able to take over control of the vehicle within a defined reaction time. Concerning functionality, for highly or fully automated functions, it is not required that the driver monitors the driving situation. Thus, it might not be possible for the driver to consider any kind of controllability of the vehicle. This may lead to a classification of C3, which would result in ASIL C/D[16] worst case.

16.4.3 Determination of ASIL and Safety Goals

The next steps concern the rating of ASIL and the definition of safety goals:

Step 3: ASIL Derived from Risk Parameters
- ASIL = f (S, E, C) based on ISO 26262, part 3, Table 4

Step 4: Elaboration of Safety Goals
- Formulation of Safety Goals
- **Definition of Safety Goal attributes** (e.g. safe state)

Definition of Safety Goal Attributes A safety goal must be specified as a top-level safety requirement. We want to avoid any unreasonable risk of a possible hazardous event (e.g. 'unwanted acceleration shall not occur'). Safety goals are not expressed in terms of technological solutions but in terms of functional objectives. If a safety goal can be attained by transitioning to, or by maintaining of one or more safe states, then the corresponding safe state(s) shall be specified. Further relevant parameters regarding a safety goal are safe state, fault-tolerant time interval (FTTI),[17] diagnostic test interval (DTI),[18] fault reaction time (FRT)[19] and safe tolerance time (STT)[20] to maintain safe state before a possible hazard may occur (see Fig. 16.6).

$$\text{FTTI} = \text{DTI} + \text{FRT} + \text{STT}$$

The definition of these parameters is very important in the case of FRT being required to have critical driving situations handled by the system or by the driver to maintain the defined safe state (e.g. ADS function level 2 defines safe state as 'driver takes over control').

Further Influences to Define a Safe State The complexity of the driving situation must be considered for the definition of safe states. Another important requirement

[16] Depending on the classification as S and/or E.

[17] Time span in which fault(s) can occur in a system before a hazardous event ([2], Part 3, 1.45).

[18] Amount of time in which a safety mechanism takes online diagnostic tests ([2], Part 3, 1.26).

[19] Time span between detecting a fault and reaching the safe state ([2], Part 3, 1.44).

[20] Amount of time between achieving the safe state before a hazard could occur.

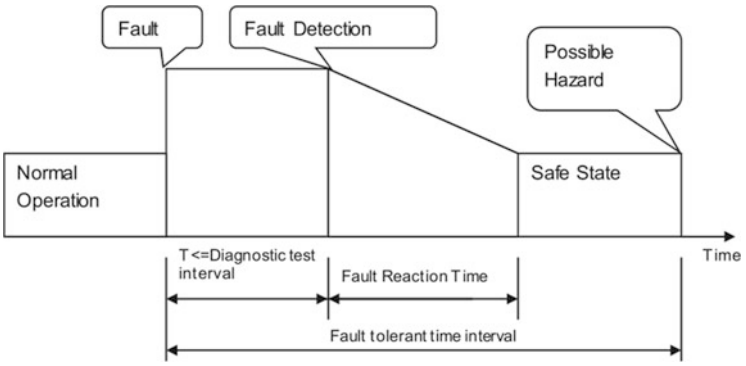

Fig. 16.6 Fault reaction time and fault-tolerant time interval [2]

Table 16.1 Overview of exemplary influences on the safe state

Item definition	Low ADS	Mid ADS		High ADS
Driver presence	YES	YES	NO	NO
System availability	Deactivation not available	Not available	Available	Available
Safe place	–	–	Stop vehicle on the same lane	Stop at the rightmost lane
Safe state scenario	Driver must take over	Driver must take over	Vehicle must stop at safe place	Vehicle must stop at safe place

in ISO 26262 concerning the safe state is '**8.4.2.4.** If a safe state cannot be reached by a transition within an acceptable time interval, an emergency operation shall be specified'.

Based on this requirement, further constraints have to be taken into account:

- *Item Definition*—provided functionality of ADS to maintain safe state (e.g. low ADS level, only comfort functions vs. high ADS level, self-driving).
- *Driver Presence*—difference between driver in the loop or not (e.g. driver's hands on the steering wheel vs. checking e-mails at the touchscreen).
- *System Availability*—possible or required degradation function depends on the level of ADS and the driver reaction in the case of malfunction.
- *Safe Place*—reachable safe place depends on the current driving situation and environmental conditions (e.g. safe state required during overtaking on the third lane of the motorway).
- *Safe State Scenario*—accessible safe state in specific driving situations including all constraints.

An overview of different influences is given in Table 16.1.

16.4.4 Functional Safety Concept

The objective of the functional safety concept is to derive functional safety requirements from the safety goals and to allocate them to preliminary architectural elements of the item or to external measures.

