Challenges in applying the ISO 26262 for driver assistance systems

Bernd Spanfelner, TÜV SÜD Automotive GmbH, Garching;
Detlev Richter, TÜV SÜD Automotive GmbH, Garching;
Dr. **Susanne Ebel**, Robert Bosch GmbH, Leonberg;
Dr. **Ulf Wilhelm**, Robert Bosch GmbH, Leonberg;
Dr. **Wolfgang Branz**, Robert Bosch GmbH, Schwieberdingen;
Carsten Patz, Robert Bosch GmbH, Schwieberdingen;

Abstract

The development of electronic, electric and programmable electronic (E/E/PE) systems is amongst other things subject to the IEC 61508 for the consideration of functional safety. In this, functional safety aims for the correct functioning of a technical system with the goal of avoiding potential safety critical situations caused by HW and SW failures. What generally is not considered in the safety standards is the prevention and restriction of safety critical situations based on the functional insufficiency of the driver assistance systems (DAS). The automobile specific characteristic of the IEC 61508, the ISO 26262, is no exception in this regard. However especially radar-, video-, or ultrasound-based functions can additionally cause potential safety critical situations coming from weaknesses in the estimation, interpretation and prediction steps necessary to realize driver assistance behavior. In this case the consequences are comparable to those of HW and SW failures and may also be safety critical.

From our understanding these weaknesses are fundamental and not avoidable – no matter what future developments in sensor technology and computing power we will see. Especially the necessary interpretation of other traffic participants' actions and the prediction of their future behavior will never be sufficiently complete to avoid misbehavior under all circumstances.

1 Introduction

The increasing need for safety applications in cars together with a high demand for unique selling points drive the development of driver assistance systems (DAS). With a growing number of road users and cars becoming faster and more powerful, today's traffic participants encounter more and more dangerous situations.

Driver assistance systems intend to support the driver in situations with a potential risk for accidents. They continuously observe and analyze the driving situation and intervene in order to clear dangerous driving situations.

Observing and analyzing needs advanced models not only for the mapping of the surrounding traffic situation through sensor measurements, but also to support a kind of 'understanding the situation'. Available driver assistance functions like ABS and ESP rely for their basic understanding of the vehicles driving state on physical models. These physical models are complete enough to demonstrate the absence of unsafe behavior of the system design. Complete technical specifications can be derived which are needed to demonstrate the correctness of the systems implementation.

Driver assistance systems reacting on their surrounding environment depend on more sophisticated models. To generate useful behavior the measurements are not only used to estimate object attributes, they also need to be interpreted in terms capable of describing all relevant traffic situations. From this 'basic' interpretation predictions need to be made to anticipate the behavior of the driver or other traffic participants. The underlying models are based on assumptions – which will not be true in all relevant situations. Without these assumptions however a timely action of the driver assistance system will never be feasible.

Using assumptions ultimately means, that cases are known where – with a given probability – unwanted behavior is generated,

This unwanted system behavior can also cause potential safety critical situations similar to HW or SW failures. To support the prevention or at least the control of HW and SW failures the ISO 26262 as a new standard for functional safety especially for passenger cars was published in November 2011. This article discusses the impact of models based on assumptions introducing the above discussed weaknesses to the development lifecycle of the ISO 26262 and possible implications of this impact. In the following this kind of models are called 'insufficient'. Furthermore, we outline an option for the treatment of the like systems.

2 ISO 26262

The International Standard (IS) of ISO 26262 [1] is the adaption from IEC 61508 [2] for the automotive industry. IEC 61508 is titled 'Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems (E/E/PE, or E/E/PES)' and is intended to be a basic functional safety standard applicable to all kinds of industry. It defines functional safety as: 'part of the overall safety relating to the EUC (Equipment Under Control) and the EUC control system which depends on the correct functioning of the E/E/PE safety-related systems, other technology safety-related systems and external risk reduction facilities.'

ISO 26262 is titled 'Road Vehicles – Functional Safety' and defines methods and measures to be taken to develop safety relevant functions for vehicles up to 3.5t and 'applies to all activities during the lifecycle of safety related systems comprised of electrical, electronic and software components'. The main scope of ISO 26262 is to avoid E/E failures of these systems. Therefore this standard includes a guidance to avoid or control these systematic and random hardware failures by appropriate requirements and processes and to reduce the expected risk to an acceptable level concerning injury or death of human beings.

The ISO 26262 defines methods for classifying safety relevant E/E systems based on hazardous events which they may cause, resulting in the ASIL (Automotive Safety Integrity Level). With respect to this for the whole life cycle of the safety relevant system, measures

to be taken are given to ensure that such situations are avoided, or at least that their appearance is reduced to an acceptable minimum.

