UNIVERSITY of York

ASSURING AUTONOMY
INTERNATIONAL PROGRAMME

Assuring the safety of highly automated driving: state-of-the-art and research perspectives

Professor Simon Burton and Dr Richard Hawkins

April 2020

Contents

1. Introduction

The transition from *hands-on* driver assistance (Levels 1–2) [1] to *hands-off* highly automated driving (HAD) (Levels 3–5) requires a number of changes to system safety approaches. For example, a higher level of component availability is required as the system cannot be simply deactivated upon detection of a component hardware fault (fail operational vs. fail safe). Furthermore, at a functional level, an approach to correctly interpreting the current driving situation including environmental conditions and making judgements regarding subsequent actions is required in order to ensure critical driving situations are avoided under all possible circumstances.

The conditions for being acceptably safe with respect to functional safety for passenger vehicles are set by ISO 26262 [2]. Adherence to this standard remains a necessary prerequisite for demonstrating the safety of automated driving in order to ensure a reliable and fault-tolerant implementation of the system with respect to random hardware and systematic failures. Safety of automated driving also requires the satisfaction of a set of safety goals at the vehicle level. Safety goals are top-level safety requirements that define how the system must perform in order that the risk of hazardous events is tolerable. The issue of the insufficiency of the system to meet the safety goals, due to inherent restrictions in sensors, actuators or the inadequacy of the intended function itself, is not addressed [3] by ISO 26262. Extensions to the standard, in particular, the Safety of the Intended Functionality (SOTIF) approach, aim to address these issues, but are currently focused on driver assistance rather than HAD systems [4]. As a result, additional approaches to those already defined by the standards must be developed and the ability of the system to meet its safety goals must be systematically argued based on "first principles" where adherence to existing standards is only one part of the overall argument. An assurance case [5] provides a convincing and valid argument that a set of claims regarding the safety of a system is justified for a given function based on the provision of evidence and a set of assumptions over its operational context.

In this report we explore the challenges involved in assuring the safety of highly automated driving systems with particular focus on the topic of functional insufficiencies (where the function does not meet the safety goals) within an open context. A framework is presented for structuring key elements of the argumentation strategy and a review of state-of-the-art is presented aligned to each of the elements of the framework. The report also uses case studies to highlight where significant research challenges still exist.

The application of established approaches to ensuring functional safety while protecting the system against cyber-physical attacks (security) is a necessary prerequisite for system safety but is outside the scope of this report.

2. Challenges of safety assurance in the open context

In 2017, when Volvo[1] started testing its automated driving systems in Australia for the first time, it encountered a situation the Swedish designers had not necessarily anticipated – kangaroos. Having "trained" the system to accurately recognise and predict the path of large mammals such as deer and elk crossing the road ahead, the movements of the marsupials responsible for 90% of the animal/vehicle collisions in Australia had the system stumped. Since then, there have been other incidents[2] of automated driving vehicles misinterpreting their surroundings with fatal consequences for the vehicle occupants and pedestrians.

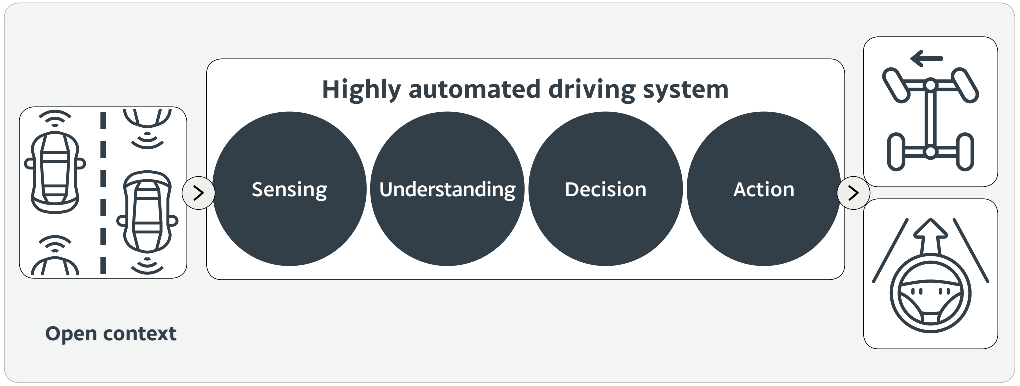


Figure 1: Functional components of a highly automated driving system

Figure 1 summarises the functional components constituting a highly automated driving system. *Sensing* components may consist of various direct sensor channels such as RADAR, LIDAR and video cameras. *Understanding* components involve interpreting the current driving situation based on the sensing inputs and can also include indirect contextual information, for example, from digital maps and vehicle-to-infrastructure systems. This includes the processing of raw sensor data to provide a logical perspective of the current situation including vehicle position and trajectory as well as the type, position and trajectories of other traffic participants. *Decision* components react based on a set of driving goals (e.g. drive from A to B) and an interpretation of the current scene to calculate the required driving strategies. *Action* components are responsible for executing the driving strategy via the set of vehicle actuators (brakes, engine, steering wheel, etc.).

We define an open context system as a system that operates within an environment which cannot be fully specified at design time, either due to the inherent complexity and unpredictability of the environment or due to the manner in which the environment evolves over time. The open context provides a key challenge to the safety validation and assurance of highly automated driving systems. In particular, safety assurance must address the issue of demonstrating that adequate system performance is guaranteed within an environment that cannot be completely specified and continuously changes during operation of the system.

[1] https://www.theguardian.com/technology/2017/jul/01/volvo-admits-its-self-driving-cars-are-confused-by-kangaroos

[2] https://www.theguardian.com/technology/2017/sep/12/tesla-crash-joshua-brown-safety-self-driving-cars

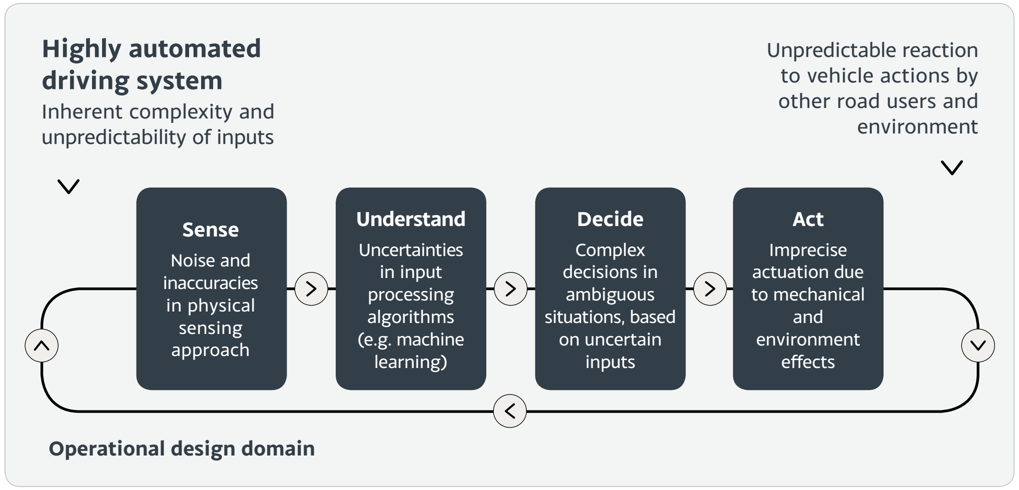

Highly automated driving system
Inherent complexity and unpredictability of inputs
Unpredictable reaction to vehicle actions by other road users and environment
Sense
Noise and inaccuracies in physical sensing approach
Understand
Uncertainties in input processing algorithms (e.g. machine learning)
Decide
Complex decisions in ambiguous situations, based on uncertain inputs
Act
Imprecise actuation due to mechanical and environment effects
Operational design domain

Figure 2: Sources of complexity and uncertainty in highly automated driving systems

Figure 2 summarises the sources of uncertainty and complexity in systems resulting from the open context. Uncertainty is inherent due to the complexity and unpredictability of the operational domain in which the system operates. For example, other road users may often behave in unexpected ways, the same object may look different in different weather conditions, or road signs may be damaged or missing. This unpredictable domain is then observed by the system via channels that are also inherently imperfect due to technical limitations of the sensing approach itself [6]. Thus, the understanding and decision-making components of the system are presented with noisy, incomplete data about the current situation. This uncertainty is typically counteracted by using multiple sensing channels (that may present conflicting information) and algorithms that make use of heuristics or machine learning functions to interpret the data. However, these algorithms are themselves inherently imprecise and introduce an additional level of uncertainty. For example, object classifications may only be provided with a particular level of confidence. Based on the resulting partial understanding of the inherently complex environment, "decision" functions are required to implement a driving strategy capable of safely navigating the vehicle to its ultimate destination.

The unpredictable nature of the impact of the vehicle's actions on its environment, for example in terms of the reactions of other drivers and road users, closes the cycle to the complex environment to be interpreted by the vehicle. In other words, a function operating according to its specification running on failure-free hardware could still cause serious safety hazards if the complexity and uncertainties inherent in the driving tasks are not adequately managed.

The dominant challenges facing the safety assurance of highly automated driving systems are therefore the derivation and validation of system safety goals that are sufficient to control the risk of hazardous events and the demonstration of their fulfilment under all feasible situations that may be encountered by the vehicle during its operation. These need to be achieved despite the complexity and uncertainty inherent in the domain, and in the sensing and understanding/decision algorithms.

3. A framework for safety assurance of highly automated driving

3.1 How safe is safe enough?

More than 90% of accidents on the road can be attributed to human error [7] and automated driving systems have the potential for making roads significantly safer by restricting the impact of potentially inattentive and unreliable human drivers.

However, these systems also introduce new classes of risks. By transferring the decision function from the driver to the machine they also bring up a number of ethical questions. A number of factors will impact on whether or not society at large will place their trust in these systems. The level of tolerable residual risk associated with the introduction of the new technology will also be seen within the context of the perceived safety benefits on the function itself. It is our task as automotive safety engineers to deliver safety arguments for the system that are convincing, objective and sound, and that can be understood and accepted by not only governing authorities but also the public at large.

The starting point of any safety argument is some definition of the safety claim that is being made. In other words, how "safe" we argue the system to be. In 2016, the German ministry of transport and digital infrastructure commissioned a report[3] into ethical considerations of automated driving. One of the recommendations of the report was that it must be shown that the automated driving systems perform, on average, better than a human driver in terms of avoiding or mitigating hazardous situations, although in some cases it may be acceptable that the performance is slightly worse than a human so long as an overall "positive risk balance" is achieved.

A related approach based on the definition in French *"Globalement au moins aussi bon"*, or GAMAB for short, refers to the principle that any new system must be at least as good any previous system it replaces. Although superficially this could be used to argue the risk equivalence to average human drivers, it could also easily well be argued that automated driving systems are not a replacement of the human driver but are instead a fundamentally new technology. In addition, this is also a problematic standpoint from a product liability perspective, as every accident could be examined individually regarding whether or not better engineering and management practices could have prevented the accident from occurring.

The ethics commission report however did not only focus on positive risk balance as a measure of an ethically acceptable level of safety. It also places emphasis on the application of a proactive driving behaviour, avoidance of accidents as much as "practically possible" and the avoidance of discrimination on the basis of any person-related characteristics. The principle of ALARP (as low as reasonably practicable), or variants thereof, are also often used in the regulation of safety-critical systems. The ALARP approach to risk assessment involves demonstrating that the cost involved in further reducing the risk would be disproportionate to the benefit gained. These judgements are typically made not only on the basis of quantitative assessments but also on an understanding of good engineering practice and existing standards. If it could be argued therefore that applying existing standards and good engineering practice could result in significantly better performance than an average human driver, then a direct comparison to current accident statistics may not be sufficient. Such a qualitative approach to arguing safety nevertheless requires some definition of what it means to drive "safely".

[3] Ethics Commission Report Automated and Connected Driving (in German), Bundesministerium für Verkehr und digitale Infrastruktur, 2017

This could include some safe driving principles such as the following:

- Maintain a proactive driving style:
    - Employ an anticipatory and predictable driving style – avoid hazardous scenarios.
    - Maintain legal compliance.
- Ensure a reactive driving style:
    - In case of violations of laws and regulations by other road users, the system reconstitutes its legal compliance.
    - If this is not possible, or other road users, animals or objects cause a hazard, prevent a possible accident or mitigate the damage.

The evidence used to support how well these claims are met by the system will include quantitative statements based on statistical analysis that reinforce our confidence in our arguments and help to illustrate the level of residual risk achieved. In addition though, a broad range of evidence should be presented based on engineering rationale as part of the system design as well as verification and validation activities.

The assurance case will provide a structured argument that the HAD function achieves safe behaviour for all conditions that meet the set of assumptions describing the target application domain. The assurance case must justify that the level of risk associated with the HAD function is acceptable. This will rely not only on technical but also societal, ethical and legal considerations. The assurance case argument will be based on the capability of the chosen system architecture itself to minimise the risk associated with insufficiencies due to domain complexity (aleatoric uncertainty), sensing errors and component insufficiencies (epistemic uncertainty) [8]. This argument shall be supported by a diverse set of verification and validation evidence that support claims of the functional sufficiency of the HAD function for its determined context as well as the validity of the set of assumptions used to delimit the open context in terms of their relevance to the target domain.

3.2 Summary of the framework

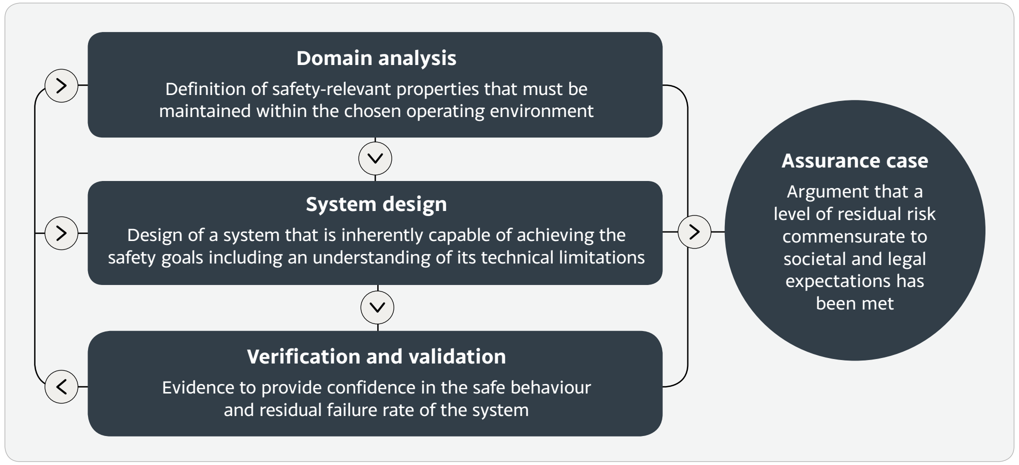

Domain analysis
Definition of safety-relevant properties that must be maintained within the chosen operating environment
System design
Design of a system that is inherently capable of achieving the safety goals including an understanding of its technical limitations
Verification and validation
Evidence to provide confidence in the safe behaviour and residual failure rate of the system
Assurance case
Argument that a level of residual risk commensurate to societal and legal expectations has been met

Figure 3: A framework for the safety assurance of highly automated driving

This section summarises a framework in which the main elements of such an argument are presented. The overall approach is summarised in **Figure 3**. At its core is a top-level definition of an acceptable level of safety and the overall aim is to develop an assurance case that argues that the level of residual risk associated with the system is commensurate to societal and legal expectations. The activities within the framework are continuously iterated as the use of the systems in the field lead to a better understanding of the operational design domain as well as the system's inherent technical limitations and subsequent improvements.

A systematic *Domain analysis* forms the basis of an understanding of the environment in which the system should operate. Classes of scenarios in which the system should operate are defined and analysed to identify dominant properties of the environment relevant to the safe operation of the system. A set of high-level safety goals for the system are then defined based on this domain analysis. This includes a systematic hazard and risk analysis involving not only a consideration of a failure in the function of the ego-vehicle, but also a systemic view of intrinsically hazardous conditions within the interaction between the ego-vehicle and its environment that must be avoided. Since the safety goals are defined based on this "determined context" for the vehicle operation, the validity of the safety assurance argument that is produced is therefore also restricted to this "determined context", as defined by the set of scenario classes and identified properties. The resulting scope is known as the operational design domain. Typical classes of scenario included in the analysis could include, for example, motorway driving at speed, and in heavy traffic, and relevant environmental properties may include weather and lighting conditions, road surface, as well as the behaviour and type of other traffic participants. Ensuring that all "relevant" classes of scenario and their dominant safety properties have been identified is a key task of the validation of such systems. The top-level safety goals of the system are also defined within the context of societal and legal expectations on safe driving behaviour. The rigorous analysis of such contextual requirements therefore also forms an important component on delimiting the scope of the safety assurance argument.

System design involves applying an iterative approach to refining the safety goals in coordination with system design decisions. At each level of refinement, assumptions made in the design are explicitly stated and analysed, so that they may also be later validated. Further iterations of the domain analysis may also be required, for example, to identify particular properties of the environment relevant to the operation of a visual pedestrian recognition system. This may identify assumptions on other components of the technical system architecture, for example, assumptions about the depth of focus and resolution of the camera that provides the input to the pedestrian recognition function. A functional and technical system design should be derived that is capable of achieving the safety goals, even in the presence of inevitable insufficiencies and faults within individual components.

Component insufficiencies may be, for example, limitations of particular sensor components in poor light conditions. Meeting the safety goals requires a means of determining and specifying which insufficiencies of individual components are acceptable in all parts of the "sense, understand, decide, act" chain. In addition, a means is required to determine how such insufficiencies either propagate through, or are minimised by, the chosen system design (e.g. through a combination of diverse sensor channels with non-overlapping insufficiencies). A compositional approach to analysing the safety of the system can be used to manage the emerging complexity and allow for re-use of standardised parts. This should involve a precise (preferably formal) specification of the assumptions each component requires to be fulfilled on its input interfaces in order to guarantee a particular performance on its output interfaces.