The following aspects have to be addressed in FSC:

- Error detection and failure mitigation
- Transition to a safe state
- Warning and degradation concept
- **Fault tolerance mechanisms**
- **Error detection and driver warning**
- **Arbitration logic**

The last three aspects will be discussed in the following in more detail:

Fault tolerance mechanisms means that a fault does not directly lead to the violation of the safety goal(s). The mechanism maintains the item in a safe state with or without any kind of degradation.

Error detection and driver warning are important to reduce the risk exposure time to an acceptable interval (e.g. engine malfunction indicator lamp, ABS fault warning lamp).

Arbitration logic is required to select the most appropriate control request from multiple requests generated simultaneously by different functions and is particularly important for the interacting functionalities of ADS.

However, not all of these aspects are always relevant for every system. Some systems do not offer any fault tolerance and some systems do not need any arbitration logic. The relevant safety measures concerning error detection, driver warning and transition to the safe state are important topics that must be considered in that phase.

16.4.4.1 Examples of FSC for Different ADS Levels

Depending on the type and degree of automation, there are several different strategies for ensuring safe operation despite faults in associated systems. This is illustrated in Fig. 16.7, which shows three potential event sequences unfolding after the occurrence of an error. From top to bottom, these can be described as follows:

An **assisted or partially automated function** can no longer be trusted to fully function and as a consequence the driver is alerted to (re)take control of the vehicle. During and after the handover, the partially automated function is prevented from working unsafely, perhaps by deactivating that function completely.

Example: Cruise control is deactivated due to a detected error. The driver is informed and takes control of the longitudinal motion of the vehicle.

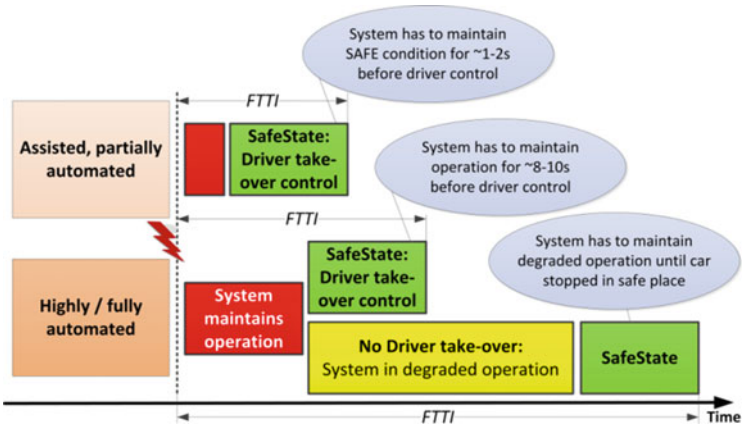

Fig. 16.7 Different concepts for transition to safe state

A **highly or fully automated function** determines that the driver needs to take over due to a detected error. The driver is informed about the need for handover of control. Due to the expected relatively long time for the handover, the automated function needs to continue to operate fully or almost fully for some time. Note: This means that the handover is initiated when the automated function is still either fully or almost fully operational.

Example: An autonomous driving system detects an error that indicates that an additional (subsequent) fault may lead to unsafe system behaviour. The driver is informed and takes control of the vehicle.

A **fully automated function** without any possibility for the driver to take over control determines that the vehicle shall be stopped in a defined time interval to avoid any hazardous event. As in the previous case, this means that the handover is initiated when the automated function is still fully or almost fully operational.

Example: An autonomous driving system detects an error that indicates that an additional (subsequent) fault may lead to unsafe system behaviour, so the automated function brings the vehicle to a safe stop within a few minutes or possibly seconds.

For the second and the third case described above, i.e. in the lower part of Fig. 16.7, it is shown that the automated function needs to be fully—or almost fully—operational for several seconds after an error occurs. If there is no driver to take over, the function has to remain operational, albeit potentially degraded, for several minutes. Thus, the implementation of such highly or fully automated functions needs to be fault tolerant in the sense that full or degraded functionality is possible even when a fault occurs in the system.

16.4.4.2 Vital Role of the Driver in the FSC

ISO 26262 sets requirements concerning error detection, driver warning and reaction of the driver. For today's automotive E/E systems, the role of the driver can be regarded as almost being covered in a cooperative manner. The driver must be able to control the vehicle on every trip (in Europe see also: Vienna Convention). By contrast, *how* the automated vehicles operate in a standardised way and *how* safety-critical aspects should be handled in a standardised way is *not* defined.