The key features of ISO 26262 are described in the introduction of each part and listed following:

- ISO 26262 provides an automotive safety lifecycle (management, development, production, operation, service, decommissioning) and supports tailoring the necessary activities during these lifecycle phases;
- ISO 26262 provides an automotive-specific risk-based approach to determine integrity levels [Automotive Safety Integrity Levels (ASIL)];
- ISO 26262 uses ASILs to specify applicable requirements of ISO 26262 so as to avoid unreasonable residual risk;
- ISO 26262 provides requirements for validation and confirmation measures to ensure a sufficient and acceptable level of safety being achieved;
- provides requirements for relations with suppliers.

In common with IEC 61508 the ISO 26262 uses also the concept of safety goals and a safety concept as follows [3]:

- a *Hazard Analysis and Risk Assessment (H&R)* identifies hazards and hazardous events that need to be prevented, mitigated or controlled;
- a *Safety Goal* is formulated for each hazardous event;
- an *Automotive Safety Integrity Level (ASIL)* is associated with each safety goal;
- the *Functional Safety Concept* is a statement of the functionality to achieve the safety goal(s) addressed by *Functional Safety Requirements*;
- the *Technical Safety Concept* is a statement of how this functionality is implemented on the system level by hardware and software addressed by *Technical Safety Requirements*; and
- *Software Safety Requirements* and *Hardware Safety Requirements* state the specific safety requirements which will be implemented as part of the software and hardware design.

Figure 1 shows the trace of the different safety requirements listed above. It contains also the design and test flow especially for the SW development. This trace is also pictured in [3] and here supplemented with additional information about the responsibility of the different levels of safety requirements.

ISO 26262 call the test phase on the top-level *system safety validation.* It aims to provide evidence of appropriateness for the intended use and aims to confirm the adequacy of the safety measures for a class or set of vehicles. Note that this does assure, that the safety goals are sufficient and have been achieved. That is, in particular the safety validation according to ISO 26262 cover possible failures due to insufficiencies in a systematic manner.

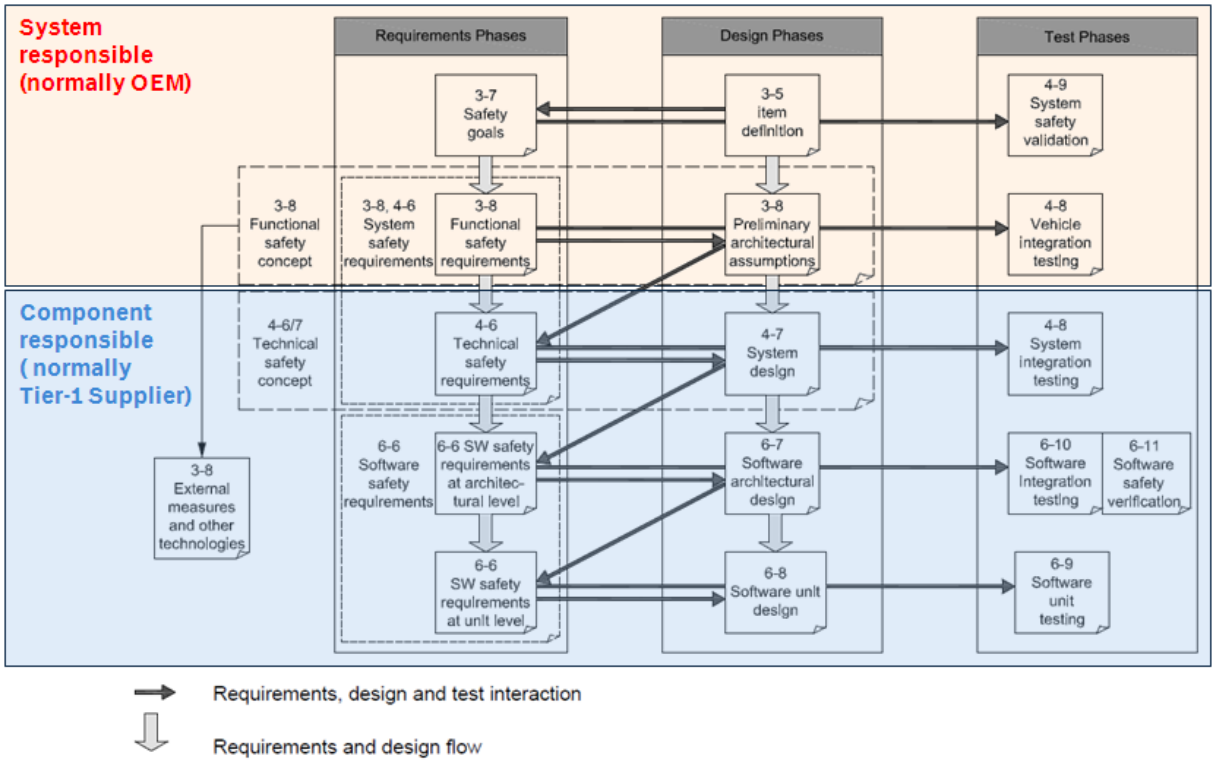

Figure 1: Safety requirements, design and test flow from concept to software

How does the ISO 26262 help with the above stated problem of using insufficient models? In our interpretation the ISO 26262 introduces methods to help circumvent problems with in a sense 'incomplete' hardware models: Estimating hardware failure rates by introducing probability to deal with missing knowledge about the hardware state of an individual system. State of the art methods to deal with modeling insufficiencies apparent in advanced driver system development however are – from our understanding – not mentioned in the ISO 26262 or other relevant standards.