In such a way, pre-qualified components can be combined to form a system architecture that can be analysed for its robustness in satisfying the safety goals. The system architecture must be robust both to functional insufficiencies of the individual components themselves, as well as to the uncertainties inherent in the domain or sensing channels.

The evidence requirements for *Verification and validation* will dramatically increase for HAD systems in comparison to previous vehicle functions. As well as the need to demonstrate high levels of availability of the function compared to the validation of the relatively simple diagnostic and safe state mechanisms of the past, a number of other specific challenges must be solved. At the component level, a high level of confidence is required that the assumes/guarantees contracts are fulfilled for each component, taking account of functional insufficiencies. This could include, for example, the formal verification of a decision algorithm, or a demonstration of the perception accuracy of a particular sensor channel under the determined set of scenario conditions. A particular challenge is to verify that machine learning algorithms fulfil their specification with an adequate level of accuracy.

In addition, evidence must be provided to validate that the scenario classes and associated properties provide adequate coverage of the real scenarios encountered in the target domain. Such validation is required at the system level, but also to confirm the assumptions made for each component (e.g. size, shape and behaviour parameters of pedestrians that were used to select training data for a machine learning-based pedestrian recognition function). Due to the complexity of the domain and the impractical number of driving hours that would be required to provide statistically relevant test results [9], sources of validation other than standard vehicle-level system tests will be required. These may include simulation as well as the statistical analysis of large amounts of data captured in the field (e.g. during the operation of previous generations of a driving function). Both the simulation and field data analysis will need to be based on the same semantic data model as was used to describe the scenario classes and properties during the domain analysis. In this way, the validation activities can be actively used to refine the domain model and confirm the appropriateness of the assumptions made to restrict the scope of the open context relevant for the safety assurance.

Lastly, in order to reason about the applicability of the non-exhaustive verification and validation results across the entire scope of the input space for the operation of the vehicle, statistical extrapolation techniques will be required that can predict residual failure rates based on the combination of analysis, simulation, test and field monitoring evidence. In particular it must be demonstrated that coverage of rare but nevertheless critical events has been achieved.

4. Application of the framework

4.1 Introduction

Within the last few years, many companies have begun development on HAD vehicles and have demonstrated prototypes under controlled conditions. The first fatal accidents caused by HAD enabled (test) vehicles have demonstrated the gap between prototypical functions and systems that can be released for unconstrained and safe operation within a given operational design domain (ODD). The capabilities required to release such systems for safety will need to be developed and introduced over time, thereby limiting the speed at which the systems can be introduced into the market. This is due to a number of factors:

- The need to develop competencies in system safety methodologies for open context autonomous systems within the automotive industry, including a significantly strong foundation in basic systems engineering principles;
- The need to resolve a number of open research questions that are required for a convincing safety assurance case;
- Technological development of the tool chains and infrastructure required for design, simulation and test of the systems;
- The efficacy of the methods referenced within the safety assurance case must be confirmed for realistic examples (e.g. ability of innovative testing techniques to demonstrate the robustness of machine learning-based perception functions);
- Pre-validated system components with known functional and performance properties must be developed for re-use that can be applied to successively more sophisticated functions without requiring a complete system re-validation.

The industrialisation of the assurance approaches for large-scale series development and release of such systems will require major changes across the industry. An iterative approach to developing these capabilities and confirming their effectiveness is therefore recommended. Example phases of increasing capabilities are shown in **Figure 4**.

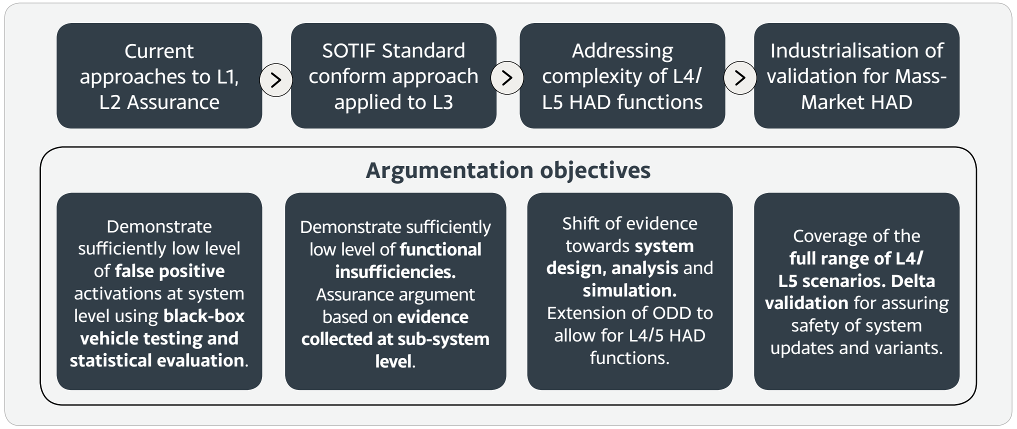

Current approaches to L1, L2 Assurance
SOTIF Standard conform approach applied to L3
Addressing complexity of L4/ L5 HAD functions
Industrialisation of validation for Mass-Market HAD
Argumentation objectives
Demonstrate sufficiently low level of false positive activations at system level using black-box vehicle testing and statistical evaluation.
Demonstrate sufficiently low level of functional insufficiencies. Assurance argument based on evidence collected at sub-system level.
Shift of evidence towards system design, analysis and simulation. Extension of ODD to allow for L4/5 HAD functions.
Coverage of the full range of L4/ L5 scenarios. Delta validation for assuring safety of system updates and variants.

Figure 4: Phases of capability improvement in safety assurance of HAD

In this section, the assurance framework is applied to two use cases from the HAD domain with differing complexity in order to demonstrate how various of the components of the framework fit together and where challenges still remain.

4.2 Level 3 highway chauffeur

The first use case will address assurance concepts for an SAE Level 3 [1] Conditional Driving Automation System that takes over control of the vehicle while driving on highways. During this time, the driver can direct their attention to other pursuits while the system takes control of the Dynamic Driving Task (DDT) and Object and Event Detection and Response (OEDR). The driver must be available to take over control in the case of a system failure or when the boundary of the ODD is met. This scenario can be seen as an extension of existing SAE Level 2 systems, for example, for longitudinal and lateral control. However, due to the fact that the driver no longer permanently monitors the system and environment, the assurance argument is not just restricted to arguing the absence of false positive events (e.g. harsh braking when not necessary) but must also consider false negative events (e.g. all objects in the trajectory of the vehicle must be detected) and performance requirements (e.g. geometric accuracy in the detection of objects and their classification). The following is a description of a potential assurance strategy for such a system based on the core components of the framework.

Domain analysis

The highway chauffeur function is limited to a restricted ODD which is described in terms of classes of scenarios according to the PEGASUS methodology (www.pegasusprojekt.de) and grouped according to a set of well-defined use cases (e.g. handover from driver to system, continuous driving in lane, overtaking, obstacle avoidance, handover from system to driver). The scenarios are defined according to the following five layers:

- Street topography;
- Traffic infrastructure (signs, traffic guidance);
- Temporal modifications of street topography and traffic infrastructure;
- Traffic participants and their interactions;
- Environmental conditions and their interactions with other properties of the scenario.

The description of the scenarios includes explicit restrictions of the ODD, for example geo-fencing to certain roads where there is complete and up-to-date information available regarding topology and infrastructure, weather conditions in which the sensor set is known to exhibit an adequate level of performance and the absence of roadworks or other traffic anomalies. To increase readability and analysability, the scenarios could be specified using Traffic Sequence Charts [10] or in the OpenScenario language (www.vires.com). The definition of the ODD leads to a set of assumptions about the environment and requirements on the detection of the ODD boundaries which must be captured in the system-level requirements specification. A semantic knowledge-based model, based on ontologies, is required in order to provide traceability between the ODD and the system specification, simulation, tests and field data.

Based on the scenario-driven description of the ODD and an understanding of the proposed function and system architecture including sensing channels and interfaces to other systems, a hazard and risk analysis for SOTIF is performed. For all identified hazardous events, acceptance criteria are defined (e.g. validation targets for the number of observed failures over time). Thus, a set of SOTIF-related safety goals are defined for the system including a concretisation of parameters for the specific vehicle, such as safe braking distance (or specification of dynamic parameters dependent on properties of the current scene) and acceptance targets.

The system specification shall include, in addition to a reference to the ODD, a description of the intended functionality and the derived safety goals. There is a risk in a scenario-driven approach that safety goals are derived with undue focus on known triggering events and lack of coverage of unknown triggering events. Due to the fact that the ODD is restricted to situations in which many of the system components (e.g. radar-based object detection) would have typically been previously validated for Level 1 and Level 2 systems, the validation of the requirements' completeness and probability of unknown triggering events can be performed predominantly in vehicle-based testing at a targeted sub-system level or system level (e.g. in the presence of a safety driver). Additionally, an analysis of other road users' expectations of system behaviour, traffic flow simulation and crash databases can provide additional sources of information during requirements elicitation.

Design for assurance

The system specification is decomposed according to the "sense, understand, decide, act" functional structure and the associated requirements are allocated to components in the technical system architecture. Thereby assumptions about the ODD or other system components that are necessary in order for each component to reach its performance targets are documented and validated against the respective component specifications and known properties. Due to the use of previously validated components with well-known properties for the restricted ODD, a formal specification and analysis of all properties of interest may not be required if alternative validation evidence can be presented to argue that the ODD-specific limitations of the components are well understood.

For the functional safety and SOTIF properties related to the safety goals, a safety analysis of the system architecture – fault tree analyses (FTA) and failure mode and effects analyses (FMEA) – should be extended to consider SOTIF properties. Deductive analyses such as FTA depend on all triggering events to be known to determine whether or not the validation target is met for that given set of basic events. Inductive analyses such as FMEA are better suited at detecting previously unidentified potential triggering events. These types of analyses are therefore to be seen as complementary rather than alternatives. The safety analyses will lead to the identification of additional system design measures (e.g. redundant sensing channels) or improved algorithmic capabilities to reduce the probability of the identified triggering events. FTAs on the other hand have the advantage of analysing the propagation of triggering events through the system.

Verification and validation

A verification and validation strategy shall be defined with regard to the risk of potentially hazardous triggering events and shall cover the ability of the sensors and sensor processing algorithms to model the environment, the ability of the decision algorithms to handle both known and unknown situations and make appropriate decisions, as well as the ability of the human-machine interface to prevent foreseeable misuse, and the manageability of the handover scenario by the driver [4].

The objective of the verification of the sensing and understanding algorithms is to determine the performance limits of the perception functions of the system and to confirm that these meet the performance requirements allocated to the sensing path. These tests, performed under lab and controlled test track conditions, shall systematically verify the sensing performance of the system across all scenarios and environmental conditions defined in the ODD.

If during the course of testing, previously unknown triggering events are discovered, then the domain analysis shall be revisited to determine whether or not the domain model shall be extended to explicitly include or exclude such conditions. Over time, an existing catalogue of conditions and concrete scenarios (e.g. road structures specific to a particular location) will be continuously extended so that it can be used as a regression test for verifying against known triggering events.

Simulation is a viable means to verify the decision functions (e.g. manoeuvre planning): in particular, to cover many variations of the scenario classes defined in the ODD and to verify the behaviour of the system in rare, but otherwise critical, situations that are either too complex or too hazardous to reproduce in the real world. The use of simulation in combination with a formalisation of safety goals also allows for search-based testing to be performed to automatically determine scenarios in which the system fails to maintain the safety goals. These scenarios will then be analysed to determine whether or not they represent a gap in the ODD or system specification. During simulation, typically a perfect sensing model is assumed. However, fault injection on the simulated sensor channels (e.g. sporadic false negatives) could also be used to determine the robustness of the system against insufficiencies in the sensors.

Once sufficient confidence in the performance of the individual sub-systems has been achieved, then system verification within the target vehicle can begin, initially on closed test tracks and later on the open road. The scenario-based approach to defining the ODD provides a suitable test coverage criterion, however it is anticipated that certain combinations of scenarios and environmental conditions may only be reached under contrived conditions (e.g. test track). System verification activities are focused on determining the performance of the system with respect to known triggering events.

System validation activities have the objective of determining the risk of unknown triggering events leading to hazardous situations. This will again involve achieving a suitable coverage of the scenarios defined in the ODD but also must include a statistically relevant number of driving miles with the ability to uncover situations not yet specified. In order to achieve the number of driving miles required in a feasible time frame, various strategies can be applied in parallel. Test fleets with dedicated safety drivers are essential, but do not scale. Additional validation information can be achieved through collecting data from vehicles with an equivalent sensor set with the Level 3 function operating in silent mode. The analysis of the field data and its correlation with the definition of the ODD requires the use of the same semantic model as applied in specifying the classes of scenarios and environmental conditions. This then allows for triggering events discovered during validation to be analysed within the context of the current understanding of the ODD and for targeted analyses to be performed (e.g. search for occurrences of a particular scenario under specific weather conditions in the recorded field data).

In addition, extreme value theory should be applied to recorded sensor data and internal state information to provide a statistical distribution of measurable events that can be used to deduce the probability of safety goals being violated. This in turn can be used to define test stopping criteria by calculating the set of data required in order to make such statistical extrapolations with sufficient confidence. Representative scenarios discovered during field testing (e.g. as yet unknown triggering events) should be recorded in a manner that they can replayed within a simulation environment in order to determine root-causes of the issues as well as to be able to use them as regression tests for future systems.

Assurance case

In essence, the assurance case must argue the absence of unreasonable risk due to functional insufficiencies through the following perspectives:

1. The specified behaviour is an accurate representation of the intended behaviour (addressing semantic gap in the specification).

2. The implemented behaviour corresponds to the specified behaviour, including in achieving the performance targets. This corresponds to verification of the implementation.

3. The implemented behaviour corresponds to the intended behaviour. This is equivalent to a validation of the implementation and follows naturally from goals 1 and 2, if these could be argued with full certainty. This is however not realistic for all but the most trivial systems and ODDs.

4. The assurance case is valid only for the restricted ODD. Therefore, a strong argument also needs to be made that the HAD function is only active within the ODD and that the system can accurately detect when the boundaries of the ODD have been met.

5. The assurance case must argue that the handover between system and driver and vice-versa is safe.

The assurance case structure consists of qualitative arguments that system insufficiencies are well understood, and unknown triggering events are minimised (SOTIF approach). This argument is supported by the use of components with known performance properties from their application to previous SAE Level 1 and SAE Level 2 systems operating in the same domain, as well as systematic approaches to simulation and test. Quantitative arguments are made when defining test stopping criteria and when using statistical analysis to support the validation arguments, for example based on extreme value theory. An overall quantitative statement regarding the residual failure rate associated with functional insufficiencies is not made.

Open research challenges

The overall assurance approach for the Safety of the Intended Function of the Level 3 highway chauffeur is summarised in **Figure 5**.

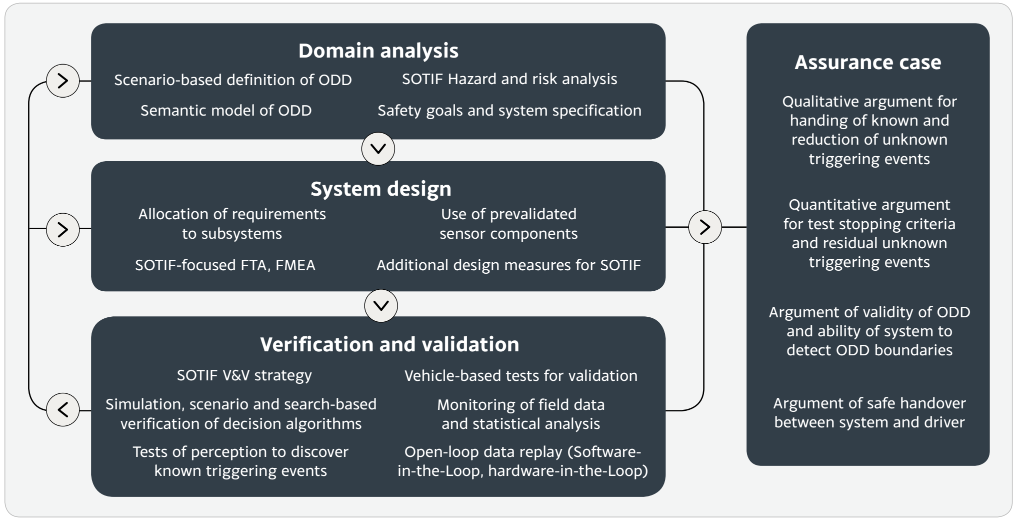

Domain analysis
Scenario-based definition of ODD
SOTIF Hazard and risk analysis
Semantic model of ODD
Safety goals and system specification
System design
Allocation of requirements to subsystems
Use of prevalidated sensor components
SOTIF-focused FTA, FMEA
Additional design measures for SOTIF
Verification and validation
SOTIF V&V strategy
Vehicle-based tests for validation
Simulation, scenario and search-based verification of decision algorithms
Monitoring of field data and statistical analysis
Tests of perception to discover known triggering events
Open-loop data replay (Software-in-the-Loop, hardware-in-the-Loop)
Assurance case
Qualitative argument for handing of known and reduction of unknown triggering events
Quantitative argument for test stopping criteria and residual unknown triggering events
Argument of validity of ODD and ability of system to detect ODD boundaries
Argument of safe handover between system and driver

Figure 5: Assurance strategy for Level 3 highway chauffeur (SOTIF focus)

Based on the state-of-the-art review conducted within this report (see Sections 5, 6, and 7), the areas of the strategy with the most significant remaining open research challenges can be summarised as follows:

- Development of a common semantic model of the ODD that can be used within the system specification, simulation, test and field data analysis. Ideally this semantic model will become an industry standard in combination with the classes of scenario.
- Extensions of systems safety analysis techniques (such as FTA, FMEA, etc.) are required in order to model the effects of propagations of component-level insufficiencies throughout the system. These methods require as prerequisites the availability of information regarding potential insufficiencies at the component level. This information could be provided, for example through systematic and extensive testing or experience in the use of the same components within the context of other functions.
- Testing approaches for determining functional insufficiencies (discovery of known triggering events) at the component level and especially for sensors and perception algorithms are required. In particular, an argument for the absence of false negatives must be made during component and system tests which poses particular challenges in the generation of relevant test data and associated ground truth. The effectiveness of the testing approaches in detecting insufficiencies and arguing the absence of unknown triggering events must be demonstrated in order for them to provide a contribution to the system safety case. This must be evaluated within the context of extreme value theory to estimate the probability of residual failures. Further research is required in order to determine appropriate data and thresholds in order for the extreme value theory approach to yield robust results.
- Due to the fact that the assurance case is predicated upon a restricted ODD, the ability of the system to accurately detect the ODD boundary and respond appropriately must be rigorously argued. This challenge is closely related to the ability to ensure a safe handover to the driver upon reaching the boundary of the ODD. This requires ensuring sufficient time for the driver to revert their attention to the driving task and may require additional driver awareness monitoring systems.