Thus, the driver needs to be familiar with different specific automated driving systems because the behaviour vehicles may differ. The training of the driver is required for specific ADS functions to ensure the driver's correct reaction within the required reaction time.

An additional aspect that must be taken into account here, and this is the 'habituation effect', i.e. the introduction of ADAS and ADS functions, will change the driving experience and require different skills of the driver. In HARA, the parameter C for controllability might change to 'uncontrollable'. In the near future, a driver may be unable to handle a critical vehicle situation without assistance systems within the required reaction time because of the lack of experience. Special driving licences for automated driving systems could be a possible scenario. However, they may not be accepted by customers who may hinder the introduction of such systems.

At present we do not train drivers to be able to deal with either a total brake failure or total loss of steering capabilities. Both braking and steering systems are extremely safe and reliable as a result so that the drivers do not need to worry about such problems at all. An alternative solution is simply to make the future ADS so safe and reliable that the drivers can fully rely on them at all times.

16.5 Supporting Methods to Handle Complexity of ADS

The complexity of these safety-critical systems must be considered, and negative effects need to be detected and mitigated by fault identification and fault mitigation techniques. Today, in the development of automotive electronic systems, there are established methods and technologies for safety activities available (e.g. safety analyses such as HARA for ASIL determination [2], failure mode and effect analysis (FMEA) [16], fault tree analysis (FTA) [17]).

The available technologies that need to be improved and developed further for their practical application in systems engineering:

- Formal/semiformal specifications by model-based systems engineering
- Formal verification by contract-based design
- Simulation and co-simulation

16.5.1 Model-Based Systems Engineering

The following definition of MBSE can be found in Friedentahl [18]:

'Model-based systems engineering (MBSE) applies systems modelling as part of the systems engineering process ... to support analysis, specification, design, and verification of the system being developed'.

The MBSE approach is a semiformal methodology to support engineers in the specification phase with analysis of the system and reduction of reality to an abstract model representation. The requirements for a specific level are defined and a virtual solution for the system is elaborated and hierarchically divided into representative components from system of systems, systems, subsystems and components. Models at a lower hierarchy level provide more specific details concerning the realisation. During the modelling phase, a separation of intended and unintended functions (= fault behaviour) is required, which is represented by specific functional properties and safety-related properties of the system. The model-based engineering approach is highly recommended by ISO 26262, part 6, for software development at ASIL C and D. This approach should be enhanced for the system level of such software-intensive systems. One of the major standardisation working groups concerning MBSE is the Object Management Group (OMG), which is an international, open membership, not-for-profit technology standards consortium. OMG task forces develop enterprise integration standards for a wide range of technologies and industries. Various standardised general purpose modelling languages are available for the system level (e.g. SysML,[21] MARTE[22] or EAST ADL[23]). These modelling languages have been elaborated, improved, applied and evaluated by many EU research initiatives by academia, research and industry partners. MBSE presents many possibilities for how to model a system through the use of different modelling elements, but for practical application, the reduction of the number of elements to a subset and provision of guidance and modelling constraints for engineers are requirements. A model-based systems engineering method[24] is a method that implements all or part of the systems engineering process and produces a system model as one of its primary artefacts. A system model provides the basis for specification of the intended behaviour of the system and is further used for identification and derivation of error models. An error model handles fault propagation over different hierarchy levels from singular components up to hazards at vehicle level. Different safety analysis methods (e.g. FTA or FMEA) can be supported by applying the error model. The output of the safety analysis defines safety measures by safety requirements for mitigation of any potential fault by detection, prevention,

[21] Systems Modelling Language—http://www.omg.org/spec/SysML/.

[22] Modelling and Analysis of Real Time and Embedded systems—http://www.omgmarte.org/.

[23] Electronics Architecture and Software Technology—Architecture Description Language—http://www.east-adl.info/.

[24] A method is a set of related activities, techniques, conventions, representations, and artefacts that implement one or more processes and is generally supported by a set of tools.

degradation or warning actions in the safety concept. A possible approach for the automotive domain by using SysML is described by Martin et al. in the SAE technical paper [19].

Biggs et al. [20] present a profile for a conceptional meta-model to cover relevant aspects of system safety and describes safety stereotype based on SysML (e.g. hazard, harm, harm context, etc.). The profile models common safety concepts from safety standards and safety analysis techniques. As a profile of SysML, it can be used to directly model the safety-related information of a system in the same model as that system's design. Furthermore, the profile supports communication between safety engineers and system developers; in order to improve the understanding on both sides of the risks, a system is vulnerable to and the features the system uses to mitigate those risks.