3 Types of models

The investigation of the compliance of driver assistance systems to the ISO 26262:2011 is closely related to the role and idea of models.

3.1 The role of models in the development

In the development process models are used to specify behavior and to derive technical decisions. Models are mappings of the reality that focus on certain aspects that are of interest for a certain purpose. Thereby, models are abstractions, which drop irrelevant details in order to antagonize complexity by combining many elements that are distinguished only by irrelevant details into a single equivalence class. The term model itself is unspecific. During a development of a system, models are used in different ways potentially having different influence on the result. We give a brief taxonomy of models and their use in systems engineering before we discuss their properties and potential impacts on the ISO 26262 process. Figure 2

gives an outline over the use of models in a typical development process according to the V-Model. In the figure we can see two types of models which we define in the subsequent sections: the gray blocks on the left part of the V are system models that describe the system under development at different levels of abstractions whereas the white arrows represent deductive models that are used to relate the different levels of abstraction of the system models.

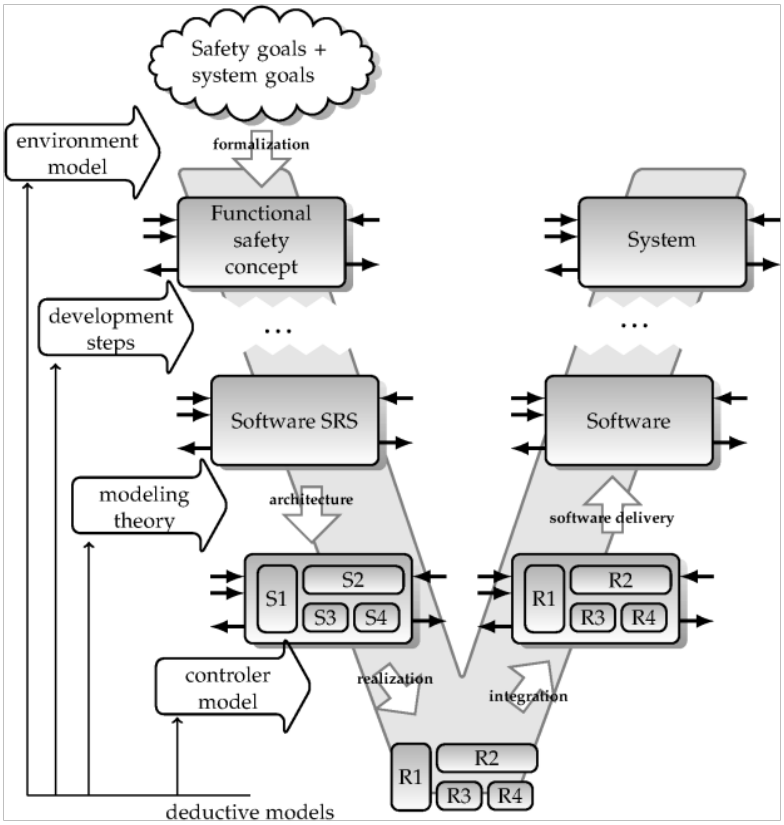

Figure 2: The use of models in a development process

3.2 System model

In the domain of software & systems engineering the notion of a model is used to describe an abstraction of the system under development. A system model in general is a collection of relations that map inputs to the system under considerations to the outputs of the like. This captures the behavior of the system under consideration. System models exist at different levels of abstraction and granularity each describing the system with respect to different aspects and containing different levels of detail. We regard a system model as a collection of mappings

$$f: I \to O$$

where I is the input domain of the system and O is the output domain.

While model based development often is reduced to automatic code generation and the use of certain tools, the actual meaning of a system model is much broader. In general system

models can be as simple as a textual description of the mapping (informal model) or as complex as a completely formal model with a mathematically well defined syntax and semantics. Any initial specification of the behavior of the system is a more or less formal model of the system.

In software & systems engineering, and according to the V-Model a number of consecutive system models are elaborated that describe the system under construction from different viewpoints and at different levels of detail. Any such system model shall be related by a proper refinement relation or shall be compatible for overlapping aspects of the views they represent.

The top level of this stack of system models captures the goals, the system serves. In the case of a safety related development these are the safety goals. Usually safety goals are independent of a certain solution or technology but depend on the functionality the system shall provide. The formulation of the safety goal already is a constraint for the system development and basically defines the problem space. As an example, we use the safety goal 'Unintended activation of the emergency brake must be prevented'.