4.3 Level 4 urban automated driving

In this section, a second application of automated driving is studied with significantly more challenges for safety assurance. SAE Level 4 High Driving Automation systems for urban automated driving require the system to take over dynamic driving tasks (DDT) and Object and Event Detection and Response (OEDR) in highly complex environments as well as providing fall-back functions in the case of system malfunctions. To illustrate the challenges involved in the safe implementation of urban automated driving it is useful to analyse the factors that led to a fatal accident involving a prototype HAD vehicle developed by Uber and discussed in detail in the article linked in this footnote[4].

The overall perspective on Uber's initial assurance strategy can be summarised by the following quote from the article:

"... ATG, like everyone in the self-driving car industry, believed that the more miles a car drove itself without help from a human, the smarter it was. But the whole industry now realized this is an overly simplistic way to measure how well a car drives."

[4] https://www.businessinsider.de/sources-describe-questionable-decisions-and-dysfunction-inside-ubers-self-driving-unit-before-one-of-its-cars-killed-a-pedestrian-2018-10?r=US&IR=T

A more detailed analysis of the events leading to the accident shows that in trying to resolve the conflict of reducing the number of false positive events to create a smooth driving experience, a subsequent increase in false negative rates was deprioritised or ignored. In addition, there were known insufficiencies in the perception capabilities of the system, particularly involving near-range sensing and the detection of pedestrians and generation of ghost objects caused by effects such as the shadows of tree branches on the road. This suggests that a systematic design and analysis of the overall sensing approach was lacking, and information gathered about insufficiencies in the sensing were not adequately addressed at the system level. In addition, the built-in emergency braking system of the host vehicle was de-activated based on the assumption that the safety driver would take over in the event of a real threat. Furthermore, simulation-based verification was found to be inadequate, leading to immature functionality to be tested on the road within a public environment.

In addition to these technical challenges, a number of further organisational and procedural weaknesses were highlighted including an incentives system that caused engineers and safety drivers to prioritise smooth driving over safety-related interventions. The various contributing factors leading to the Uber accident serve to highlight the challenges of urban automated driving (in addition to those already discussed for the Level 3 scenario).

- The vehicles will operate in a far more complex and unpredictable “crowded” environment. This will include a higher density of objects that must be correctly identified and classified at much shorter distances in “distracting” surroundings.
- The increased level of perception ability will (at least, based on current sensor technology) rely on a cluster of numerous sensors of various modes (radar, LIDAR, camera). Understanding the limitations of each of these sensor types, their interactions and potential common cause triggering events, will be essential to arguing a sufficient performance.
- Due to the complexity and unpredictable nature of the ODD, a scenario-driven approach is unlikely to provide the level of completeness required without an infeasible explosion in the level of detail in the scenario specification.
- There will be an increasing number of potential scenarios in which the safe intended behaviour cannot be precisely defined. A clear set of ethical and legal principles will be required to guide the specification of system behaviour in these cases.
- Machine learning will play an increasingly significant role in the perception algorithms in order to extract structure from the complex environment (e.g. convolutional neural networks for pedestrian detection and pose estimation) as well as for decision making (e.g. reinforcement learning for trajectory planning). Machine learning poses particular problems in demonstrating the sufficiency of the system to react under all conditions.
- The initial operation of Level 4 automated driving services within specific cities with the close cooperation of city authorities and homologation authorities also opens up the possibility of making use of city and traffic infrastructure to increase the reliability of the HAD function. This leads to allocation of safety requirements on the infrastructure outside the scope of the vehicle manufacturers and existing safety standards.

Domain analysis

Although still restricted by a given ODD that may include certain roads within a given city or certain weather conditions, the environment in which a Level 4 urban automated vehicle will operate will be significantly more diverse and complex than that for the Level 3 highway chauffeur. In particular, the close interaction between the vehicle and various types of other traffic participants such as pedestrians, cyclists, cars, buses, trams will inevitably lead to complex scenarios in which the safe intended behaviour cannot be easily defined, leading to a new type of semantic gap in the specification, defined in [11], as the moral responsibility gap. For this reason, a set of requirements that represent the legal, societal and ethical expectations on the behaviour of the vehicle must be defined and a clear definition of the boundaries of the vehicle behaviour with respect to these constraints is required (i.e. the vehicle should not be expected to make "moral" judgements[5]).

When capturing the ODD for urban automated driving, even for a restricted ODD, the number of possible properties that could impact on the safe operation will be huge, in particular when taking into account dynamic effects, such as reactions of pedestrians and other road users to the ego-vehicle's behaviour. When analysing the domain, an enumeration of all possible critical scenarios and properties will quickly lead to unmanageable complexity explosion with the high chance of incompleteness with respect to critical corner cases. This will require abstractions to be found that correspond to phenomenological properties of the domain critical to the safety of the driving function. The abstracted properties will then need to be used to define the ODD without the need to enumerate all possible scenarios. Certain properties will be specific to the sensor set, for example reflectivity and density of objects may be relevant for radar-based but not for camera-based perception systems.

These phenomenological abstractions may be discovered via expert analysis, extensive tests, machine learning analysis of field data [12], as well as simulation including physical simulation of sensor properties. The set of classes chosen to represent the ODD will, of course, need to be thoroughly and continuously validated in the field as they are likely to evolve over time.

The above-mentioned level of domain analysis will lead to a more detailed definition of safety goals that also need to take into account the societal, legal and ethical constraints, as well as the phenomenological domain analysis. These safety goals will need to be refined and specified for specific classes of scenarios (e.g. pedestrian crossings) with reference to the semantic ontologies of road topologies, objects and environmental conditions. It is anticipated that the complexity of such safety goal definitions will need to go far beyond that proposed in [13] and will require a concerted industry-wide initiative to perform this analysis and form a consensus for acceptable safety goals for urban environments. In addition, a set of suitable technical performance indicators must be defined in order to measure progress against these safety goals to determine the sufficiency of a system to operate safely within the defined domain.

System design

The increased complexity of the requirements derived from the safety goals and allocation to individual components will require a higher level of rigour and formality in the design process in order to be able to argue that the design is sufficient for the safety goals. Requirements on perception algorithms will vary according to the scenario, for example, while turning at pedestrian crossings, the near 360 degree range sensing is most critical, while when driving at higher speeds the ability to detect and predict pedestrian movements from a distance is required.

[5] http://theconversation.com/self-driving-cars-why-we-cant-expect-them-to-be-moral-108299

This may result in potentially conflicting optimisation goals that must be resolved during the design process. Based on the performance of current sensor technologies and perception algorithms, these requirements will lead to a large number of environment sensors (approximately 40 for vehicles currently in development). In order to argue the adequacy of this complex sensing system to detect all possible conditions of the ODD, a structured decomposition of the perception requirements based on the domain analysis is required and allocated to scenarios, together with spatial zones of the surroundings. In refining the requirements through the system design, the assumptions regarding the environment as well as the capabilities of system components must be clearly identified. The importance of design-based arguments based on systematically derived phenomenological domain attributes will increase for this class of systems, as testing alone can no longer provide a strong verification argument, due to the inherent uncertainty and complexity of the environment.

Model-based systems engineering approaches will be required in order to manage the complexity of verifying that the system design meets the requirements. In particular, design-by-contract approaches to specifying system components, including all assumptions about possible uncertainties on their inputs, insufficiencies in their function and failure modes in their execution, will provide the formal means of reasoning about the robustness of the system as a whole. Physical simulation of sensors, detailed component analysis, verification of performance requirements and fault injection tests will provide the information regarding the basic fault models at the component level. The model-based specification of the system design will then allow for an automated system-level analysis of the safety goals and multi-criteria optimisation (e.g. trade-off between robust coverage of perception requirements and the number of sensors).

The application of design-by-contract approaches becomes particularly relevant when integrating machine learning-based components into the system. For perception functions, this may involve so-called early fusion architectures (the machine learning function takes a number of diverse sensors as inputs) or late fusion (each sensor input is processed separately and semantic information is then provided to a separate function to perform cross-checking and plausibility checks to develop an understanding of the scene). A clear definition of the assumptions about the operating domain and system components is required, as is a definition of the guarantees that can be made by the machine learning function, including its level of robustness in terms of detection accuracy. This assumes/guarantees relation must then be confirmed by targeted analysis and verification applied to both the trained function and the training data itself.

For machine learning functions in the decision and planning components of the system, it is unclear whether or not behavioural constraints can be defined in enough detail in order to ensure that the safety goals are always met, in particular in the case where reinforcement learning is used. Instead, a continuous observation of the environment to ensure that assumptions are met despite an evolving domain and the application of constrained learning techniques will be required. Multiple levels of system monitoring will nevertheless be required, in order to continuously monitor the boundaries of the ODD and system performance and to determine when an automated fall back to a safe operating mode is required.

Verification and validation

The verification and validation strategy for Level 4 urban automated driving must address the issues of increasing domain and system complexity including many situations that cannot be predicted during design time or safely tested in real systems.

This requires that an increasing emphasis is placed on simulation, including at the perception level, as well as ever more systematic approaches to arguing the completeness of validation tests in the vehicle. Driving both of these will be the phenomenological attributes and semantic ontologies in addition to the set of scenario classes defined for the domain. The application of simulation and search-based testing to identify realistic corner cases for the perception will require accurate models of the sensors and environment and a clear specification of the performance properties. In particular, physical simulation of the sensors as well as combined sensing, perception, decision-level system simulation will require significant computing resources involving much capital investment.

With an increased reliance on simulation to gather assurance evidence, the integrity of the simulation and test environments as well as the transferability of the results also needs to be called into question. Simulation environments currently under development are still at an early stage, or reliant on open source development processes; a concerted effort will therefore be required in order to provide a qualified simulation and test infrastructure for automated driving.

Vehicle-based tests will remain essential for validating the assumptions made during domain analysis, system design and verification. In order to argue a coverage of not only the huge range of scenario classes but also the variations of phenomenological properties, the ability to directly trace situations monitored in the field to the domain model is essential. A semantic analysis of field monitoring data will also be required in order to identify gaps in the domain model, i.e. where behaviour was observed for which no causes in the domain model or as yet unidentified relationships between domain properties could be found. In order to argue the statistical relevance of the validation data, carefully selected performance indicators must be found that provide robust results when applying extreme value theory. Nevertheless, it is infeasible to expect that sufficient field data can be collected before release. Therefore, a successive approach to introducing Level 4 urban automated driving services is required that continuously collects validation data commensurate to a controlled widening of the ODD and level of authority given over to the system.

Assurance case

The assurance case will follow a similar structure to that for the Level 3 highway chauffeur case, in terms of qualitative arguments to demonstrate safety for a given ODD and quantitative arguments to increase confidence in the verification and validation measures. In a number of places, an even greater level of rigour in the argumentation will be required. In particular, arguing a sufficient level of understanding of the domain requirements and demonstrating that machine learning-based functions met their design contracts will require a cross-industry consensus to develop accepted argumentation strategies, eventually leading to a new generation of safety standards.

Due to the difficulty in capturing the variability of the domain using scenario descriptions, an over-reliance on scenario-based requirements and tests to demonstrate the safety of the vehicle should be avoided. This may lead to the temptation to focus assurance activities on passing specific scenarios rather than arguing the overall safety of the system for the entire ODD, even for scenarios which have not been defined in detail.

In addition, whereas for the Level 3 highway chauffeur class of systems, an argumentation is required for the safe handover between system and driver, for Level 4 urban automated driving an argument is required for the safe interaction between the system and other traffic participants in order to demonstrate that the behaviour of the vehicle does not promote additional hazards, for example due to reactions of pedestrians to different driving styles of automated vehicles.

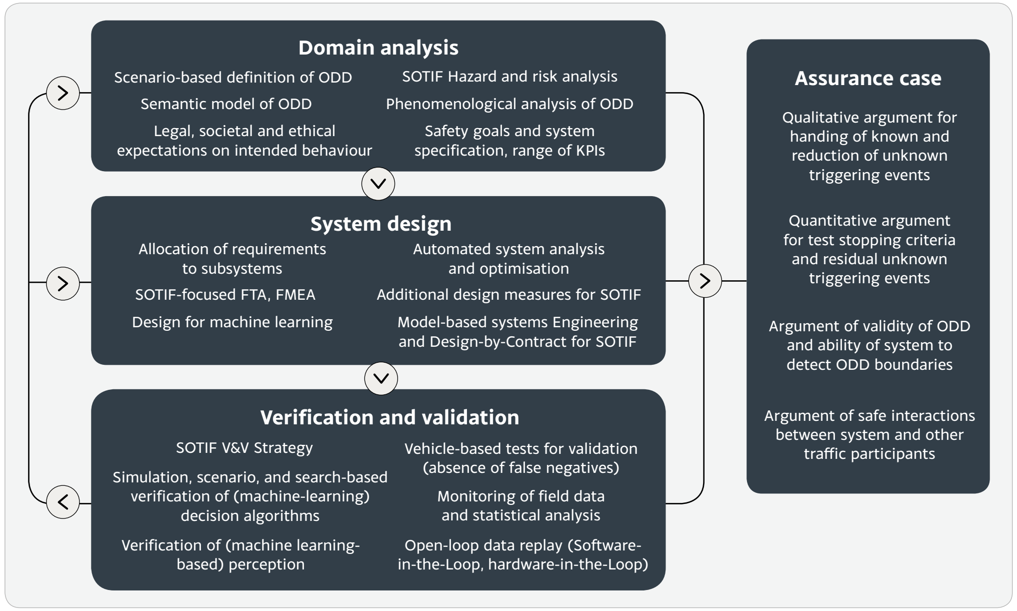

Domain analysis
Scenario-based definition of ODD
SOTIF Hazard and risk analysis
Semantic model of ODD
Phenomenological analysis of ODD
Legal, societal and ethical expectations on intended behaviour
Safety goals and system specification, range of KPIs
System design
Allocation of requirements to subsystems
Automated system analysis and optimisation
SOTIF-focused FTA, FMEA
Additional design measures for SOTIF
Design for machine learning
Model-based systems Engineering and Design-by-Contract for SOTIF
Verification and validation
SOTIF V&V Strategy
Vehicle-based tests for validation (absence of false negatives)
Simulation, scenario, and search-based verification of (machine-learning) decision algorithms
Monitoring of field data and statistical analysis
Verification of (machine learning-based) perception
Open-loop data replay (Software-in-the-Loop, hardware-in-the-Loop)
Assurance case
Qualitative argument for handing of known and reduction of unknown triggering events
Quantitative argument for test stopping criteria and residual unknown triggering events
Argument of validity of ODD and ability of system to detect ODD boundaries
Argument of safe interactions between system and other traffic participants

Figure 6: Assurance strategy for Level 4 urban automated driving (SOTIF focus)

Open research questions

The overall assurance approach for the Safety of the Intended Function for Level 4 urban automated driving is summarised in **Figure 6**. Based on the state-of-the-art review conducted within this report in comparison to the Level 3 scenario, the areas of the strategy with the most significant remaining open research challenges have been identified as follows:

- Approaches are required to ensure that societal, ethical and legal constraints are adequately covered in the resulting specification of the safe intended behaviour and that the completeness and consistency of these requirements are validated and their implementation verified.
- An industry-wide consensus on a set of representative use cases and scenarios for the testing of basic capabilities is required. This will include an investigation into the extent to which the scenario-based domain models and assurance methods developed for Level 3 functions can be applied to the increased complexity of Level 4/Level 5 use cases.
- Notations and methods are required for performing and recording the phenomenological analysis of the ODD with the particular focus on managing the complexity of the semantic ontologies and identifying abstractions that nevertheless capture the most critical properties to be considered in the safety concept.
- A set of standardised safety goals are required that can be formally expressed, including the consideration of dynamic aspects of the ODD as well as a set of technical key performance indicators to indicate the systems performance in relation to these goals. These key performance indicators must also be able to monitor the occurrence and probability of incidents where the system performed inadequately but did not lead to an accident.

- Scalable, tool-supported model-based systems engineering methods are required, supported by formal and automated analysis for capturing the requirements and system design, including the representation of performance-related requirements on the components that can only be formulated probabilistically. These methods must be supported by automated optimisation approaches to calculate optimal system architecture designs that satisfy all safety goals while optimising properties such as cost, energy consumption, weight and system availability.
- Targeted evaluation and industry-wide collaboration are required to develop an understanding of the types of automated driving functions for which machine learning not only leads to performance benefits but for which safety assurance arguments can be made. Future safety standards must be developed that explicitly consider machine learning functions from a design and verification perspective.
- Targeted research is required in order to understand the impact of automated vehicles in urban environments on the behaviour of other drivers and road users. Design measures and communication strategies will need to be found and agreed across the industry to reduce the risk of "collaboration hazards" caused by false assumptions made by traffic participants and automated vehicles.