The MBSE approach by using SysML covers the following concerns [18]:

- *Provide a common and standardised description language* to improve the communication between system engineers and engineers from other disciplines.
- *Support of the performance of different kinds of checks* of the system model for the verification of specification rules (e.g. for the system design, to achieve correctness and completeness).
- *Improve the processing of the system modelling artefacts* by using transformation of the system model to another description model and extension with other relevant aspects (e.g. error modelling).
- *Traceability of relevant safety artefacts* is provided, and so the change management and impact analysis of particular safety concerns are possible. A further benefit of MBSE is the possible reuse of existing best practices by different kinds of patterns for requirements definition, safety design and safety argumentation.

16.5.2 Formal Verification by Contract-Based Design

Contract-based design (CBD) is a formal method for specifying what a component/system is able to offer (e.g. service, data, information, energy) for its environment by means of so-called guarantees and what a component/system requires (e.g. service, data, information, energy) from its environment by 'assumptions' [21]. Guarantees may be the performance and restrictions of output interface/channels which are only valid if all assumptions are confirmed. Assumptions define the environmental constrains for the input interface or channel of a system or component. The coupling of software-intensive systems and their components is hard to handle. It is difficult to handle all potential hidden links that could affect the safety of a system. CBD is able to guarantee that the system model only engages defined system states. By applying CBD, only specified system states are allowed and the coupling and communication of systems is only permitted via defined and well-known channels.

It is possible to provide patterns to assume and guarantee contracts which are defined for different characteristic such as timing, safety, security, etc. or patterns

that are formalised to be checked automatically. The sum of all the system patterns defines all possible contracts.

CBD describes system components to be black boxes and defines their behaviour via interfaces with other system components. All kinds of dependability aspects are formulated as contracts, for example, timing (e.g. real-time contracts), safety (e.g. ASIL x or reaction time) and security (e.g. authentication certificates) and are manageable by this means.

Different hierarchical levels of contracts are defined as follows:

- Contracts between different SW modules
- Contracts between SW modules and HW components
- Contracts between different HW components
- Contracts between HW components and subsystems

CBD is able to coordinate interoperability and boundary limits of components and services they provide and also data over different hierarchical organisations. By modularization, it is possible to reduce the complexity of the components during system design. Every component is described by a limited catalogue of properties and constraints which establish safety. Conflicts between contracts are found very easily by means of a consistency test, if all contracts are free of any contradictions. Satisfactory tests check whether the implementation of a component is consistent with the contract. Adequate tooling support is now finally available today (e.g. for model checking). Several publications discuss the use of contracts in context of the requirements of the engineering and safety standards such as ISO 26262 [22].

A new methodology to support the development process of safety-critical systems with contracts is presented by Baumgart et al. [23]. They compared existing meta-models also stating their shortcomings in relation to their approach, and they introduced the semantic foundation of our meta-model. They described their concepts of abstraction levels, perspectives and viewpoints and provided a proof of concept with exemplary use cases.

Westman et al. [24] shows that safety requirements can be characterised by contracts for an item and its elements with guarantees that constitute the safety requirements, by providing explicit requirements on their environments as assumptions. A contract therefore enriches a safety specification for an item/element by explicitly declaring what each element/item expects from the environment to ensure that the safety requirements are satisfied. Furthermore, they showed that consistency and completeness of safety requirements can be ensured through verifying the dominance property of contracts.

Past and recent results as well as novel advances in the area of contracts theory are presented by Benveniste et al. [25]. They show that contracts offer support to certification by providing formal arguments that can assess and guarantee the quality of a design throughout all design phases. Furthermore, they showed that contracts can be used in any design process: Contracts provides an ‘orthogonal’ support for all methodologies and can be used in any flow as a supporting technology in composing and refining designs.

16.5.3 Simulation and Co-simulation

Simulation methods are commonly used in the automotive industry where complex embedded systems from different cooperative disciplines are referenced to realise highly interdependent functions. In this context, simulation methods allow engineers to predict the behaviour of complex embedded systems without an available prototype of the entire system. Complex systems like ADS require a data structure that considers the behavioural interactions within the system because of their multidisciplinary nature. A combination of simulation and MBSE methodology supports modelling activities and improves the integration of simulation activities in the design process. This combination supports a system presentation for addressing the overall behavioural aspects of the product (multi-physics, local and global behaviours) and thus considers several system levels.

The ISO 26262 standard recommends the use of simulation methods for verification on different system integration levels (e.g. ISO 26262 part 3 for verification of the controllability parameter of HARA [26]). For system design verification, ISO 26262, part 4, Table 3 suggests simulation as a highly recommended method and a technique, e.g. fault injection and back-to-back test for ASIL C and D.