Depending on the formulation of the safety goals, they can either be of general purpose or already contain functional aspects of the system. We are not going to discuss the pros and cons for each but rather note that different categories of safety goals exist.

In the stack of system models the first functional model is the system specification. With respect to the ISO 26262 initially this is the functional safety concept. The system specification describes the behavior of the system only at its interface to the environment in order to achieve the defined goals. In an ideal development, the system specification only describes all possible solutions without any implications about a certain realization. System models at lower levels refine their preceding model by add information e.g. about the architecture, separation in HW and SW, a deployment to ECUs, the choice of certain algorithms etc.

3.3 Deductive model

The concept of a model is also used in another context in software & systems engineering. Namely, the mappings that relate two consecutive refinements in a development (i.e. two models of the system) are called a model as well. This notion of a model is closely related to the notion of a model as used in e.g. physics that is used to give a simplified description of the nature in order to enable reasoning or prediction. For relations of technical the terms meta model or modeling theory exist and describe the set of all possible system models that can be expressed by a modeling technique. By relating the elements of the meta model, the mapping between two levels of abstractions can be established in a general way. This relation supports the deductions, respectively the technical decisions that are necessary in order to generate successor model.

What is often ignored in software & systems engineering is the fact that the activity of defining goals and a system specification already involves deductions and decisions. Basic knowledge and assumptions about the environment have been used to generate the system specification according to the defined goals. Going back to the example of a safety goal in the previous section, in order to be able to make the verifiable, it is necessary to define the term 'unintended' more precisely, possibly already in technical terms. One obvious option is to

refine the goal into a requirement like ‘the emergency brake must not be activated if no obstacle is detected’. This refinement involves a deduction that postulated the absence of an obstacle as a necessary condition for ‘unintended’.

Even more, already the initial verbalization of the goal is the result of a deduction. The ultimate goal is to avoid hazards for road users. Surely there are situations where an unintended braking does not endanger anybody. The choice of the safety goal above is the result of a deduction that uses the assumption that preventing the activation of an emergency brake even in situations where this prevention would not be necessary is a valid measure.

Usually the knowledge that is used to support a refinement is based on models as well. As we can see, the kind of information that is used for relating subsequent system models is very different. Therefore, we call models that are used to derive design decisions deductive *models* subsequently. Deductive models relate two models (c.f. Figure 3). If we regard two system models $A = (I_A, O_A(f_A))$ and $B = (I_B, O_B(f_B))$ as algebraic structures with I_x and O_x as the respective inputs and outputs and f_x as the respective set of functions, that are captured by the system models each, generally speaking a deductive model F is a function $A \rightarrow B$ that realizes two kinds of mappings:

- the mapping of the carrier sets of the two structures $I_A \rightarrow I_B$, $O_A \rightarrow O_B$ and
- the mapping of the function symbols of each algebraic structure $\mathrm{f_A} \rightarrow \mathrm{f_B}$

The properties of these mappings (e.g. partial mappings, surjectivity, injectivity or homomorphism) affect the system model that is created by the deduction as well as the relation between the two system models that is established by the deductive models and used for verification. Some of these properties are discussed in section 4.

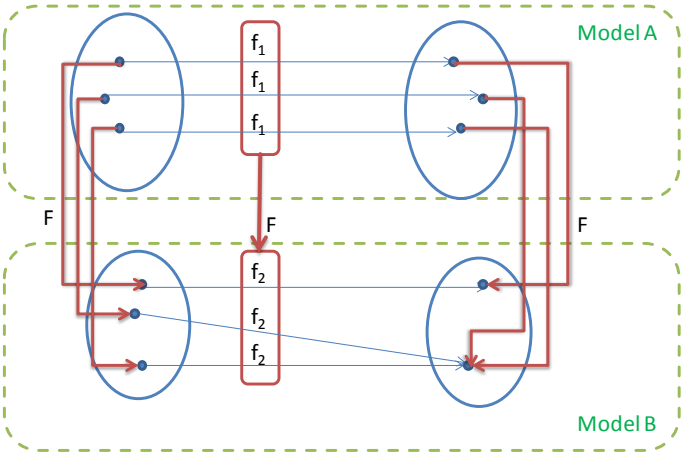

Figure 3: A deductive model relating two system models

In essence, a deductive model captures the knowledge a design decision is based on e.g. by relating the concept of an obstacle in the upper level model to a concrete representation of an obstacle as a certain object with a certain trajectory in the lower level. Of course this example requires a further deduction that substantiates the concept of an object.

For the consideration of driver assistance systems and the ISO 26262 the deductive model that is used to create the system specification out of the (safety) goals is of main interest.

This deductive model – in literature sometimes called the environment model – is used to capture the relevant relations in the environment in order to derive the appropriate system behavior. For driver assistance systems this often comprises models of the physics of the environment.