5. Domain analysis and the definition of safety goals

5.1 Introduction

This section provides an overview of state-of-the art research and industrially applicable methods for performing the activities related to the domain analysis component of the framework.

In this context, domain analysis is used to identify classes of conditions that could lead to hazardous events. According to ISO 26262 [2], hazardous events can be defined as situations in which the vehicle can no longer avoid an accident without third-party intervention, situations in which not all relevant traffic participants have an adequate assessment of the situation, or situations in which traffic participants demonstrate unpredictable behaviour. The standard ISO PAS 21448 [4] for the Safety of the Intended Functionality (SOTIF) defines *triggering events* as specific conditions of a driving scenario that serve as an initiator for a subsequent system reaction possibly leading to a hazardous event. The standard requires that measures are defined in order that safe behaviour is maintained in the presence of all *known triggering events* and that the probability of *unknown triggering events* is minimised. The SOTIF standard does not give detailed guidance on how to identify relevant interactions with the environment or the insufficiencies of sensors, actuators and algorithms that could lead to triggering events.

5.2 Determining the open context

Identifying hazardous events and triggering events requires firstly that the intended system behaviour, in terms of its interaction with the environment, is understood. This in turn requires an understanding of the scope of conditions that may occur within the environment. The ODD defines the domain over which the automated vehicle is designed to operate safely. SAE J3016 [1] defines the ODD as *"Operating conditions under which a given driving automation system or feature thereof is specifically designed to function, including, but not limited to, environmental, geographical, and time-of-day restrictions, and/or the requisite presence or absence of certain traffic or roadway characteristics."* Only once the intended safe behaviour is clearly defined within a certain ODD can deviations from the intended functionality due to known and unknown triggering events be properly considered. If an adequate assurance case cannot be formulated for the absence of unreasonable risk due to functional insufficiencies and failures within a given ODD, then a restriction of the ODD may be necessary (e.g. to eliminate the possibility of triggering events not currently covered by the sensor set).

Neither ISO 26262 nor ISO PAS 21448 provide clear guidance on how to achieve this level of understanding of the intended behaviour for complex systems operating within an open context. For autonomous driving, a methodology is therefore required to generate a definition of safe behaviour of the system within the context of its target environment. A key part of the definition of the ODD must refer to classes of traffic situations to which the vehicle will be exposed and their accompanying environmental conditions. These traffic situations can be described in terms of scenes and scenarios, defined in [14] as follows:

*A **scene** describes a snapshot of the environment including the scenery and dynamic elements, as well as all actors' and observers' self-representations, and the relationships among those entities. Only a scene representation in a simulated world can be all-encompassing (objective scene, ground truth). In the real world it is incomplete, incorrect, uncertain, and from one or several observers' points of view (subjective scene).*

*A **situation** is the entirety of circumstances, which are to be considered for the selection of an appropriate behaviour pattern at a particular point of time. It entails all relevant conditions, options and determinants for behaviour. A situation is derived from the scene by an information selection and augmentation process based on transient (e.g. mission-specific) as well as permanent goals and values. Hence, a situation is always subjective by representing an element's point of view.*

*A **scenario** describes the temporal development between several scenes in a sequence of scenes. Every scenario starts with an initial scene. Actions and events as well as goals and values may be specified to characterize this temporal development in a scenario. Other than a scene, a scenario spans a certain amount of time.*

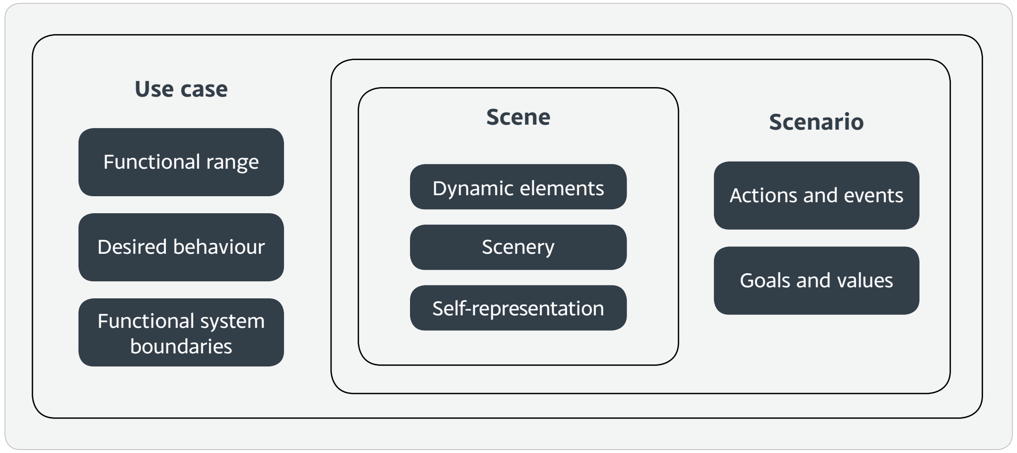


Figure 7: Taxonomy of use case, scene and scenario [14]

Based on the above definitions, the functional description of the system can therefore be defined in terms of a set of use cases that includes a functional range and the desired behaviour, the specification of system boundaries and the definition of one or more usage scenarios. The basic components of use cases, scenes and scenarios are summarised in **Figure 7** [14]. Furthermore, scenarios are well suited to describe test cases, in either a simulated or a real environment, by extending the scenario descriptions with pass/fail criteria. In order to derive a complete specification of the safe intended behaviour of the system it is also necessary to consider the legal and societal expectations on the system.

One key challenge to the safety assurance of highly automated driving is providing sufficient confidence that all possible systematic triggering events are known and controlled at the time of the release of the system. This requires that the use case and scenario catalogue covers a sufficient range of scenarios and their properties. [12] suggests that it is possible to identify structural principles in the complex space of driving situations and environmental conditions, and that these principles could be "learnt" based on an analysis of field data and accident databases labelled with ground truth. In general, methods for analysing the domain and identifying relevant representations can be categorised as follows:

- Analysis of pre-existing knowledge of the domain, including analysis of accident databases.
- Analysis of field data.
- Simulation.

The scenarios generated must be rigorously specified such that their completeness can be argued. This will help to ensure that experiences from operation in the field can be used to extend the model. It will also ensure that a consistent and unambiguous interpretation of the scenarios and associated test results is possible across all platforms. OpenSCENARIO (www.vires.com) is an attempt to standardise a language for describing autonomous driving scenarios and is used within the Pegasus project (www.pegasusprojekt.de). The Pegasus project describes a set of criticality indicators that can be derived from a scenario-based analysis and used as part of requirements derivation. An example of such a criticality indicator is "time-to-collision".

Traffic sequence charts (TSCs) [10] (see Figure 8) are an extension of the OpenSCENARIO approach and provide a formal semantics based on ontologies of categories for artefacts which must be observable in real traffic situations. Formally, TSCs visualise first-order real-time temporal logic formulas that refer to artefacts in the real-world model and focus on describing the required dynamic behaviour of the vehicle within different sets of situations. However, they do not serve to specify the range of inputs required to be considered when assessing potential insufficiencies in the sensing and understanding modules of the system. [10] proposes a development process by which scenarios are collected based on, for example, analysis of accident databases and then successively abstracted to cover classes of scenarios that share critical conditions. This set of scenarios is then used as the basis for successively refining the functional and non-functional requirements on the system.

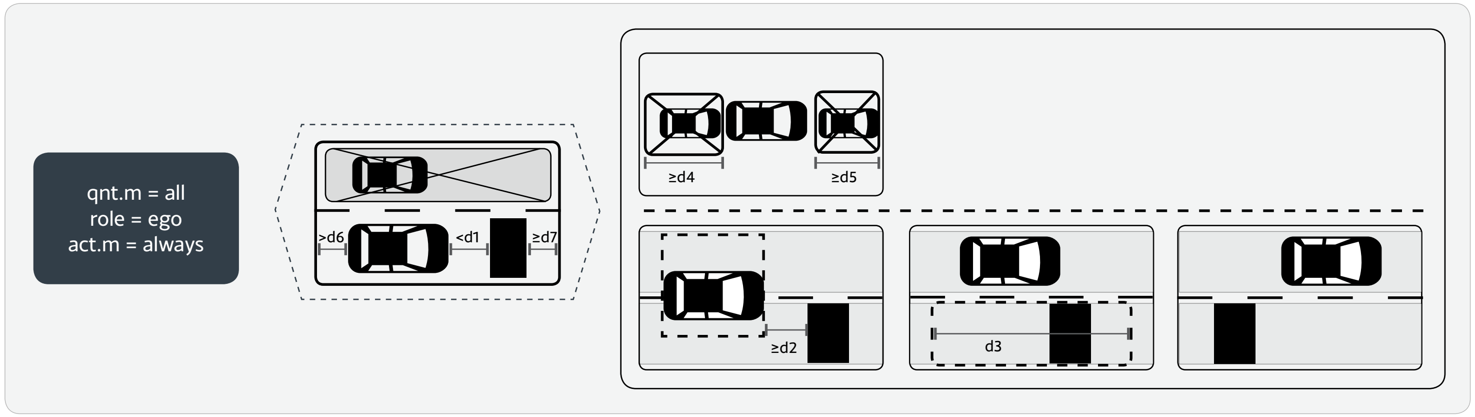


Figure 8: Example traffic sequence chart from [10]

5.3 Semantic modelling of the ODD

In order to be able to ensure traceability between the ODD and system specification, simulation and test environments and field-based monitoring data, a common semantic model capable of describing the ODD is required. This traceability is essential in order to refine the definition of the ODD over time to incorporate all findings from later phases of the development and deployment process. For example, choices of sensors will influence which environmental conditions need to be captured within the ODD. Triggering events captured during field-based testing that cannot be traced to the ODD definition imply that extensions of the semantic model and scenario classes are required. Semantic ontologies [15] have been proposed as a means for creating a common model for describing the ODD. For example [16] provides an ontology definition for an operational world model for autonomous driving that covers road structures, infrastructure and topologies, vehicles, pedestrians and animals and their associated behavioural models, other objects and environmental conditions. [17] proposes how knowledge-based systems can be used to provide an ontology of the ODD defined using the Web Ontology Language (OWL) standardised by the World Wide Web Consortium (W3C). The representation of the in first order logic allows for completeness and consistency checks to be made on the ontologies.

5.4 Systems theoretic approaches

An alternative to a scenario-based approach to domain analysis is to instead take a systems theoretic approach. Some work has investigated the use of applying Systems Theoretic Process Analysis (STPA) [18] to the safety analysis of automated driving [19]. Such approaches are based on the concepts of systems and control theory in order to identify control scenarios that can lead to hazards, and then develop the associated safety constraints. These approaches allow a focus to be placed on the interactions between the system and its environment and how certain states of the system and its environment could lead to hazards, regardless of whether a functional failure occurred within the system or not. As such, they offer an approach that may be better suited for analysing system insufficiencies than the safety analyses proposed by ISO 26262 [2], which are more focused on the consequences of specific types of component failures (random hardware faults, software bugs, etc.) of the system.

A scenario-based approach (e.g. [10]) to domain analysis and identification of criticality indicators has, due to its anecdotal approach to specification, the risk of incompleteness. On the other hand, system theoretic models (e.g. [19]) do not naturally scale to the open context issue of a highly complex and continuously changing environment. Due to an inadequate consideration of the environment, neither approach appears to be able to offer a complete analysis of triggering events that could lead to insufficiencies in the perception functions (sensing, understanding), incorrect decisions, and subsequent hazardous events.

5.5 Formulating safety goals

Currently, there is no industry consensus regarding the set of safety goals that an automated vehicle must achieve. These safety goals are inherently related to the ODD in which the vehicle operates and must not only ensure the safety of the vehicle occupants and surroundings but must not be so conservative as to render the vehicle unusable. In other words, the safety goals must define the boundary between an agile, natural driving style, crucial for user acceptance and unacceptable dangerous behaviour. Responsibility-Sensitive Safety (RSS) published by MobileEye [13] proposes a mathematical interpretation of the “elusive directive called duty of care”, an imprecise notion in traffic laws.

To this end, RSS formalises five “common sense” rules, which can also be interpreted as top-level safety goals:

- Do not hit someone from behind.
- Do not cut in recklessly.
- Right-of-way is given, not taken.
- Be careful of areas with limited visibility.
- If you can avoid an accident without causing another one, you must do it.

RSS constrains the path planning output to valid trajectories, for example, in the sense that minimum safety distances are maintained. These constraints are defined in terms of properties of the environment calculated by the perception function and can therefore also be monitored during run-time. The RSS approach aims to reduce the need for exhaustive validation tests by encapsulating safety properties as a formal model that can be “proven” to be adhered to at run-time. An additional layer of properties can be defined that minimise the impact of dangerous situations caused by other traffic participants (e.g. overtaking vehicle cuts in and violates the safe driving distance). The RSS approach is based on the following premises:

- The set of traffic scenarios used to derive the “common sense” rules are exhaustive.
- The behaviour of other traffic participants can be modelled using simple kinematic models.
- A realistic set of parameters can be found for these kinematic models based on explicitly stated assumptions, for example, the maximum reasonable deceleration that a lead vehicle might apply. Mobileye emphasises that these parameter values should be societally agreed upon and confirmed by regulators.

It is as yet unclear to what extent a complete specification of the safety goals in this way is feasible, due to the dependency on the above set of premises. Therefore, it is expected that the domain analysis and safety goal formulation are continuously validated and extended based on an analysis of simulation and field-based monitoring.

6. System design

6.1 Introduction

This section provides an overview of state-of-the-art research and industrially applicable methods for performing the activities related to the system design component of the framework. As discussed in Section 3, a black-box, data-driven approach to the verification of automated driving systems alone is not economically tractable [9]. Therefore, black-box verification techniques must be augmented with approaches by which individual elements of the system with well understood performance properties can be verified to a sufficient level of confidence. An analytical argument can then be formed that the composition of individual system elements leads to a sufficiently safe system overall. A key objective of the "design for assurance" activities is therefore to reduce overall verification effort, and increase confidence in the system performance by decomposing the system assurance task into a number of smaller, more tractable activities based on evidence from the system design and the inherent properties of individual components.

6.2 Requirements decomposition and the semantic gap

ISO 26262 [2] and ISO PAS 21448 [4] require that the safety goals (as identified via hazard analysis and risk assessments for the chosen classes of scenarios) are iteratively refined into a set of functional safety requirements which in turn are allocated to a technical system architecture and refined further into technical safety requirements. These technical safety requirements [20] are eventually allocated to individual hardware and software components. At all levels of system decomposition, assumptions are made about the behaviour of the environment or other components in the system. For complex systems, such as those required to implement highly automated driving tasks, it is essential that these assumptions are explicitly stated and validated as part of the assurance process. The challenge of demonstrating the refinement of the safety goals to a complete set of functional and technical safety requirements under consideration of all system and environment assumptions is described in [20] as the "semantic gap". A semantic gap occurs within the development process when the intended functionality is more diverse than the actual functionality specified by the implemented requirements (creating a gap between what is intended and what is specified). There is potential for semantic gaps to be introduced at multiple points in a systems engineering process, as summarised in **Figure 9**. Satisfaction arguments have been proposed [21] [20] as a means of providing rich traceability between requirements at different refinement levels. Satisfaction arguments take into account domain knowledge and assumptions while describing the specification refinement to provide an explicit record of the strategy used to decompose requirements.

It is currently unclear how the resulting effort and complexity in the requirements elicitation and management tasks associated with such approaches can be efficiently handled. One approach is to separate the specification of the domain, including all assumptions, from the specification of the safety requirements [22] and the specification of the system function itself. This allows the refinement of the safety requirements to be addressed in a more formal and complete manner than that of the more complex functionality. In addition, by describing the ODD separately (see Section 5 on domain analysis), this can be amortised over a number of system functions and variants.

This in turn leads to the potential for industry-wide standardisation of ODDs for different classes of scenario (e.g. German Autobahns, city-specific Urban Environments) to establish a common baseline on safety requirements while retaining the potential for differentiation via functional comfort, performance and cost. As the scope of desired system behaviour expands, the specifications can then be extended as required (e.g. increased scenario coverage or to allow for alternative perception channels).

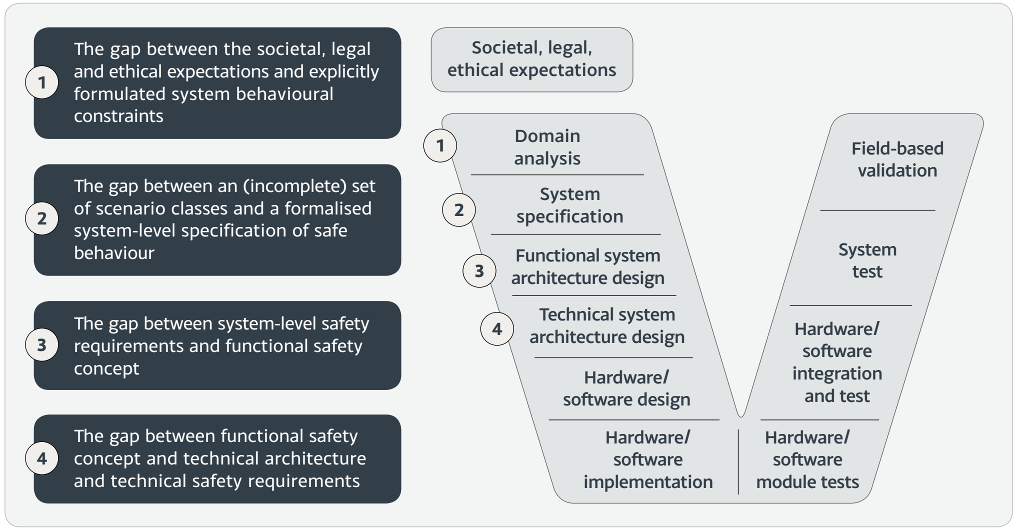

1 The gap between the societal, legal and ethical expectations and explicitly formulated system behavioural constraints
2 The gap between an (incomplete) set of scenario classes and a formalised system-level specification of safe behaviour
3 The gap between system-level safety requirements and functional safety concept
4 The gap between functional safety concept and technical architecture and technical safety requirements
Societal, legal, ethical expectations
1 Domain analysis
2 System specification
3 Functional system architecture design
4 Technical system architecture design
Hardware/ software design
Hardware/ software implementation
Hardware/ software module tests
Hardware/ software integration and test
System test
Field-based validation

Figure 9: Potential for semantic gaps in the system development process

Ultimately, it will be unrealistic to presume that a complete requirements specification can be created for such a complex system as HAD operating within an open context. Validation activities (see Section 7) must therefore be defined that determine whether the level of residual semantic gap is sufficiently small (equivalent to demonstrating a low enough probability of unknown triggering events). An industry consensus is likely to be required in order to determine the depth of detail required in requirements elicitation and analysis against the need for extensive validation tests.