A model-based workflow for safety-critical embedded system is shown by Karner et al. [27]. Their approach covers three main aspects during the development of safety-critical systems, namely, system modelling, system simulation and system verification based on simulation. By using the Software Process Engineering Metamodel (SPEM), the workflow is defined in a consistent and seamless way, allowing continuity from preliminary concepts up to the final system verification report. Aligned with requirements given by ISO 26262, the demonstrated workflow enables safety verification at system level during an early stage of development by using modelling and simulation.

A system modelling-based approach for the integration and test of automotive embedded systems is proposed by Krammer et al. [28]. A V-model is introduced, targeting process-oriented needs for safety, and indicates whether modelling languages in favour can be applied best. To establish a link between safety goals and the structure of simulation models, the initial model is enriched with necessary information and transformed to a language suitable for advanced simulation tasks. SystemC has the capabilities to support this approach for hardware and software evenhandedly. The integration of SystemC into a co-simulation environment also enables the usage of external simulation models within the proposed architecture. The proposed system modelling-based approach enables safety verification and validation at an early stage of development.

Graignic et al. [29] propose a software framework based on a data model that manages complex system structures. This data model structures behavioural information that considers three major interactions: interactions between components simulation models, interactions considering multilevel behaviours (e.g. use of component simulation for a module simulation) and interactions between domain behaviours (e.g. thermal impact on mechanical components) in a so-called

co-simulation environment. Such methods can be used to perform early validation of the specifications by the MBSE approach to provide early validation feedback of adequate safety measures.

In the context of automated driving, different aspects beyond embedded systems behaviour are simulated such as the interaction of a vehicle with its environment, other vehicles or systems (e.g. Simulation of Urban MObility—SUMO [30, 31]), the interaction of a vehicle with a driver or the interaction of vehicle subsystems for dynamic proof of a specified behaviour of systems and components [32].

16.6 Further Safety-Related Topics

In the following section, more safety-related topics will be discussed that must be taken into account for the engineering of ADS.

16.6.1 Influence of Security on Safety Functions

One objective of system development is to ensure 'freedom of unreasonable risks' in any operational condition. This objective has different meanings depending on whether safety or security aspects are considered. From the safety point of view, the risk to the environment arising from inside of the system must be minimised (and this apart from a system including humans). This can result in a technical failure in the system, for example, fire hazard due to a high-voltage battery system of an electric vehicle or an accident because of an unintended acceleration of the ADS. Regarding security, potential threats to the system through the environment, which could result from intentional manipulations, e.g. a hacker attack, must be minimised. While the term safety represents the system view on any potential hazards of the system to the world outside, security concerns by contrast the aspects from the outside world to the inside of the vehicle and the influence on the vehicle internal systems. The goal of security measures is to protect the system from unauthorised use and manipulation (hacker, low-cost spare parts, etc.). The discipline of security in the automotive industry concerns the growth in vehicle functions and the innovation potentials in the networking of vehicles with the environment (e.g. other cars) or Internet of things (e.g. cloud services). The particular challenge on the one hand is the linking of the two disciplines' safety and security for utilising synergies and on the other hand the prevention of conflicting effects. Different motivations for unauthorised access scenarios in vehicles are possible [33]:

- Manipulation of the vehicle and its components as well as the corruption or deactivation of vehicle functions—attacking of 'availability of a service' (e.g. change of torque limits of the electric machine that could damage the powertrain)

- Vehicle tuning by changing functional properties—attacking of 'functional integrity' (e.g. chip tuning, manipulation of the speedometer or deactivation of warning messages)
- Illegal attempts to obtain personal data—attacks on 'personal integrity' (e.g. the driving behaviour of the user, preferences for shops, restaurants or hotels)

ISO 26262 provides guidance for automotive development process issues concerning functional safety lifecycles. However, a process for security concerns is not state of practice for automotive engineering. Many similarities exist between safety and security on a common abstraction level, and it would appear to be useful to interweave ISO 26262 development processes with security concerns. After defining a security item, the result of these considerations could be the consideration of security risks and the preparation of hazard analyses. Security goals with corresponding security measures can, hence, be derived from the analyses. After system design, verification and validation, a joint assessment should take place to rate the functional safety level reached according to ISO 26262 and any safety threat on the security side. Based on the similarities of these two disciplines, it would appear to be wise and necessary to expand the ISO 26262 framework by aspects of security topics. The extent to which these suggestions or other methods are expedient will be established in the course of an ongoing discussion in different standardisation communities [33].

16.6.2 Liability of ADS

Liability is a crucial topic in the context of future automated vehicles because legal authorities need an answer to the question, 'who was responsible?' in case of an accident.