4 Properties of Models

Models are relations that are created for a certain purpose and include knowledge as well as assumptions. The quality of knowledge, the validity of the assumptions and the rate of abstraction (i.e. what details are dropped in order to generate equivalence classes) determine the accuracy and the expressiveness of the models. Properties of deductive models influence the properties of the system models in a development step: in some sense, the system models inherit the properties of the deductive models. Furthermore, the quality of the deductive model affects the quality, effectiveness and finally the validity of any verification of a development step. Thereby, deductive models are not created from scratch or without already having certain principles in mind. A deductive model that is used for the deduction of an airbag safety goal in a setting with acceleration as the indicator looks completely different from one that is used in a camera setting. Finally, without having a setting in mind, it may turn out that neither of both is appropriate and a completely different deductive model will fit best. Subsequently, we explain some properties that we regard as important in the context of the ISO 26262 and especially for driver assistance systems.

4.1 Complete models

We defined models as sets of functions. Every function has a domain and a range. Formally, a model is complete, if for every element of the domain some element in the range exists that is mapped. For deductive models this is a bit more complex. Complete deductive models are complete with respect to both mappings i.e.

- every element of the input and output domains set of the one model is related to an element of the carrier set of the other model, and
- for every function symbol of the one model some corresponding function symbol in the other model is mapped.

Partiality is special setting in information theory. A partial function is not defined for all elements of a given domain. This is in contrast to mathematics where a domain is defined to be the set for which a function is defined. The notion of partiality allows relating a universe of possible inputs to a functions actually accepted inputs. Only if a deductive model is complete and uniquely defined, an unambiguous development step is supported and a proper verification is possible.

For deductive models it is not always necessary to be complete. In contrast, a system model should be complete - at least at some time in the development - in order to allow a proper implementation, a comprehensive verification, and the derivation of appropriate tests. Note that verification only is possible for those inputs to a system where the system description defines some output. In order to achieve this, a deductive model should at least be subjective with respect to the input domain of the system model that is the result of the deduction.

4.2 Insufficient Models

In practice, deductive models are not always complete or uniquely defined nor are the applied abstractions always suitable. This especially affects physical models, models that resemble estimation, interpretation and prediction, and models that are used to capture natural language expressions like ‘dangerous situation’ or ‘accident is inevitable’. This causes different, sometimes negative effects in the development. We discuss some of the most important ones from the viewpoint of the ISO 26262 subsequently.

4.2.1 Oversimplified Models

A very good example for a simplification is the modeling of an accident by acceleration. The concept of an accident is mapped to a certain amount of acceleration with a certain vector but ignores any other aspects that are typical for accidents. An oversimplification does the same but excludes relevant discriminating factors.

Deductive models relate two system models, i.e. their functions and carrier sets. In case of the deductive model that is used to generate the system specification out of the goals this includes a mapping of the relevant aspects of reality into an input domain of the system. This is accompanied with an abstraction in order to counteract complexity.

An abstraction is a simplification that maps different elements of the domain to a single element in the range i.e. it generates equivalence classes. The elements of an equivalence class are thought to share relevant properties and only differ in those aspects that are not relevant. Therefore, it is assumed that they do not need to be distinguished. This is a valid approach to prevent the consideration of too many, often insignificant details. However, for many reasons, an abstraction may be inappropriate or even invalid. The result is a mapping such that elements that shall be distinguishable cannot be distinguished any more after the abstraction.

If such invalid abstraction is embedded in a deductive model, the inputs to the system appear equal for two elements that shall be distinguishable. As a result and since systems are deterministic, the system reacts to the inputs as defined in the system model according to the equivalence class. While for some inputs this still is correct with respect to the goals, for others this may be incorrect.

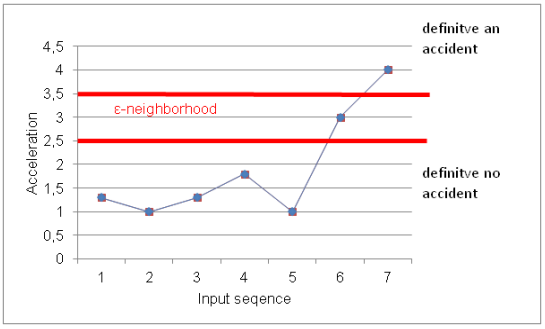

Figure 4: Ambiguous deductive model. Only the epsilon-neighborhood is ambiguous, other areas are definitive

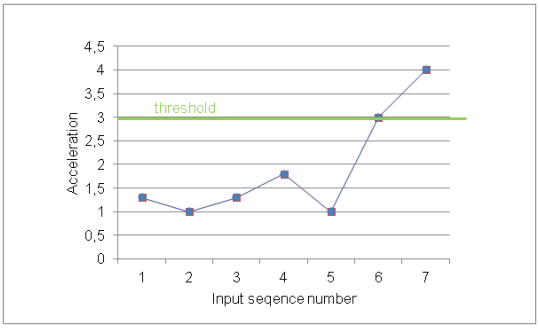

Figure 5: Mapping in the system model. The designers choose to activate the airbag if a certain threshold is exceeded

As an example take the airbag system as documented by a court decision [4]. An accident was abstracted to be recognized by certain acceleration. Surely, any accident will have certain acceleration but other events may have as well. By the deductive model, events that should be distinguishable were mapped to the same system input certain acceleration). As a result, the system model was designed to activate the airbag if the acceleration exceeds a certain value in a certain direction. In the case mentioned above, a pothole caused the same acceleration and triggered the airbag finally leading to an accident.