6.3 Model-based systems engineering and design-by-contract

In order to handle the emerging complexity of the system requirements decomposition and system design, a number of characteristics of the system must be considered and optimised simultaneously. Model-based systems engineering [23] approaches allow for the specification of multiple perspectives on the functional and technical system architecture such that the dependencies between components across different architectural levels can be explicitly considered and specified [24]. A number of modelling languages have been developed ranging from the domain agnostic, such as SysML [25], to domain specific, such as AADL (Architecture Analysis & Design Language) [26], EAST-ADL [27], PREEVision (www.vector.com/preevision). **Figure 10** shows a simplified description of the system architecture for a driving assistance system using SysML.

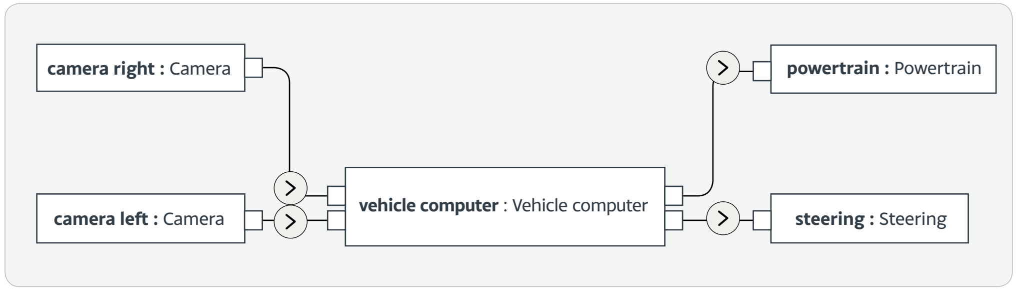


Figure 10: SysML internal block diagram of a driver assistance system [28]

The use of formal modelling languages can increase the precision with which statements about the system design can be explicitly documented and made amenable to analysis. However, the inherent complexity of the system design remains. Contract-based design techniques are a means by which the system complexity can be managed by breaking the system design into individually analysable components that are connected via ports and signals. A contract [29] specifies what each system or component expects from its environment (assumptions) and guarantees to its environment in turn (guarantees). The definition of assumptions and guarantees for each component allows for a compositional argument to be made for properties at the system level while allowing for each component to be considered as an independently verifiable “black box”. This allows for the verification complexity to be reduced and nevertheless precise statements about system properties to be made [30]. **Figure 11** shows example assumptions and guarantee conditions for the vehicle computer component from **Figure 10**.

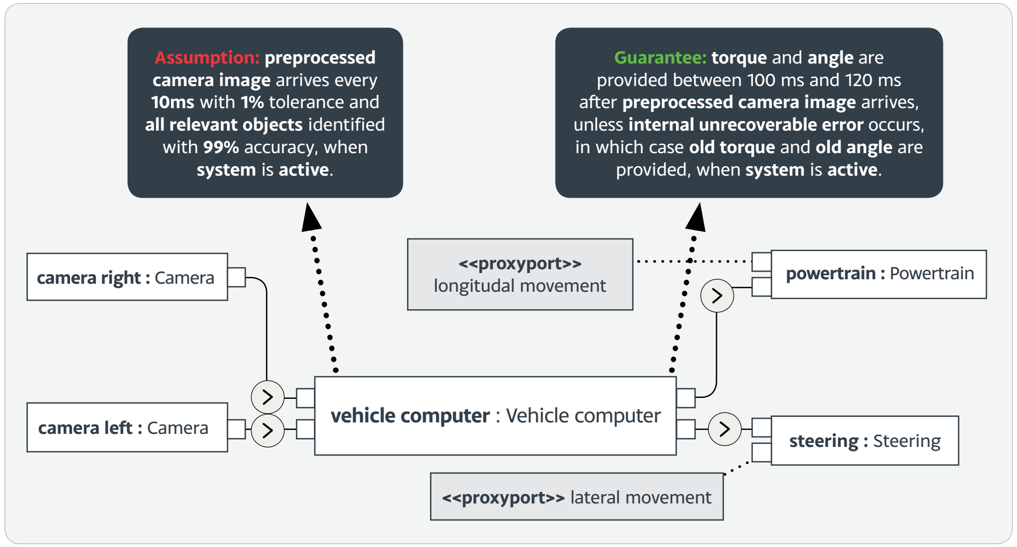

Assumption: preprocessed camera image arrives every 10ms with 1% tolerance and all relevant objects identified with 99% accuracy, when system is active.
Guarantee: torque and angle are provided between 100 ms and 120 ms after preprocessed camera image arrives, unless internal unrecoverable error occurs, in which case old torque and old angle are provided, when system is active.
camera right : Camera
<<proxyport>> longitudal movement
powertrain : Powertrain
camera left : Camera
vehicle computer : Vehicle computer
steering : Steering
<<proxyport>> lateral movement

Figure 11: Example contract-based design

For safety contracts, only safety-related properties and relevant environmental assumptions need to be considered, which can limit the state space to be explored during formal analysis. However, the safety contracts also must consider faults in the component’s environment that lead to erroneous inputs as well as execution faults of the component itself. At the component level it must be verified that within a reasonable level of confidence, the component fulfils its guarantees for all inputs that meet the set of assumptions (including faulty inputs and execution faults).

Conditional Safety Certificates [31] and Vertical Safety Interfaces [32] propose languages for specifying assumptions between components (horizontally) and between system abstraction layers (vertically). An example of horizontal safety contracts could include assumptions that a behavioural planning algorithm makes on the accuracy of a machine learning-based perception algorithm in order to choose the right course of action. An example of a vertical contract could be assumptions that the machine learning algorithm makes on the integrity of the hardware-based calculations necessary in order to correctly classify its inputs. Formalising safety contracts, for example using linear temporal logic [33], [34] allows for automated proof that certain compositional conditions, such as refinement of contracts, are met [35], thus providing a way for constructing compositional safety arguments [36] that ensure completeness for a given set of properties (often defined at the system's interfaces to its environment).

6.4 Model-based safety analyses

A complementary approach to safety contract modelling is the modelling of faults for a component, and the analysis of the propagation of those faults through the system. There are a number of approaches for doing this, such as using component fault trees [37] or Hip-Hops (Hierarchically Performed Hazard Origin and Propagation Studies) [38]. As an example, **Figure 12** shows an excerpt of a component fault tree for the Vehicle Computer component from **Figure 10** [28]. Based on an analysis of the component fault trees of each component, fault trees can be automatically generated for top-level system events.

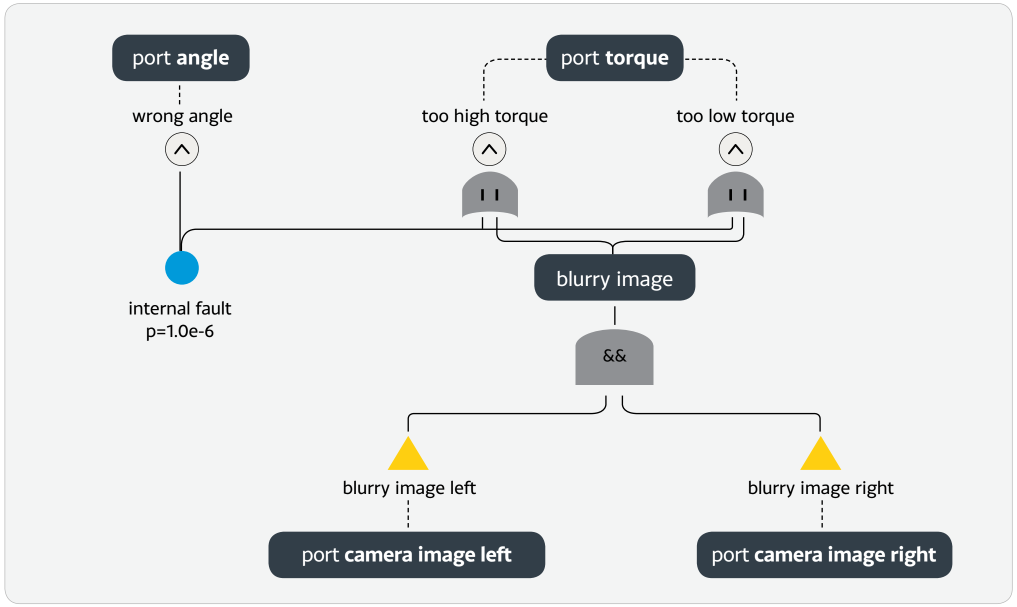


Figure 12: Component fault tree for the Vehicle Computer described in Figure 10 [28]

The ability to automatically analyse the impact of individual component faults on system-level failures based on model-based system designs allows for the possibility to automatically generate system design alternatives that optimise certain failure behaviours (e.g. overall system failure rate) [28]. The richer the failure model is, the more accurate an overall multi-goal optimisation can be.

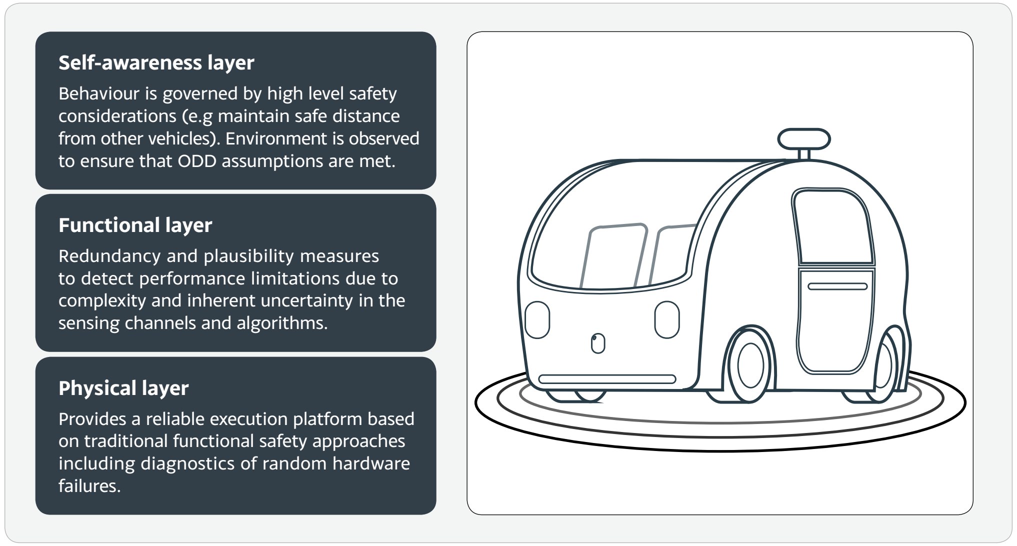
The use of contract-based design and component fault models is seen as particularly relevant in the development of HAD functions, due to the need to manage the high level of functional complexity and interaction between many system components. In particular, the system must cater for faults and insufficiencies in the components themselves while ensuring that all system-level safety goals are met. The use of safety contracts and component fault models allows for the integration and reuse of pre-validated third-party components in a modular safety case without the need for implementation details. This may facilitate the management of complexity explosion in the assurance argument when integrating proven components (e.g. radar sensors, already used within driver assistance contexts) by explicitly modelling knowledge about their known fault behaviour and performance limitations. Furthermore, the formalised contract-based design and failure propagation analysis techniques described above can be used to evaluate the effectiveness of design options for fault tolerance and redundancy mechanisms (see below).

In order for contract-based design and automated safety analysis approaches to cover failure modes associated with functional insufficiencies, model-based approaches must be extended to include a representation of component performance in terms of potential deviations from an ideal result as well as the probability of such deviations occurring [13]. Examples of such deviations would be inaccuracies in the location of objects within a scene. Small deviations between the sensed location and actual location may occur frequently, if not always, yet may have little effect on the safety goals. Larger deviations may occur less frequently but may have an impact on the safety goals. A model-based approach to specifying HAD functions must therefore consider functional performance of components within the safety contracts as well as during the analysis of fault propagation. Estimation of the insufficiencies of the components, for example, through simulation and test, will be discussed in Section 7.

6.5 Fault tolerance, monitoring and redundancy

A complex system is expected to have residual transient or latent faults. Therefore, a highly available safety-critical system must be designed to operate safely even in the presence of component faults. A key property to be considered during system design is therefore the ability to maintain a safe state in the presence of faults. This includes the definition of fault tolerance mechanisms as well as behavioural (also known as graceful) degradation such as coming to a safe stop at the side of the road when a system failure occurs. For highly automated driving scenarios, the driver cannot be relied upon to provide controllability of the situation. Therefore, the technical requirements on monitoring and handling component faults increase dramatically. Even fault detection and mitigation concepts within a system layer may not achieve a perfect level of coverage. Hierarchical layers of run-time monitoring are proposed (see **Figure 13**) to observe system properties at various levels of abstraction. At the *physical layer*, standard functional safety techniques are applied to ensure that the execution platform retains its integrity and to trigger a safe state upon detection of failures. Examples of such monitoring would be hardware diagnostics to detect random hardware failures. At the *functional layer*, measures must be found to counteract function performance issues due to inherent limitations in the sensing hardware (e.g. field of view of radar sensors) and algorithms (e.g. inaccuracies of machine learning). Such measures may include using heterogeneously redundant sensor channels, plausibility checks or the adaptation of driving behaviour based on anticipated sensor performance [39]. At the *self-awareness layer*, function-independent safety constraints identified from an analysis of the operational design domain are continuously monitored, as are the validity of assumptions made on the domain during system design (see recommendation of SAE J3061 [1], also known as functional boundary monitoring [40]).

An example of the monitoring of safety goals suggested by [13] is to ensure that decisions made by the planning components of the HAD function do not lead to a violation of safe longitudinal distance to the vehicle ahead, given current speed and road conditions.



Self-awareness layer
Behaviour is governed by high level safety considerations (e.g maintain safe distance from other vehicles). Environment is observed to ensure that ODD assumptions are met.

Functional layer
Redundancy and plausibility measures to detect performance limitations due to complexity and inherent uncertainty in the sensing channels and algorithms.

Physical layer
Provides a reliable execution platform based on traditional functional safety approaches including diagnostics of random hardware failures.

Figure 13: Levels of abstraction in run-time monitoring

A model and contract-based approach to system specification and design allows for the same models to be used to generate run-time monitoring components.

This helps to enable run-time, as well as design-time assurance through the continuous monitoring of properties that are important for the validity of the assurance case. The fault tolerance and monitoring systems must not only mitigate “classical” types of faults such as random hardware failures or software error, but also performance insufficiencies that are inherent in technologies such as machine learning [41]. Such components can be used for monitoring both functional and non-functional performance properties [42]. The use of “Self-Adaptation Envelopes” is described in [41] as a way to encapsulate undependable parts of the system. The function of these undependable parts can then be suppressed if their behaviour is observed to fall outside a pre-specified safety-envelope. A similar approach has been defined in the form of Dependability Cages [43], that are capable of comparing observed scenarios to previously specified safe behaviour. This enables the checking of functional correctness and the validity of assumptions on the ODD at run-time. The RSS approach proposed by MobileEye [13] could also form the basis of generating run-time checks of the formalised safety goals at the vehicle behaviour level.

Within the automotive industry we currently see a move away from federated E/E architectures, characterised by electronic control units (ECUs) dedicated to specific (groups of) functions that communicate with other ECUs [44]. These are being replaced by integrated architectures based on powerful general-purpose computing platforms. In parallel, the complexity and interaction between software components are increasing, requiring new software architectures that can nevertheless fulfil the timing, safety and reliability constraints of the embedded domain. One advantage offered by the availability of increased general-purpose computing power and advanced software paradigms such as Service Oriented Architectures is the possibility for run-time reconfiguration of the system, at least for the short period of time and limited functionality required to bring the vehicle to a safe stop [45].

Classic approaches to homogenous hardware redundancy as prevalent in the aerospace domain are unlikely to take hold for many system components (with the exception of certain actuators such as steering control) due to economic drivers of the mass market. Therefore it must be ensured that the reconfiguration process itself is executed to an appropriate level of integrity, including ensuring that sufficient resources are available to execute the fail operational functionality and that the performance of the function (in terms of functional insufficiencies) can be demonstrated to be adequate, even after reconfiguration.

6.6 Machine learning

One of the challenges caused by applying machine learning methods to implement parts of the "sense, understand, decide, act" function chain is that a precise specification of the required behaviour is often not possible. It is the very fact that the machine learning functions are able to infer the target function without a specification, based on the presented training data, that makes them desirable for use within an open context. This introduces uncertainty into the safety assurance process, which is further compounded by the unpredictable and opaque nature of the performance of the algorithms.