Different responsibilities can be found under the law [34], e.g.:

- *Liability of the vehicle keeper*: Any operational risk in connection with an automobile is born by the vehicle keeper—ADS will not change the liability for the operation of automatic systems in motor vehicles.
- *Liability of the driver*: In damage event a fault of the driver is legally assumed (under civil law) until proof of the contrary is provided. In case of a fault of the ADS, the driver still has the option to insist on proof of exoneration.
- *Motor vehicle liability insurance*: If a harmed third party raises claims against vehicle keepers or drivers, they will be covered by the insurance—ADS will not cause any relevant change of the liability principles of the motor vehicle liability insurance.
- *Product liability of the manufacturer*: The OEM is liable if a defective product was brought to the market being subject to product liability. The OEM must provide evidence that the product was not defective and did not cause damage. The drivers must be instructed carefully in order to reasonably influence their expectations about the system's capabilities and to encourage drivers to perform

any necessary overriding functions. The safety of the system design is closely linked to the instructions given to the driver.
- *Liability of the infrastructure*: Future highly and fully automated vehicle functions will require precise data. These data will refer to local conditions too and will require a time stamp. The vehicle infrastructure should be able to provide all necessary information and is also liable for safe and secure functionality.

Ethical aspects will also play a role. In complex driving situations, events may occur that are difficult to handle by human drivers and that could lead to so-called dilemma situations. Sometimes, it is not possible to manage critical situations without harming any people. Thus, a decision has to be made to determine the minimum of harm. A decision between 'plague or cholera?' is a difficult one for humans to make, but it is even more difficult for machines. Future highly and fully automated systems will need certain risk determination algorithms that can rise to situations of this kind.

For this reason an 'event data recorder' in the vehicles will be a requirement for recording relevant information about crashes or accidents. Information from these devices is collected and analysed after a crash to help in determining exactly what happened. This will be similar to the 'black box' found in airplanes, which records all critical data in the course of a flight. Further research is needed for the assessment and classification according to the level of abstraction and degree of automation for a standardised definition and understanding.

16.6.3 Validation of ADS Functions

Systematic testing methods are very important for the validation of ADS functions (e.g. concerning safety aspects) [37]. For such complex systems, test methods must comprise a combination of simulation and real-world testing for different levels of integration like xiL (x in the loop) and model/software/processor/hardware/vehicle in the loop approaches. The most widely used approaches for the validation of driving functions are based on the V-model, endurance testing, xiL testing, open-loop offline perceptions tests, 'Trojan horse' tests, stepped implementation tests, complex tests and so on. All these testing methods have different potentials and disadvantages, for example, 'Trojan horse' tests are functional tests without hazardous effects in serial cars [35].

A further issue of ADS functions is that a strategy for safety confirmation cannot be implemented because a malfunction mechanism cannot be caused by the function but by decisions of the system. Although a test is able to characterise safety-relevant system states, there could be system reactions during automated driving situation where the decisions cannot be affected by the ego-vehicle alone. The actions and reactions of other road users must be anticipated, but a 100 % expectation cannot be ensured. Adequacy here cannot yet be reached on basis of road user reaction models. The system reaction is going to be probabilistic and the decision on

accuracy will become time dependent and ascertainable only in simple situations. The first development of automated function was concentrated on technology goals. But without appropriate validation steps for safety-critical automated functions, the vehicles cannot hope to be established on the consumer market.

16.7 Conclusion

The ISO 26262 standard is intended to be an automotive functional safety standard for handling hazards caused by malfunctioning behaviour of E/E safety-related systems including interaction of these systems. It does not address the nominal performance of E/E systems such as powertrain control or any kind of ADAS. For this reason the ISO standard is also applicable to any level of automated driving. But the complexity of such systems is much higher than today's engineers are used to deal with, because of the high degree of networking functionalities that must be handled. Different kinds of challenges must be considered to realise ADS functions in an adequate manner. Following challenges were discussed in this chapter: Increasing complexity of highly interconnected functions and influence of system attributes, such as availability, reliability, safety and security, must be harmonised.

The concept phase of ISO 26262 becomes more important for ADS functions, because the development of ADS requires the engineering approaches and technologies beyond state of the art.

In particular, influence of the driver in the HARA, definition of safety goals and corresponding attributes for specific levels of ADS (e.g. safe state) as well as the changes of the functional safety concept from fail-safe to fail-operational strategies.

Today, several methods are available to support complex systems but they must be improved for the development of ADS. Possible technologies were discussed to handle the increasing complexity: model-based systems engineering, formal verification by contract-based development, as well as simulation and co-simulation. Which of those methods are adequate and applicable to meet a specific safety-critical demand still has to be defined and argued in the individual safety cases with respect to the specific context.