Many e.g. physics based deduction models used today are oversimplified because the number of influencing factors in reality is much too large to be considered in a reasoning. Similar, an estimation, interpretation or prediction can only be an approximation that considers a subset of all possible influencing factors. However, experience shows that oversimplification of many of those models only affects small areas or only areas which are usually irrelevant. We illustrate this in Figure 4 and Figure 5. The left figure demonstrates that there is a small area (we call this an epsilon neighborhood) where the acceleration is ambiguous for detecting an accident. Since a system is deterministic, the designers choose a threshold that is within the epsilon neighborhood and usually optimized for a safety goal. Since the epsilon neighborhood is sufficiently small, in practice the ambiguity of the deductive model has no significance.

4.2.2 Probabilistic Models

In our context probabilistic models are a special case of oversimplified models. If for technical reasons, e.g. by restrictions of available sensor technology, only an oversimplified deductive model is available, additional indicators are used. These indicators are not definitive. They rather raise the confidence in detecting a certain item in situations where sensor input is ambiguous.

While in practice probabilistic deductive models can be used to define system models that use additional input in order to determine the reactions which in turn raises the availability of the system, from a safety point of view the use of probabilistic models opens another dimension of complexity. Safety is all about probabilities that a hazardous event will injure or kill people. Since probabilistic is hard to deal with, the aspects that are regarded to be probabilistic in the ISO 26262 are restricted to hardware failure rates. For hardware failure rates, a sufficient amount of data is available that allows to judge on the confidence in the given rates.

For functional aspects of a systems that are the result of a deduction based on probabilities, no such data is available that is already accepted commonly. Therefore an approach demonstrating the appropriateness of the resulting system is necessary.

4.2.3 Underspecified models

Object recognition and classification problems are examples where underspecified deductive models are used. While it is impossible to describe all existing objects and their classes, a deductive models I) applies a simplification by working out discriminating factors and II) giv-

ing samples of the still large amount of combinations of these factors and their classification. Only pursuing steps generate a generalization of the classification that allows an application of the classification to arbitrary combinations of the discriminating factors while the correctness of this generalization remains a matter of further analysis.

In general, for underspecified models, one knows that there is a relation for all elements but due to complexity only samples are available. Often, the samples are generalized to build a deductive model that is used in a development step. This is iterated while adapting the deductive model by optimization. The performance of the deduction is evaluated by tests of the generated system model to comply with the initially given goals.

There are many approaches to generalize and optimize this kind of deductive models. Apart from analytical methods one notable is machine learning. Another is design by experiments. In any case, a function class is chosen to represent the deductive model. This function class has parameters that can be tuned. During the iterations the parameters are changed such that the system model that results from applying the deductive model performs well with respect to the given goals.

However, there is a drawback of this approach. Many factors influence the performance of the optimization and finally may result in deductive models that are oversimplifies or even wrong. Figure 6 and Figure 7 demonstrate this issue. The basic model was chosen to be a linear equation. By optimization, it was possible to find parameters that result in sufficiently precise results. However, the real relation was a sinus based one. No parameters exist that allow the optimization of a linear equation to approximate a sinus based equation.

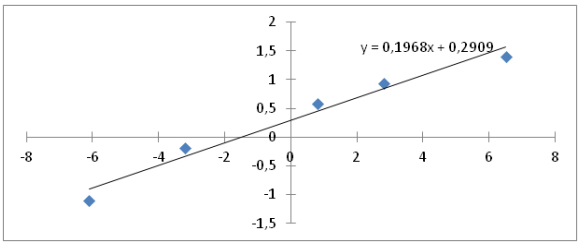

Figure 6: Samples for a deductive model and an optimized deductive model based on a linear equation

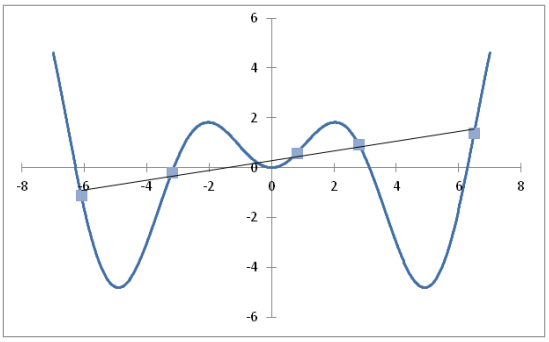

Figure 7: The actual function that was the basis for the samples

Another challenge is to avoid oversimplification. Usually a deductive model is underspecified because only insufficient data is available and the actual relations in reality are so complex that they cannot be completely captured in a model. Again abstraction is a valid countermeasure and any generalization that is used to complement underspecified models usually incorporates abstractions. However, the trick is to choose the generalization such that the resulting deductive model has no significant oversimplifications.