In order to argue the claim that functional insufficiencies within the machine learning function are minimised, it is important to understand the causes of such insufficiencies. As interest in machine learning safety has grown: a number of authors [46], [47], [48] have investigated different causes of performance limitations in machine learning functions. Some examples applicable to the highly automated driving are described as follows.

Scalable oversight and distributional shift

One of the key differences in machine learning techniques compared to algorithmic approaches is the lack of a detailed specification of the target function. Instead, the functional specification can be seen to be encoded within the set of training data. Therefore, if the training data do not reflect the target operating context, then there is a strong likelihood that the learned function will exhibit insufficiencies. Critical or ambiguous situations, within which the system must react in a predictably safe manner, may occur rarely or may be so dangerous that they are not well represented in the training data. Consider, for example, the situation where a small child enters the road ahead between two parked vehicles. This leads to the effect that critical situations remain undertrained in the final function (scalable oversight). The potential of scalable oversight has profound consequences for the selection of training data. It must be argued that the training data contains an equal distribution of all classes of critical situations and object classes or that the selected training leads to an appropriate level of generalisation. Such considerations will also be necessary in order to reflect legal and ethical concerns regarding discrimination against certain demographic groups, such as ethnic minorities that may not be strongly represented in training and test data.

In addition, the system should continue to perform accurately even if the operational environment differs from the training environment (distributional shift) [47]. This effectively can be formulated as the robustness of the system to react in a shift of distribution between its training and operational environment. Distributional shift will be inevitable in most open context systems, as the environment constantly changes and can adapt to the behaviour of actors within the system.

For example, car drivers will adjust their behaviour within an environment in which autonomous vehicles are present, vehicle and pedestrian appearances change over time, etc. Addressing distributional shift will require design measures to monitor for distributional shift in the field and to identify when a retraining of the machine learning function is required.

Robustness of the trained function

Machine learning techniques are typically chosen for their ability to approximate target functions based on a finite set of training data. This has advantages over procedural techniques where the function to be implemented may be too complex to specify or implement algorithmically due to an open context environment or due to the unstructured nature of the input data. In other words, when presented with new data, the function will predict a correct answer based on already observed input/output pairs. An often cited problem, associated with neural networks, is the possibility of adversarial perturbations [49], [50] [51]. An adversarial perturbation is an input sample that is similar (at least to the human eye) to other samples but that leads to a completely different categorisation with a high confidence value. It has been shown that such examples can be automatically generated and used to "trick" the network. Although it is still unclear to what extent adversarial perturbations could occur naturally or whether they would be exploited for malicious purposes, from a safety validation perspective, they are useful for demonstrating that features can be learnt by the network and assigned an incorrect relevance. Therefore, methods are required to minimise the probability of such behaviour especially in critical driving situations. One of the factors that is often attributed to this class of problems is that the set of possible functions is exponentially larger than those that can be represented through machine learning techniques. Therefore, the likelihood that a machine learning technique would select an appropriate approximation appears at first glance very unlikely. The authors in [52] argue, however, that deep learning is nevertheless effective because the function to be approximated is rooted within the physical universe and physics favours certain classes of exceptionally simple probability distributions that deep learning is uniquely suited to model. The challenge, therefore, is how to ensure that the machine learning algorithms focus on those physical properties of the inputs relevant to the target function without becoming distracted by irrelevant features; in other words, act within the same hierarchical dimensions as the target function [52]. This has an impact on the selection of training data as well as the application of "explainable AI" techniques to better understand the features used by the machine learning function for the decisions. Additionally, constructive measures such as protection of confidentiality and integrity of critical data and software components, plausibility checks, diverse sensing functions etc. may need to be applied to protect against deliberate manipulation based on adversarial perturbations.

Differences between the training and execution platforms

As discussed above, machine learning functions can be sensitive to subtle changes in the input data. When using machine learning to represent a function that is embedded as part of a wider system as described here, the input to the neural network will have typically been processed by a number of elements already [48], such as image filters and buffering mechanisms. These elements may vary between the training and target execution environments, leading to the trained function becoming dependent on hidden features of the training environment not relevant in the target system.

In addition, typical reliability issues in the target hardware (e.g. random hardware failures) may not manifest themselves directly as obviously erroneous outputs, due to the data-driven approach where deviations of individual parameters or calculations may have subtle but relevant effects on the overall decision made by the neural network.

Contract-based design and monitoring

The use of contract-based design has been proposed [53] in order to restrict this level of uncertainty within the design and provide a clearer specification that can be used for verification activities. Assumptions with respect to the ODD (e.g. distribution of certain types of objects) and quality of input signals (e.g. camera blur) are recorded in the contract. Performance targets on the function, systematically derived from the system safety goals (see Section 6.2) are then assigned to the guarantees of the contract [54]. In formulating the guarantees, it is important to note that not only positive performance criteria but also typical failure modes and their likelihood should be considered. The assumptions and guarantees of the safety contract for the machine learning function can then be used to analyse the impact of insufficiencies on the overall system behaviour (see above) as well as to ensure that the system design fulfils conditions necessary for the function to meet its guarantees. Further discussion on verification methods for demonstrating that the guarantees are fulfilled can be found in Section 7.

In many cases it may not be possible to provide sufficient verification evidence that the machine learning function meets its guarantees under all conditions to fulfil its assumptions. In these cases the design-by-contract methodology provides a mechanism to reason about the impact of further restrictions on the input space and also how additional design measures external to the machine learning function – such as redundant calculation based on heterogeneous approaches or plausibility checks – can be used to meet the performance guarantees and required level of residual uncertainty. Such design measures can be used at run-time to verify that the assumptions about the function inputs are met and whether the outputs meet a set of given safety constraints [55]. [56] describes an approach by which activation patterns of a convolutional neural network are observed at run-time and compared against typical activations that were caused by training data. In this way, a measure of whether the trained function is “outside its comfort zone” can be gained in order to determine the level of trust in its output during run-time and to detect issues such as distributional shift.

7. Verification and validation evidence

7.1 Introduction

This section provides an overview of state-of-the-art research and industrially applicable methods for performing the activities related to the verification and validation component of the framework.

The costs and technical challenges associated with demonstrating the absence of risks due to performance issues is seen as one of the largest barriers to the market entry of HAD systems [57]. Attempting to demonstrate the safety of autonomous vehicles using road tests alone would involve between millions and billions of miles of driving. For the occurrence rates required by HAD functions, measuring the probability of hazardous events through vehicle testing therefore does not scale, even with large fleets of continuously running vehicles. Significantly more testing hours than the inverse of the target failure rate are required in order to provide a statistically relevant statement about the probability of critical failures. For example, to argue the mean distance between collisions of 3.85 million km (based on German crash statistics) with a confidence value of 95%, applying Poisson statistics [7] a total of 11.6 million test kilometres must be driven without collisions [58]. Even when applying such statistical extrapolations, a strong argument must still be applied for the representativeness of the test scenarios to argue that sufficient coverage of system states and external triggering events has been achieved.

Controllability, observability and repeatability also provide additional significant challenges when testing highly automated driving systems, in particular with regard to the type of functional insufficiencies which are the subject of this report:

- **Controllability**: Due to the complexity and unpredictability of the ODD, it is extremely difficult to consistently control all relevant attributes of the domain and vehicle state in order to systematically perform tests at the vehicle level, for example in order to test robustness against known triggering events. This may be due to the complex interactions between a number of domain attributes that lead to a triggering event, the rare natural occurrence of such events or the inherent danger associated with the situations themselves (e.g. realistically testing for false negatives for a pedestrian detection function).
- **Repeatability**: Aleatoric uncertainty in the ODD, epistemic uncertainty in the sensors, perception and decision algorithms, as well as the effects of memory (e.g. training history), lead to pseudo-non-deterministic behaviour with regard to triggering events for functional insufficiencies. Even though for any given set of identical inputs and internal state the system may present identical results, it is nevertheless nearly impossible to directly reproduce all aspects of any given situation. This leads to major challenges when demonstrating the robustness of the function within the ODD and reproducing failures which occur in the field for further analysis in the lab.
- **Observability**: Due to the lack of a detailed specification of the required system behaviour under all possible conditions, there are significant challenges in defining the set of testing criteria and therefore also the pass/fail criteria for the tests (including a definition of "ground truth"). Additionally, the pass/fail criteria might not be able to be defined in a binary manner but instead be defined on

a continuous scale that takes into consideration the objective level of danger in the situation in combination with the appropriateness of the system's response [58]. The complexity of the domain and difficulties in evaluating the coverage of all relevant domain attributes also increase the difficulty of defining and measuring test completeness criteria.

As functional insufficiencies are in effect systematic failures of the system, in addition to providing a purely statistical argument for a residual level of risk, evidence must be gathered that all critical scenarios within the ODD have been covered and that the absence of unknown triggering events can be argued. In order to achieve this in an economical manner, an optimal strategy is required that combines verification and validation methods to achieve an optimal coverage of test goals while maximising the possibilities for automation.

7.2 Verification and validation strategy

The objective of a verification and validation strategy is to confirm quality attributes of the system under test using an optimal combination of methods distributed across the development and deployment process. These quality attributes typically include functionality, reliability, usability, efficiency, maintainability and portability, for example as defined in the ISO/IEC 9126 Standard "Software engineering – Product quality" [59]. The quality attributes can vary in priority depending on the requirements of the system under test and current development phase and are in turn used to derive the set of detailed verification and validation (V&V) goals. An assurance case is typically formulated as a set of assertions about the product which are backed up by evidence in the form of development work products. The contribution of verification and validation to the assurance case can therefore be defined in terms of the set of quality attributes considered by the V&V strategy that in turn support the assertions in the assurance case. This requires a specific interpretation of the quality attributes defined [59] in the safety context:

- In confirming the **functionality**, it should be ensured that all safety-relevant functions (e.g. the algorithm for detecting lane departure) and safety mechanisms (e.g. detection of hardware failures) are free from critical errors. To provide the necessary level of confidence, the V&V strategy should ensure a greater level of test depth compared with non-safety-relevant functions and be reliant on more than one method of verification (e.g. a combination of test, review and analysis).
- Confirming the system **reliability** should involve quantitatively determining that the probability of a system failure and insufficiencies leading to a violation of a system safety goal is sufficiently low. Analytical approaches to calculating the predicted reliability should be validated through empirical evidence in the field and large-scale simulation based on realistic distributions of input data.

The topic of functional insufficiencies can be seen as a combination of the above quality attributes, where systematic functional failings of the system may manifest themselves in a pseudo-non-deterministic manner more symptomatic of system reliability issues. In addition, the following safety-specific attributes need to be considered:

- The **validity of the safety concept** should be confirmed in terms of its adequacy to ensure the continued safe operation of the vehicle under all foreseeable failure conditions. The argument for the validity of the safety concept will typically be based on a combination of vehicle-level testing, fault injection tests, targeted safety analyses such as FMEA and FTAs (see Section 6) and expert reviews.

- When defining the safety concept, a number of **assumptions** are typically made (e.g. regarding driver reactions to cancel inopportune steering impulses, distribution of critical scenarios in the domain, etc.). For the safety case to hold, the validity of these assumptions must be confirmed. This may involve targeted tests at the vehicle level as well as simulations and analysis.

- A **quality assurance** is required that all activities necessary for sufficiently ensuring safety of the product have been adequately performed. This assurance is typically achieved not only through testing that has the focus to ensure that errors have not been added to the product but also through reviews and analyses that can detect a wider range of issues, such as whether critical aspects remain unconsidered during the system analysis phase, or whether suitable programming techniques have been applied.

ISO 26262 [2] also contains a number of requirements on the verification and validation of the product in terms of analyses, reviews and tests which can be summarised as follows:

- **Part 2 – Management of functional safety:** Confirmation reviews are performed to ensure that safety-relevant work products conform to the corresponding ISO 26262 requirements before being released.
- **Part 3 – Concept phase:** Verification reviews of the hazard and risk analysis, safety goals and functional safety concept are performed. The effectiveness of the functional safety concept is evaluated through analysis and test.
- **Part 4 – Product development: System level:** Verification reviews of the technical system safety requirements and system design are performed. The integrated system is tested against the technical system safety requirements and verified against the system design. Compliance of the integrated system against the safety goals is confirmed and the safety goals themselves are validated.
- **Part 5 – Product development: Hardware level:** Verification reviews of the hardware safety requirements and hardware design and hardware safety analyses are performed. Evidence of the effectiveness of hardware-level safety mechanisms is provided (e.g. based on analysis or test). Compliance of the hardware to the hardware safety requirements is confirmed through testing.
- **Part 6 – Product development: Software level:** Verification reviews of the software safety requirements, software architecture design and implementation are performed. Analysis is performed at the unit design and code level. Testing is performed at the unit, software integration and system integration levels to verify conformance to the software safety requirements.
- **Part 8 – Supporting processes:** All verification activities throughout the development process are planned and specified. Verification of reused software in its new context is performed. Qualification through test of pre-developed hardware components is performed.

The summary above makes clear the need to fulfil the requirements of the ISO 26262 standard as a prerequisite for addressing the issues of functional insufficiencies in order to ensure that the overall system concept and its implementation is sufficiently robust before considering the specific hazards associated with functional insufficiencies.

Requirements on the verification and validation of the product from the perspective of the "Safety of the intended function" as defined by the ISO PAS 21448 [4] standard can be summarised as follows:

- Evaluate the safety of the intended function with regard to **known triggering events** to provide an argument that the residual risk associated with hazards caused by known insufficiencies in the system is sufficiently low. The standard suggests a number of verification methods that should be used in an appropriate combination for performing verification at the sensor, decision algorithm and actuation component levels, as well as at the system integration level.

- Evaluate the safety of the intended function with regard to **unknown triggering events** to provide an argument that the residual risk associated with hazards caused by unknown insufficiencies in the system is sufficiently low. The methods proposed by the standard for validation are focused on system-level properties and should include a rationale for the amount of testing performed in relation to the overall target for residual risk.

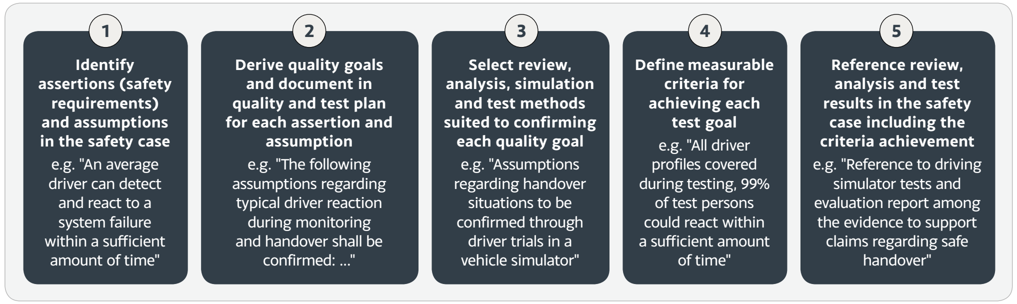
In doing so, the V&V strategy shall confirm the following system properties [4]:

- The ability of sensors and the sensor processing algorithms to model the environment;
- The ability of the decision algorithms to handle both known and unknown situations and to make the appropriate decisions according to the environment model and the system architecture;
- The robustness of the system or function;
- The ability of the human-machine interface to prevent reasonably foreseeable misuse; and
- The manageability of the handover scenario by the driver.

For each V&V requirement in both standards, a set of alternative methods is given with no guidance as to which combinations thereof should be applied in which circumstances. In addition, the relationship between the V&V methods to the safety case is not explicitly addressed, i.e. how does applying the prescribed methods demonstrate the safety of the end product? A comprehensive strategy is therefore required that justifies the selection of the methods and thereby demonstrates a clear contribution to increasing the product quality and providing a convincing argument for safety. This justification can be decomposed into a number of steps, an example test strategy derivation is shown in **Figure 14**:

- A concretisation and prioritisation of the quality attributes to be confirmed is made and a partitioning of the V&V effort accordingly, to ensure that the most critical (i.e. safety-relevant) attributes are assigned sufficient resources to be covered adequately.
- V&V methods are selected, based on a clear understanding of their effectiveness in detecting certain types of errors and quality issues in particular phases of the development. The methods are then distributed across development phases and combined accordingly. For example, at the software unit implementation phase, unit testing is effective at detecting discrepancies in the code with respect to the detailed software design but is unlikely to detect subtle programming errors leading to run-time exceptions for which static code analysis is more effective. Performing such an analysis at a later phase of development, however (e.g. at handover from software supplier to system integrator), will lead to an impractical amount of rework to correct and re-verify the code. The combination of methods should aim to ensure a full coverage of the quality attributes in the focus of the test strategy and their associated types of faults, while minimising the level of redundancy between the methods.

- For each method selected to confirm a particular quality attribute, a set of criteria should be defined that measure as directly as possible the contribution of the analysis, review or test method to the associated quality attribute. It is the achievement of these criteria that then "make the case" for the associated assertions in the safety case. For example, part of the safety case may read: "The residual risk associated with unknown triggering events is sufficiently low because: field-based testing has covered 100% of the scenario classes identified during domain analysis with the detection of 0 critical errors, and targeted testing on the test track under a range of conditions and in simulation have covered 100% of the domain attributes identified as relevant for perception tasks und uncovered 0 common cause failures".
- By formulating the assertions in the assurance case as described, it becomes possible to validate the strength of the assurance case by questioning whether it is possible for the system to nevertheless contain critical errors even if all the above-mentioned criteria are met.



1 **Identify assertions (safety requirements) and assumptions in the safety case** e.g. "An average driver can detect and react to a system failure within a sufficient amount of time"

2 **Derive quality goals and document in quality and test plan for each assertion and assumption** e.g. "The following assumptions regarding typical driver reaction during monitoring and handover shall be confirmed: ..."