An enhancement of ISO 26262 that provides guidance for handling such highly complex systems would be useful. In the near future, that kind of application-specific guidance has to be discussed within the working group of ISO 26262 for the upcoming enhancement of the standard. This enhancement should be included in the upcoming revision of the standard which is scheduled to be released by the beginning of 2018. In particular part 3 of the standard needs additional guidance to classify hazardous events during the hazard analysis and risk assessment to determine the ASIL and the system level activities to handle highly networked systems.

16.8 Acknowledgements

The research work has been funded by the European Commission within the EMC^2 under the ARTEMIS JU. The authors acknowledge the financial support of the COMET K2—Competence Centres for Excellent Technologies Programme of the Austrian Federal Ministry for Transport, Innovation and Technology (BMVIT), the Austrian Federal Ministry of Science, Research and Economy (BMWFW), the Austrian Research Promotion Agency (FFG), the Province of Styria and the Styrian Business Promotion Agency (SFG).

References

1. K. Bengler et al., Three Decades of Driver Assistance Systems: Review and Future Perspectives, in *Intelligent Transportation Systems Magazine*, IEEE 6.4, 2014, pp. 6–22
2. International Organization for Standardization, ISO 26262—Road Vehicles—Functional Safety, Part 1–10. ISO/TC 22/SC 32—Electrical and Electronic Components and General System Aspects, 15 Nov 2011
3. European Commission, *CARE Project: Road Safety Evolution in the EU*, Mar 2015, [On-line] http://ec.europa.eu/transport/road_safety/pdf/observatory/historical_evol.pdf. Accessed 12 Oct 2015
4. O. Carstena et al., Vehicle-based studies of driving in the real world: the hard truth? Accid. Anal. Prev. **58**, 162–174 (2013)
5. SAE International, SAE J3016—Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems. J3016-201401, 1 Jan 2014
6. National Highway Traffic Safety Administration (NHTSA), Preliminary Statement of Policy Concerning Automated Vehicles, 30 May 2013, [On-line] http://www.nhtsa.gov/staticfiles/rulemaking/pdf/Automated_Vehicles_Policy.pdf. Accessed 12 Oct 2015
7. Austrian Federal Act, Governing the Liability for a Defective Product (Product Liability Act). 21 Jan 1988, [On-line] www.ris.bka.gv.at/Dokumente/BgblPdf/1988_99_0/1988_99_0.pdf. Accessed 12 Oct 2015
8. International Electrotechnical Commission, *IEC 61508—Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems*, 2nd edn. TC 65/SC 65A—System aspects, 4 Apr 2010
9. R.W.A. Barnard, What is wrong with Reliability Engineering? INCOSE Int. Symp. **18**, 357–365 (2008). doi:10.1002/j.2334-5837.2008.tb00811.x
10. N. Leveson, *Engineering a Safer World: Systems Thinking Applied to Safety*. MIT Press, Jan 2012, [On-line] https://mitpress.mit.edu/books/engineering-safer-world. Accessed 12 Oct 2015
11. H. Butz, Safety and Fault Tolerance in a Complex Human Centred Automation Environment. Innovation Forum Embedded Systems, Munich, 24 Apr 2009, [On-line] http://bicc-net.de/events/innovation-forum-embedded-systems. Accessed 12 Oct 2015
12. H. Butz, Systemkomplexität methodisch erkennen und vermeiden, in *Anforderungsmanagement in der Produktentwicklung*, R. Jochem, K. Landgraf (Hrsg) (Symposion Publishing GmbH, Düsseldorf, 2011), pp. 183–217
13. Stanford Encyclopedia of Philosophy, Emergent Properties, 28 Feb 2012, [On-Line] http://plato.stanford.edu/archives/spr2012/entries/properties-emergent. Accessed 12 Oct 2015
14. D. Campos et al., Egas–collaborative biomedical annotation as a service. Proc. Fourth BioCreative Challenge Evaluation Workshop **1**, 254–259 (2013)