5 Insufficient Models in the ISO 26262

Insufficiencies of models are omnipresent. The example of the airbag already demonstrates that even supposedly mature models introduce hazards e.g. due to oversimplification. In the last chapter it was demonstrated that during the development of certain complex systems like surround sensor based DAS insufficiencies in deduction model and system models, respectively, occur. These insufficiencies are not due to technical faults but as a matter of principle by nature. Note in this connection that the choice of the sensor technology might reduce the insufficiencies. But in general it is not possible to eliminate these insufficiencies completely due to the typical trade-off between true and false positive by object recognition. To see this note first that in simple terms failures due to insufficiencies lead to false positives in object recognition. Figure 8 illustrates a schematic receiver operating characteristic of an object recognition system. Although optimizing the system might slightly change the actual characteristic, the main trade off remains and the demand for higher benefit will necessarily lead to certain false positives.

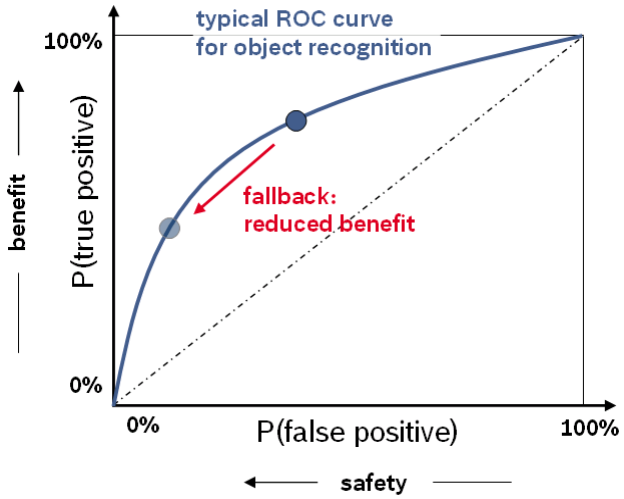

Figure 8: Schematic ROC curve for object recognition

The insufficiencies of deductive models have impact on the appropriateness of specifications in the setting of a safety related development. While models that are a priori available (i.e. are not underspecified) in principle are accessible to further analysis and the impact of possible oversimplification can be estimated in advance, underspecified models are only completed during design time and affect the effectiveness of verification measures. Subsequently, we discuss the cause and impact of insufficiencies – especially of underspecification and oversimplification – from the viewpoint of the ISO 26262.

5.1 The value of measures in the ISO 26262

In order to be able to discuss the impact of insufficient models on the development lifecycle, we shortly summarize the characteristics of this lifecycle. Thereby, we focus on the system part and the software part (Part 4 and 6 in the ISO 26262) of this lifecycle because most of the issues are relevant for these phases.

In the ISO 26262, the V-Model is used as a reference for the development process. Figure 9 presents an overview of the V-Model. The V-Model is organized into layers of abstraction of which each refines the specification of the previous layer and restricts the solution space by introducing design decisions. As already shown in Figure 2, each refinement may be (and usually is) guided by deductive models that correspond to the design decisions. Each layer serves as a specification for the next lower one and takes the role of a proof obligation for the subsequent development. Specific to the V-Model, and in this context especially important, is the tight integration of specification and verification on all layers.