3 **Select review, analysis, simulation and test methods suited to confirming each quality goal** e.g. "Assumptions regarding handover situations to be confirmed through driver trials in a vehicle simulator"

4 **Define measurable criteria for achieving each test goal** e.g. "All driver profiles covered during testing, 99% of test persons could react within a sufficient amount of time"

5 **Reference review, analysis and test results in the safety case including the criteria achievement** e.g. "Reference to driving simulator tests and evaluation report among the evidence to support claims regarding safe handover"

Figure 14: Example test strategy derivation

The application of a structured design process and use of design-by-contract paradigms (see Section 6) can be exploited when conceiving the V&V strategy by providing a clear definition of the quality attributes that should be applied at each level in the system decomposition. The design-by-contract paradigm also provides a definition of the assumptions that are to be validated at the interfaces of the components and allows for separately focused activities to verify that the integration requirements of the components are met. The combination of design-by-contract and systematic V&V strategy development supports the development of a modular safety assurance case allowing evidence at the component level to be reused within other system contexts. The design-by-contract approach does not preclude the need to validate safety at the system level, but it does allow for arguments to be made regarding the fundamental performance limits of individual components and the robustness of the system in the presence of insufficiencies at the component level. System-level validation activities can then focus on coverage of the ODD and the confirmation of assumptions made on both the ODD and system components during design.

The following subsections of this report assess a number of V&V methods in relation to their ability to generate evidence that supports the test goals described above.

7.3 Formal verification

Formal verification covers static analysis approach that prove particular properties of a system. It is defined in [2] as follows:

"formal verification": method used to prove the correctness of a system against the specification in formal notation of its required behavior.

whereby a formal notation is defined as follows:

"formal notation": description technique that has both its syntax and semantics completely defined.

Such techniques that may include symbolic model checking [60] and theorem proving have an advantage over traditional test techniques in that they can provide a complete analysis of the entire input space rather than just providing anecdotal evidence based on the chosen test data. As such, there has been much interest in the past in the application of formal verification methods to demonstrating properties of safety-critical systems.

According to the principles of the V&V strategy described above, within the context of automated driving it needs to be clear which system properties formal verification approaches are well suited to. Formal verification requires a formal specification of the properties to be proven as well as a model of the implementation to be verified.

Section 6 described the use of design-by-contract approaches to system design. Formal verification can be used to verify that the composition relations between formally specified assumptions/guarantees chains in the system decomposition [33], [34], [35], [36] are consistent.

This allows for an increase in trust in the functionality at the system level, once the components have been verified according to their individual contracts, which due to controllability and observability issues can be typically performed to a greater depth than system-level tests allow for.

An additional application of formal verification within the context of automated driving is to verify that the behavioural planning components (decision components within the "sense, understand, decide, act" model) adhere to a set of pre-defined safety constraints such as those described in [13]. As discussed in Section 6, it is however unclear whether a sufficiently complete specification of safe driving behaviour is possible for such techniques to be realistically applied.

In [61] the authors discuss an approach to analysing autonomous systems using model-checking. They focus on the agent responsible for decision-making, and firstly model its behaviour and describe its interface. They then establish a required property by model-checking the agent within a model of the real-world environment. From this they can derive properties of the overall system by using theorems or analysis of the environment.

Although formal approaches to verification can be very powerful, there are also a number of limitations that must be taken into account. The first and perhaps most significant is that the formal proof of properties, as has been described, requires assumptions to be made about the environment and operation of the system which are hard to specify, and even harder to verify. If the assumptions do not hold in the operational system, then the analysis performed may be invalid, and it may lead to unfounded confidence in the system. Secondly, many formal verification techniques can be hard to use for non-experts, providing an impediment to widespread adoption. As tool support improves and becomes more user-friendly, it is hoped that this impediment may reduce.

7.4 Verification of machine learning functions

The development of methods for demonstrating the performance of machine learning functions to the level of integrity required by safety-critical systems is currently an emerging field of research.

It is expected that, analogous to traditional algorithmic-based software approaches, a diverse set of complementary evidence based on constructive measures, formal analysis and test methods will be required. In this section, we discuss different categories of potential evidence that can be used to support such an assurance case.

When developing a verification strategy for machine learning functions, a set of test goals are required against which the effectiveness of the approaches can be measured. In [62], the authors suggested the following dependability attributes of neural networks applied to automated driving perception tasks which should be covered by a verification strategy:

- **Robustness** against effects such as distortion or adversarial perturbation;
- **Interpretability** related to which features were actually learned by the function;
- **Completeness** related to coverage of important scenarios during training;
- **Correctness related to the ability of the trained function to perform its task without errors.**

The choice of training data has a direct impact on accuracy of a machine learning function. Criteria are therefore required in order to determine whether or not the training data have the potential to lead to a sufficient level of performance, including:

- **Training data volume:** A sufficient amount of training data is used to provide a statistically relevant distribution of scenarios and to ensure a stabilisation of a strong coverage of weightings in the neural network.
- **Coverage of known, critical scenarios:** Domain experience based on well-understood physical properties of the system and environment as well as previous validation exercises ensures the identification of classes of scenarios that should exhibit similar behaviour in the function.
- **Minimisation of unknown, critical scenarios:** Some critical attributes of the input space may not be known during system design [63]. A combination of systematic identification of equivalence classes in the training data and statistical coverage during training and validation will therefore be essential to minimise the residual risk of insufficiencies due to inadequate training data.

A key component of demonstrating the correctness of traditional safety-critical software is introspective techniques that include manual code review, static analysis, code coverage and formal verification. These techniques allow for an argument to be formulated on the detailed algorithmic design and implementation but cannot be easily transferred to the machine learning paradigms. Other arguments must therefore be found that make use of knowledge of the internal behaviour of the neural networks.

- **Saliency maps:** Based on the back propagation of results in the neural network, Saliency maps [64] highlight those portions of an image that have greatest influence on classification results. As such, they can be used to provide a manual plausibility check of results as well as to determine potential causes of failed tests.
- **Explanations:** Another line of research tries to generate natural language explanations referring in human understandable terms to the discriminating contents of an input image to explain which features were relevant for the classification [65].
- **Distinguishability measures and adversarial perturbations:** The robustness of the trained function appears to be related to its susceptibility to adversarial perturbations.

[66] introduces the concept of distinguishability, whereby the difficulty of the classification task is calculated as a distinguishability measure. Furthermore, it was shown that the robustness of the classifier against random noise is proportional to its robustness against adversarial perturbations. The use of distinguishability measures could therefore provide an indicator of the potential robustness of the trained function.

Due to the inherent restrictions of the applicability of white-box approaches to the verification of the trained function, a strong emphasis will remain on testing as a means to estimate the achieved performance of the trained function. Standard approaches to testing machine learning functions involve reserving a proportion of the data collected for training purposes to performing validation tests. These tests naturally suffer from the same inadequacies as described above for the training data. Several additional test approaches are therefore being developed.

- **Synthetic data generation and search-based testing:** Based on advances in computer graphics realism as well as the possibility to generate data with specific properties, the use of synthetically generated data may also play a role [67] in the assurance case. Synthetic data can be used to generate huge numbers of test cases, in particular to cover critical or rare situations, otherwise not adequately represented in naturally occurring data. The use of synthetic data also allows test cases to be automatically generated together with the corresponding ground truth. This allows for search-based optimisation approaches to be applied to automatically generate (physically feasible) images which produce incorrect classifications. However, the use of synthetic data also implies the introduction of the additional assumption in the assurance case that the synthetic data would lead to test results that are indeed representative of the operational environment.
- **White-box coverage tests:** At present, there is no clear consensus on which stopping criteria to apply when testing machine learning functions. Due to the fact that deep neural networks operate in a highly dimensional feature space, choosing test cases based on a set of domain-specific equivalence classes is less likely to be effective, as there is a high chance that these do not match the feature dimensions learnt by the neural network. White-box criteria have been proposed based on the concept of neuron coverage to determine the completeness and effectiveness of the test data. This involves calculating the ratio of activated neurons (activation values above a given threshold) to the total number of neurons for a given set of input data [68], [69]. These approaches have also been combined with search-based testing techniques to create variations of test data that achieve coverage. These techniques are only applicable in combination with functional criteria and it is as yet unclear how effective such white-box techniques are at discovering performance issues in the neural networks.

7.5 Simulation

The use of simulation when testing one or more components of a HAD system addresses the issues of controllability, observability and repeatability (see Section 7.1) by simulating the software interfaces of the system within a controlled, synthetic environment. By using synthetic test data, a greater coverage of the ODD and scenario classes can be achieved, including those conditions that are difficult or dangerous to reproduce in the real world. If simulated tests can be demonstrated to be representative they can be used to form an argument to reduce the amount of driving hours required to form a statistical argument for freedom of unacceptable risk in the final system, thereby making the release argument for HAD more economically feasible.

The use of simulation can be used in the verification of different parts of the system and at different levels of system integration.

Simulation for verifying sensing and understanding functions: This form of testing aims to verify the robustness of the sensing and understanding functions across a wide range of conditions. Synthetic scenes can be generated using photo-realistic graphics engines that automatically include ground truth data needed for either training or verification. Such data sets are also already publicly available (e.g. Virtual KITTI [70], or SYNTHIA [71]) allowing for the benchmarking of different implementations. Due to the need to explicitly generate the input data based on a set of pre-defined characteristics (e.g. lighting, weather conditions, etc.) this type of testing is well suited to verifying the response of the system to known triggering events that can be generated at scale and with well-controlled coverage and variation. An alternative approach to using synthetic data is to augment data directly recorded in the environment (e.g. using existing images of street scenes). In [72], the authors applied augmentation of image data to verify the robustness of pedestrian detection functions against perturbations in images that reflect physical conditions such as haze and defocus. While the use of simulated input data can be an efficient approach to verifying the sensing and understanding functions, questions will remain regarding the representativeness of the data and transferability of the results, especially where the processing performed by the function is opaque and sensitive to subtle differences in input data unrecognisable by humans as is the case with many machine learning approaches. Therefore the use of simulation can only support and not replace real-world tests which are required to identify previously unknown triggering events.

- **Physical simulation of sensing functions:** An alternative application of simulation to sensing and understanding functions is to perform extensive simulation of the physical properties of the sensors themselves. This may include, for example, investigating optical properties of camera lenses [73], [74], simulation of the propagation of radar reflections [75] or the simulation of weather effects on LIDAR performance [76]. This level of simulation can verify that the underlying principles of the sensors are sufficient to meet the performance requirements of the HAD function. Likewise, they can also shed insight into potential "blind spots" of the sensing principles which in turn can be used during system design, when analysing the propagation of sensing failures during the system. This method of simulation is typically extremely computing resource-intensive and is therefore not typically used for performing large-scale testing but instead for analysing specific physical properties of the system during the design phase.
- **Simulation for verifying decision functions:** At present, the most common form of simulation involves simulating the decision functions using an abstract representation of the environment (e.g. in the form of object lists, road models and simulated traffic behaviour). This type of testing is used to collect evidence of the functional correctness of the driving properties of the system in a wide range of simulated traffic scenarios. A key technical challenge to this type of simulation is the creation of realistic models of the environment, including the behaviour of other traffic participants that can react to decisions of the ego-vehicle in a closed-loop simulation [77]. In their publicly available Voluntary Safety Self-Assessments, a number of companies have emphasised the use of simulation to create safety evidence. Waymo [78] describes how the use of simulation is used to demonstrate that the vehicle masters the 28 core competencies defined by the US Department of Transportation. The simulation is based on a high resolution model of a geo-fenced area in which the vehicles are tested on the roads. Scenarios recorded during on-road testing are then digitalised and used to create large numbers of variations to explicitly cover the core competencies.

7.6 Vehicle and field-based testing

As discussed in section 7.5, simulation and component-based tests are well suited for collecting large amounts of verification evidence either synthetically generated or derived from previously observed situations. As such, the testing is well suited to confirming the system behaviour in the presence of known triggering events. In order to develop a claim that the probability of previously unknown triggering events is sufficiently low, testing in the vehicle under real-world conditions is unavoidable. The relationship between field-based testing, controlled vehicle tests (e.g. using dummy objects on a test track), and simulation is summarised in **Figure 15**.

The objectives of in-vehicle and field-based testing are therefore to collect evidence of the safe operation of the system within the target domain. Due to the rare occurrence of hazardous situations (e.g. linked to unknown triggering events), an infeasibly large amount of driving data would be required to quantify the probability of hazards.

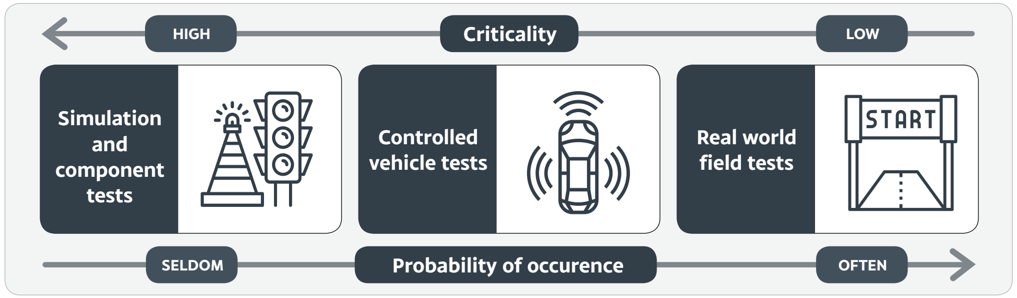


Figure 15: Relationship between simulation, controlled vehicle tests, and field tests as part of a comprehensive test strategy

However, if data from the extensive sensor set of the vehicle, in combination with observations of the internal system state, can provide information regarding how close a vehicle was to a hazardous situation, then this could provide a richer set of data that can be extrapolated to estimate the probability of a violation of safety goals. One such indicator could be a calculated time-to-collision, or the use of other vehicle sensor data as reference value. Extreme value theory (EVT) [79], [80], [81] has been proposed as a method for performing such evaluations. In EVT, extreme events are modelled using a statistical distribution which is then used to perform extrapolation.

Due to the characteristics of the open context domain, it may be misleading to discuss testing completeness criteria in this testing phase. Instead criteria must be found to form an argument that the residual failure rate is sufficiently low (e.g. several orders of magnitude of average accident rates due to human error). An estimation of "accident free miles" and the extrapolation of results based on EVT could be extended with the following additional criteria:

- **Disengagement index:** This metric describes the number of disengagements over time or per distance travelled of an automated driving system where the system itself disengages due to detecting the limits of its ODD or a supervising driver disengages. The number of disengagements of automated driving vehicles is required by the California Code of Regulations and has therefore provided insight into the maturity of systems from various companies. The advantage of this metric is that it counts more frequent statistical events (potential but avoided accidents) than actual accidents. However the actual achieved metric is highly sensitive to the difficulty and distribution of the scenarios driven. Furthermore, the relationship between avoided accidents and actual accidents is unlikely to hold for a number of corner cases (which may be uncontrollable for a human supervisor).

- **ODD coverage:** Testing coverage of the ODD and validating the completeness of the ODD often go hand-in-hand during vehicle testing. Vehicle and field tests should demonstrate the coverage of the defined ODD in all its relevant dimensions, while simultaneously identifying as yet undiscovered scenario classes or relevant domain characteristics. In both cases, a semantic model of the domain in which the target ODD is specified and a means of transferring real-world observations into this model in order to measure coverage or detect gaps are required. This leads to an observability problem whereby as yet unknown triggering events may not be directly measured by the vehicle's target sensor set and decision algorithms and whose presence must therefore be indirectly inferred. Aiming for ODD coverage also highlights the controllability problem whereby many situations stipulated in the ODD model may be extremely rare or difficult and dangerous to reproduce. This problem is often solved through testing within controlled environments such as in proving grounds under artificially generated conditions (such as water spray jets and cardboard cut-out pedestrians). Relevant testing criteria related to ODD coverage could therefore be percentage of ODD scenarios/equivalence classes covered or number of driving miles between newly discovered ODD requirements.

8. Conclusions

This report has described a framework for assuring the safety of highly automated driving, with particular focus on the topic of functional insufficiencies caused by the inherent complexity and uncertainty in the operating domain, sensing technologies and decision algorithms. It was argued that only a holistic approach to forming a safety argument based on the systematic analysis and modelling of the target domain, a demonstrably robust technical safety concept and diverse set of verification and validation evidence will lead to a sufficiently convincing assurance case for the system. In particular, the report described how understanding the relationship between domain analysis, system design, verification and validation evidence and the assurance case is crucial in order to produce economically feasible and adequately safe systems. For example, restricting the domain in which the system operates may allow for a stronger safety argument to be created with less effort than for a wider scope of operation. However, this introduces the need for additional arguments that the system will only ever operate within the restricted domain and that the boundary of the intended scope of operation can be detected.

The report has identified a number of areas of additional research both at the detailed methodological level and regarding the overall assurance strategy that must be addressed by academia and industry. These include, among others:

- Development of a common, standardised semantic model of the ODD that can be used for system specification, simulation, test and field data analysis;
- Extension to existing safety analysis approaches to address functional insufficiencies and not just traditional fault models;
- Approaches for collecting reliable evidence for the robustness of machine learning functions over a wide range of situations;
- Technical approaches to detecting the boundary of the target operational domain in order to transition to a safe state when the vehicle exits the domain scope for which it has been released;
- Approaches for assessing the safety risk and mitigating against hazards associated with the interaction between the automated system and the driver and between the system and its environment including pedestrians and other vehicles;
- Definition and agreement of a societally acceptable level of residual risk associated with different classes of system failures. The acceptable level of residual risk may vary with the failure classes and is unlikely to be directly related to the level of risk associated with human-operated vehicles.

Due to the complexity and scope of the field of work described here, this report is inevitably incomplete. Each component of the assurance case methodology described here is in itself an extremely active and often very early field of research. The report has therefore not attempted to provide a complete and up-to-date description but instead describes the underlying principles of the methodological components and their contribution to the overall assurance case. In addition, a number of relevant topics have not yet been addressed but should be pursued in future releases of this report or related work.