15. IAV GmbH—Ingenieurgesellschaft Auto und Verkehr, Standardized E-Gas Monitoring Concept for Gasoline and Diesel Engine Control Units, Version 6, 22 Sept 2015, [On-Line] https://www.iav.com/en/publications/technical-publications/etc-monitoring-concepts. Accessed 12 Oct 2015
16. International Electrotechnical Commission, IEC 60812—Analysis techniques for system reliability—Procedure for failure mode and effects analysis (FMEA), TC 56—Dependability, 26 Jan 2006
17. International Electrotechnical Commission, IEC 61025—Fault tree analysis (FTA). TC 56—Dependability, 13 Dec 2006
18. S. Friedenthal, A. Moore, S. Rick, *A Practical Guide to SysML: The Systems Modeling Language*, 3rd edn. (Morgan Kaufmann, Amsterdam, 2014)
19. H. Martin et al., Model-based Engineering Workflow for Automotive Safety Concepts. No. 2015-01-0273, SAE Technical Paper, 2015
20. G. Biggs et al., A profile for modelling safety information with design information in SysML. Softw. Syst. Model **15**(1), 147–178 (2014). Springer
21. B. Meyer, Applying 'design by contract'. Comput. IEEE **25**(10), 40–51 (1992). 2015
22. J.-P. Blanquart et al., Towards cross-domains model-based safety process, methods and tools for critical embedded systems: The CESAR approach, in *Computer Safety, Reliability, and Security*, ed. by F. Flammini, S. Bologna, V. Vittorini. Lecture Notes in Computer Science, vol. 6894 (Springer, Berlin, 2011), pp. 57–70
23. A. Baumgart et al., A model-based design methodology with contracts to enhance the development process of safety-critical systems, in *Software Technologies for Embedded and Ubiquitous Systems*, ed. by S.L. Min, R. Pettit, P. Puschner, T. Ungerer. Lecture Notes in Computer Science, vol. 6399 (Springer, Berlin, 2011), pp. 59–70
24. J. Westman et al., Structuring safety requirements in ISO 26262 using contract theory, in *Computer Safety, Reliability, and Security*, ed. by F. Bitsch, J. Guiochet, M. Kaâniche (Springer, Berlin, 2013), pp. 166–177
25. A. Benveniste et al., Contracts for System Design. INRIA, Rapport de recherche RR-8147, Nov 2012, [Online] http://hal.inria.fr/hal-00757488. Accessed 12 Oct 2015
26. M. Fischer et al., Modular and scalable driving simulator hardware and software for the development of future driver assistance and automation systems, in *New Developments in Driving Simulation Design and Experiments*, 2014, pp. 223–229
27. M. Karner, et al., System Level Modeling, Simulation and Verification Workflow for Safety-Critical Automotive Embedded Systems. No. 2014-01-0210, SAE Technical Paper, 2014
28. M. Krammer, H. Martin et al., System Modeling for Integration and Test of Safety-Critical Automotive Embedded Systems. No. 2013-01-0189, SAE Technical Paper, 2013
29. P. Graignic et al., Complex system simulation: Proposition of a MBSE framework for design-analysis integration. Proc. Comput. Sci. **16**, 59–68 (2013)
30. D. Krajzewicz, Traffic simulation with SUMO—Simulation of urban mobility, in *Fundamentals of Traffic Simulation, Series: International Series in Operations Research and Management Science*, ed. by J. Barceló, vol. 145 (Springer, Berlin, 2010)
31. J. Erdmann, Lane-Changing Model in SUMO. German Aerospace Center (2014), [On-Line] http://elib.dlr.de/89233/1/SUMO_Lane_change_model_Template_SUMO2014.pdf. Accessed 12 Oct 2015
32. A. Rousseau et al., Electric Drive Vehicle Development and Evaluation Using System Simulation, in *Proceedings of the 19th IFAC World Congress*, 2014, pp. 7886–7891
33. M. Klauda et al., Automotive Safety und Security aus Sicht eines Zulieferers, 4 Oct 2013, [Online] http://subs.emis.de/LNI/ Proceedings/Proceedings210/13.pdf. Accessed 12 Oct 2015
34. T. M. Gasser, Legal consequences of an increase in vehicle automation. Bundesanstalt für Straßenwesen, 2013, [On-Line] http://bast.opus.hbznrw.de/volltexte/2013/723/pdf/Legal_consequences_of_an_increase_in_vehicle_automation.pdf. Accessed 12 Oct 2015
35. H. Winner, W. Wachenfeld, Absicherung automatischen Fahrens, in *6.FAS-Tagung München*, 29 Nov 2013, [On-Line] http://tubiblio.ulb.tu-darmstadt.de/63810/. Accessed 12 Oct 2015

36. B. Walker Smith, *SAE Levels of Driving Automation*. The Center for Internet and Society at Stanford Law School, 18 Dec 2013, [On-line] http://cyberlaw.stanford.edu/loda. Accessed 12 Oct 2015
37. H. Winner et al., Handbuch Fahrerassistenzsysteme, 3. Auflage. ATZ/MTZ-Fachbuch, (Springer Fachmedien, Berlin, 2015)