These topics include:

- Arguing the safety of human driver/automated system interactions, in particular in critical situations and during handover of tasks;
- Analysis of emerging properties and associated hazards due to the interaction between the automated driving system and its environment such as pedestrians, other road users and automated systems;
- The use of infrastructure to support the safety of automated driving and additional challenges for safety assurance;
- Addressing the issue of allocated moral and legal responsibility to automated systems and the consequences on the use of machine learning, system design and safety assurance.

Finally, the capabilities described in this report will need to be developed over time to allow for the effectiveness of the methods to be demonstrated in systems within restricted domains or with a lower level of criticality or automation before being applied to more sophisticated systems. Consensus must be formed through industry-wide collaboration (e.g. within publicly funded projects) and a proactive dialogue with homologation and legislative authorities. Eventually, best practice must be represented in appropriate standards. These standards, however, should focus on the requirements to be fulfilled by the assurance strategy rather than the specific methods used, as it is likely that these methods will continue to evolve over time as theoretical and tooling advances are made, leading to more economical approaches to reaching an equivalent level of assurance.

9. Bibliography

[1] "SAE J3016 Surface Vehicle Recommended Practice, (R) Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles," SAE, 2018.

[2] ISO, ISO 26262: Road vehicles - Functional safety, Geneva, Switzerland: ISO, 2011.

[3] B. Spanfelner, D. Richter, S. Ebel, U. Wilhelm, W. Branz and C. Patz, "Challenges in applying the ISO 26262 for driver assistance systems," in Tagung Fahrerassistenz, 2012.

[4] ISO, ISO/PRF PAS 21448: Road vehicles - Safety of the intended functionality, Geneva, Switzerland: ISO, 2018.

[5] IEEE, IEEE Standard Adoption of ISO/IEC 15026-1 - Systems and Software Engineering - Systems and Software Assurance, New York, USA: IEEE, 2014.

[6] G. Macher, O. Veledar, M. Bachinger, A. Kager, M. Stolz and C. Kreiner, "Integration Analysis of a Transmission Unit for Automated Driving Vehicles," in Gallina B, Skavhaug A, Schoitsch E, Bitsch F (eds) Computer Safety, Reliability, and Security. SAFECOMP 2018. Lecture Notes in Computer Science, vol 11094.Springer, Cham, 2018.

[7] K. Nidhi and S. M. Paddock, "Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?," Transportation Research Part A: Policy and Practice, vol. 94, pp. 182-193, 2016.

[8] A. Der Kiureghian and O. Ditlevsen, "Aleatory or epistemic? Does it matter?," Structural Safety, vol. 31, no. 2, pp. 105-112, 2009.

[9] J. C. G. Markus Maurer, Autonomes Fahren: Technische, rechtliche und gesellschaftliche Aspekte, Springer Vieweg, 2015.

[10] W. Damm, S. Kemper, E. Möhlmann, T. Peikenkamp and A. Rakow, "Using Traffic Sequence Charts at the development of HAVs," in Embedded Real Time Software and Systems-ERTS2018, 2018.

[11] Z. Porter, I. Habli, H. Monkhouse and J. Bragg, "The moral responsibility gap and the increasing autonomy of systems," in International Conference on Computer Safety, Reliability, and Security, 2018.

[12] W. Damm and R. Galbas, "Exploiting Learning and Scenario-based Specification Languages for the Verification and Validation of Highly Automated Driving," in 2018 IEEE/ACM 1st International Workshop on Software Engineering for AI in Autonomous Systems (SEFAIAS), 2018.

[13] S. Shalev-Shwartz, S. Shammah and A. Shashua, "On a Formal Model of Safe and Scalable Self-driving Cars," arXiv preprint arXiv:1708.06374, 2018.

[14] S. Ulbrich, T. Menzel, A. Reschka, F. Schuldt and M. Mauer, “Defining and Substantiating the Terms Scene, Situation, and Scenario,” in IEEE 18th International Conference on Intelligent Transportation Systems, 2015.

[15] S. Geyer, M. Blatzer, B. Franz, S. Hakuli, M. Kauer, M. Kienle, S. Meier and et al., “Concept and development of a unified ontology for generating test and use-case catalogues for assisted and automated vehicle guidance.,” IET Intelligent Transport Systems, vol. 8, no. 3, pp. 183-189, 2014.

[16] K. Czarnecki, “Operational World Model Ontology for Automated Driving Systems – Parts 1 and 2,” Waterloo Intelligent Systems Engineering Lab (WISE) Report, University of Waterloo, 2018.

[17] G. Bagschik, T. Menzel and M. Maurer, “Ontology based scene creation for the development of automated vehicles,” in 2018 IEEE Intelligent Vehicles Symposium (IV), 2018.

[18] N. Leveson, Engineering a safer world: Systems thinking applied to safety, MIT Press, 2011.

[19] A. Abdulkhaleq, D. Lammering, S. Wagner, J. Röder, N. Balbierer, T. Raste and H. Boehmert, “A Systematic Approach Based on STPA for Developing a Dependable Architecture for Fully Automated Driving Vehicles,” in 4th European STAMP Workshop 2016, 2017.

[20] C. Bergenheim, R. Johansson, A. Söderberg, J. Nilsson, J. Tryggvesson, M. Törngren and S. Ursing, “How to reach complete safety requirement refinement for autonomous vehicles,” in CARS 2015-Critical Automotive applications: Robustness & Safety, 2015.

[21] K. Attwood, T. Kelly and J. McDermid, “The use of satisfaction arguments for traceability in requirements reuse for system families,” in Proceedings of the International Workshop on Requirements Reuse in System Family Engineering, Eighth International Conference on Software Reuse, 2004.

[22] P. Koopman and M. Wagner, “Challenges in autonomous vehicle testing and validation,” SAE International Journal of Transportation Safety, vol. 4, no. 1, pp. 15-24, 2016.

[23] D. C. Schmidt, “Model-driven engineering,” IEEE Computer, vol. 39, no. 2, pp. 25-31, 2006.

[24] S. Burton and A. Habermann, “Automotive Systems Engineering and Functional Safety: The Way Forward,” in ERTS 2012 - Embedded Real Time Software and Systems, 2012.

[25] S. Friedenthal, A. Moore and R. Steiner, A practical guide to SysML: the systems modeling language, Morgan Kaufmann, 2014.

[26] Architecture Analysis & Design Language (AADL), Standard AS5506B, SAE International, 2012.

[27] P. Cuenot, P. Frey, R. Johansson, H. Lönn, Y. Papadopoulos, M.-O. Reiser, A. Sandberg and et al, "The east-adl architecture description language for automotive embedded software," in Dagstuhl Workshop on Model-Based Engineering of Embedded Real-Time Systems, Berlin, Heidelberg, Springer, 2007, pp. 297-307.

[28] P. Munk, A. Abele, E. Thaden, A. Nordmann, R. Amarnath, M. Schweizer and S. Burton, "Semi-automatic safety analysis and optimization," in Proceedings of the 55th Annual Design Automation Conference, 2018.

[29] B. Meyer, "Applying Design by Contracts," Computer, vol. 25, no. 10, pp. 40-51, 1992.

[30] I. Sljivo, B. Gallina and B. Kaiser, "Assuring degradation cascades of car platoons via contracts," in International Conference on Computer Safety, Reliability, and Security, 2017.

[31] D. Schneider and M. Trapp, "Conditional safety certification of open adaptive systems," ACM Transactions on Autonomous and Adaptive Systems (TAAS), vol. 8, no. 2, p. 8, 2013.

[32] B. Zimmer, S. Bürklen, M. Knoop, J. Höfflinger and M. Trapp, "Vertical safety interfaces–improving the efficiency of modular certification," in International Conference on Computer Safety, Reliability, and Security, 2011.

[33] A. Pnueli, "The temporal logic of programs," in 18th Annual Symposium on Foundations of Computer Science, 1977.

[34] R. Alur and T. A. Henzinger, "Real-time logics: Complexity and expressiveness," Information and Computation, vol. 104, no. 1, pp. 35-77, 1993.

[35] A. Cimatti and S. Tonetta, "Contracts-refinement proof system for component-based embedded systems," Science of Computer Programming, vol. 97, pp. 333-348, 2015.

[36] M. Grabowski, B. Kaiser and Y. Bai, "Systematic Refinement of CPS Requirements using SysML," in Modellierung 2018, 2018.

[37] B. Kaiser, P. Liggesmeyer and O. Mäckel, "A new component concept for fault trees," in Proceedings of the 8th Australian workshop on Safety critical systems and software, 2003.

[38] Y. Papadopoulos, M. Walker, D. Parker, E. Rüde, R. Hamann, A. Uhlig, U. Grätz and L. Rune, "Engineering failure analysis and design optimisation with HiP-HOPS.," Engineering Failure Analysis, vol. 18, no. 2, pp. 590-608, 2011.

[39] R. Johansson, A. Samieh, S. Bengtsson, C. Bergenheim, O. Bridal, A. Cassel, D.-J. Chen and et al., "A Strategy for Assessing Safe Use of Sensors in Autonomous Road Vehicles," in International Conference on Computer Safety, Reliability, and Security, 2017.

[40] M. Hörwick and K.-H. Siedersberger, "Strategy and architecture of a safety concept for fully automatic and autonomous driving assistance systems," in Intelligent Vehicles Symposium (IV), IEEE, 2010, pp. 955-960.

[41] G. Weiss, P. Schleiss, D. Schneider and M. Trapp, “Towards Integrating Undependable Self-Adaptive Systems in Safety-Critical Environments,” in ACM/IEEE 13th International Symposium on Software Engineering for Adaptive and Self-Managing Systems, 2018.

[42] B. Cheng, K. Eder, M. Gogolla, L. Grunske, M. Litoiu, H. Müller, P. Pelliccione, A. Perini, Q. Naumann, B. Rumpe, D. Schneider, F. Trollmann and N. Villegas, “Using Models at Runtime to Address Assurance for Self-Adaptive Systems,” in Models@ run. time, Springer, Cham., 2014, pp. 101-136.

[43] A. Aniculauesei, J. Grieser, A. Rausch, K. Rehfeldt and T. Warnecke, “Towards A Holistic Software Systems Engineering Approach for Dependable Autonomous Systems,” in 018 ACM/IEEE 1st International Workshop on Software Engineering for AI in Autonomo, 2018.

[44] M. Di Natale and A. L. Sangiovanni-Vincentelli, “Moving from federated to integrated architectures in automotive: The role of standards, methods and tools.,” Proceedings of the IEEE, vol. 98, no. 4, pp. 603-620, 2010.

[45] P. Schleiss, C. Drabek, G. Weiss and B. Bauer, “Generic Management of Availability in Fail-Operational Automotive Systems,” in International Conference on Computer Safety, Reliability, and Security, 2017.

[46] K. R. Varshney, “Engineering safety in machine learning,” in Information Theory and Applications Workshop (ITA), 2016.

[47] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman and D. Mané, “Concrete problems in AI safety,” arXiv preprint arXiv:1606.06565, 2016.

[48] D. Sculley, G. Holt, D. Golovin, E. Davydov, T. Phillips, D. Ebner, V. Chaudhary, M. Young, J.-F. Crespo and D. Dennison, “Hidden Technical Debt in Machine Learning Systems,” in Advances in neural information processing systems (pp. 2503-2511), 2015.

[49] A. Nguyen, J. Yosinski and J. Clune, “Deep neural networks are easily fooled: High confidence predictions for unrecognizable images.,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 427-436, 2015.

[50] A. Kurakin, I. Goodfellow and S. Bengio, “Adversarial examples in the physical world.,” arXiv preprint arXiv:1607.02533, 2016.

[51] J. H. Metzen, T. Genewein and B. Bischoff, “On Detecting Adversarial Pertubations,” in 5th International Conference on Learning Representations, 2017.

[52] H. W. Lin, M. Tegmark and D. Rolnick, “Why does deep and cheap learning work so well?,” Journal of Statistical Physics, vol. 168(6), pp. 1223-1247, 2017.

[53] S. Burton, L. Gauerhof and C. Heinzemann, “Making the case for safety of machine learning in highly automated driving,” in Making the case for safety of machine learning in highly automated driving, 2017.

[54] L. Gauerhof, P. Munk and S. Burton, “Structuring validation targets of a machine learning function applied to automated driving,” in International Conference on Computer Safety, Reliability, and Security, 2018.

[55] S. Shafei, S. Kugele, O. H. Mohd and A. Knoll, “Uncertainty in Machine Learning: A Safety Perspective on Autonomous Driving,” in International Conference on Computer Safety, Reliability, and Security, 2018.

[56] C.-H. Cheng, G. Nuhrenberg and H. Yasuoka, "Runtime Monitoring Neuron Activation Patterns," in arXiv preprint arXiv:1809.06573, 2018.

[57] K. Bengler, K. Dietmayer, B. Farber, M. Maurer, C. Stiller and H. Winner, "Three decades of driver assistance systems: Review and future perspectives," IEEE Intelligent Transportation Systems Magazine, vol. 6, no. 4, pp. 6-22, 2014.

[58] D. Åsljung, J. Nilsson and J. Fredriksson, "Using Extreme Value Theory for Vehicle Level Safety Validation and Implications for Autonomous Vehicles," IEEE Transactions on Intelligent Vehicles, vol. 2, no. 4, pp. 288-297, 2017.

[59] International Organization for Standardization, "ISO/IEC 9126 "Software engineering – Product quality"," 2001.

[60] R. Simmons, C. Pecheur and G. Srinivasan, "Towards automatic verification of autonomous systems," in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000), 2000.

[61] M. Fisher, L. A. Dennis and M. P. Webster, "Verifying autonomous systems," Communications of the ACM, vol. 56, no. 9, pp. 84-93, 2013.

[62] C.-H. Cheng, G. Nuhrenberg, C.-H. Huang, H. Ruess and H. Yasuoka, "Towards Dependability Metrics for Neural Networks," in 16th ACM-IEEE International Conference on Formal Methods and Models for System Design (MEMOCODE 2018), 2018.

[63] J. Attenberg, P. Ipeirotis and F. Provost, "Beat the machine: Challenging humans to find a predictive model's "unknown unknowns"," Journal of Data and Information Quality (JDIQ) , vol. 6, p. 1, 2015.

[64] K. Simonyan, A. Vedaldi and A. Zisserman, Deep inside convolutional networks: Visualising image classification models and saliency maps, arXiv preprint arXiv:1312.6034, 2013.

[65] L. A. Hendriks, Z. Akata, M. Rohrbach, J. Donahue, B. Schiele and T. Darrel, "Generating visual explanations," in European Conference on Computer Vision, 2016.

[66] A. Fawzi, O. Fawzi and P. Frossard, "Analysis of classifiers' robustness to adversarial perturbations," Machine Learning, vol. 107, no. 3, pp. 481-508, 2018.

[67] S. R. Richter, V. Vineet, S. Roth and V. Koltun, "Playing for data: Ground truth from computer games," in European Conference on Computer Vision, 2016.

[68] K. Pei, Y. Cao, J. Yang and S. Jana, "Deepxplore: Automated whitebox testing of deep learning systems," in Proceedings of the 26th Symposium on Operating Systems Principles, 2017.

[69] Y. Sun, X. Huang and D. Kroening, Testing Deep Neural Networks, arXiv preprint arXiv:1803.04792 , 2018.

[70] A. Gaidon, Q. Wang, Y. Cabon and E. Vig, "Virtual worlds as proxy for multi-object tracking analysis," in 2016, Proceedings of the IEEE conference on computer vision and pattern recognition.

[71] R. German, L. Sellart, J. Materzynska, D. Vazquez and A. M. Lopez, "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016.

[72] Z. Pezzementi, T. Tabor, S. Yim, J. K. Chang, B. Drozd, D. Guttendorf, M. Wagner and P. Koopman, “Putting image manipulations in context: robustness testing for safe perception,” in 2018 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), 2018.

[73] D. Gruyer, M. Grapinet and P. De Souza, “Modeling and validation of a new generic virtual optical sensor for ADAS prototyping,” in IEEE Intelligent Vehicles Symposium, 2012.

[74] D. Hospach, S. Mueller, O. Bringmann, J. Gerlach and W. Rosenstiel, “Simulation and evaluation of sensor characteristics in vision based advanced driver assistance systems,” in 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), 2014.

[75] L. Hammarstrand, M. Lundgren and L. Svensson, “Adaptive radar sensor model for tracking structured extended objects,” IEEE Transactions on Aerospace and Electronic Systems, vol. 48, no. 3, pp. 1975-1995, 2012.

[76] R. Rasshofer, M. Spies and H. Spies, “"Influences of weather phenomena on automotive laser radar systems,” Advances in radio science, vol. 9, no. B 2, pp. 49-60, 2011.

[77] V. Punzo and B. Ciuffo, “Integration of driving and traffic simulation: Issues and first solutions,” IEEE transactions on intelligent transportation systems, vol. 12, no. 2, pp. 354-363, 2011.

[78] Waymo, “Waymo Safety Report - On the Road to Fully Self-Driving,” 2017.

[79] P. Songchitruksa and A. P. Tarko, “The extreme value theory approach to safety estimation,” Accident Analysis & Prevention, vol. 38, no. 4, pp. 811-822, 2006.

[80] A. P. Tarko, “Use of crash surrogates and exceedance statistics to estimate road safety,” Accident Analysis & Prevention, vol. 45, pp. 230-240, 2012.

[81] J. K. Jonasson and H. Rootzén, “Internal validation of near-crashes in naturalistic driving studies: A continuous and multivariate approach,” Accident Analysis & Prevention, vol. 62, pp. 102-109, 2014.

UNIVERSITY of York

ASSURING AUTONOMY

INTERNATIONAL PROGRAMME

york.ac.uk/assuring-autonomy