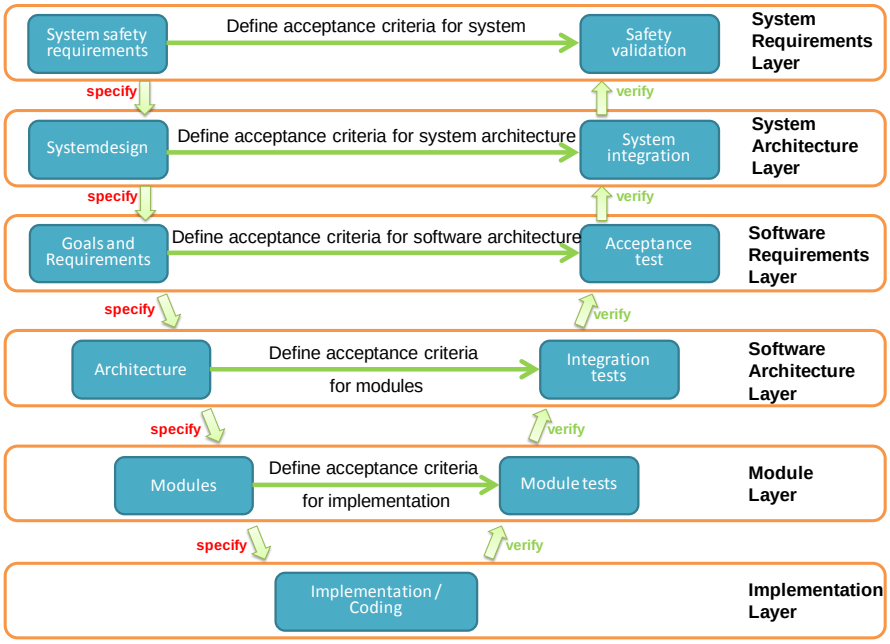

Figure 9: The V-Model and its layers of abstraction

The measures defined in the ISO 26262 aim at ensuring the quality of all (including intermediate) development work products by making them accessible to analysis and approval. By verification measures like reviews, simulation, (semi)formal verification and testing, the compliance of any subsequent development work products with the specification at each layer is demonstrated. In essence, these verification measures demonstrate that a development step indeed is a refinement and does not introduce additional behaviors nor does it deviate from already defined behaviors. The introduced refinements may only detail out a behavior without contradicting any already existing one.

For that purpose, a preferably extensive investigation is executed that comprises all aspects of a system's specification and compares the actual behavior with the supposed behavior. If all verification measures contribute to the argument for compliance, a work product is assumed to be correct with respect to the initial requirements. In the setting of safety related systems this includes the adherence to the safety goals.

5.2 The impact of insufficiency on meaningful specifications

A specification is an abstract description of the aspired behavior (possibly extended by non-functional properties) [5]. Each such description may already be based on assumptions and models, depending on the language and wording that is used. For example take the airbag with the two usual safety goals

- The airbag must open in the case of an accident
- The airbag must not open in the case of no accident

First, the first safety goal intends to prevent passenger from injuries. However, it is already a modeling assumption that activating the airbag prevents injuries. In fact there are situations where keeping the airbag closed even in the case of an accident would be better or activation at a different time or with a different intensity would ultimately protect the driver. The safety goal requiring an unspecific activation of the airbag in the case of an accident already is the result of a deduction from the superordinate goal to save passengers to a setting where airbags are used for this purpose. However, currently it is a commonly accepted model that activating the airbag is better than doing nothing in the average case.

Second, while defining an accident in terms of acceleration, which is quite common, already model knowledge is used again. The fact that acceleration is not always an adequate indicator do define an accident (i.e. it is an oversimplification) has been discussed above.

During every step over to the next, more detailed specification in the system development, the use of an insufficient model may result in a specification that is insufficient as well.

In contrast to the oversimplified models, the impact of the underspecification in the model cannot be judged in advance – even not theoretically.

We already explained that underspecified models only map samples. For a video driven driver assistance system e.g. this may be given as a database of thousands of hours of driving together with their classifications. Nevertheless, even such an amount comprises only a very small partitioning of all possible driving situations and nothing is specified yet that covers all those visual inputs and their classification that are not contained in the sample set.

While a complete but oversimplified model is available a priori, underspecified models are only complemented during development after relevant decisions were made:

- the complexity class is estimated and a general purpose model chosen accordingly
- large parts of the system were implemented in order to enable the optimization (e.g. by manual adjustment of parameters or machine learning)

Therefore, it is not possible to refer to experiences or results from physical experiments to justify the appropriateness or to judge the quality of the complementation a priori because a gap exists in the refinement chain. The gap is exemplary illustrated in Figure 10. In that figure the gap created by the use of a underspecified deductive model during a step over from the system safety requirements to the software safety requirements is illustrated. Similar gaps may be observed at any other development step as well. While the safety goals and even the system safety requirements are application specific (if a proper deduction was made), the steps in the development starting with the software requirements are only loosely related and have a general purpose character. Only the optimization of parameters after implementing parts of the algorithm makes these parts of the system application specific.

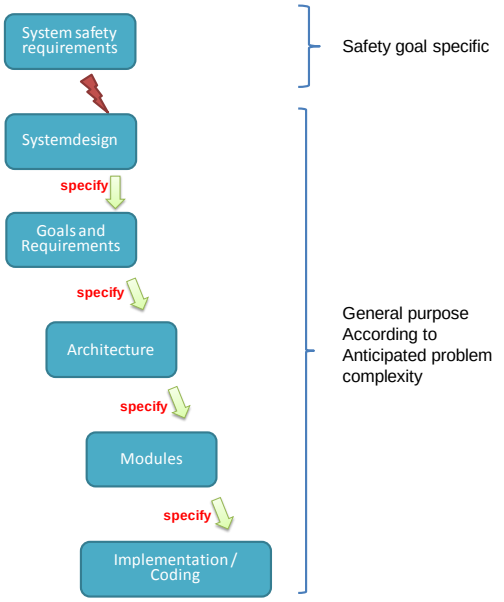

Figure 10: The gap between the safety goal and the software requirements specification

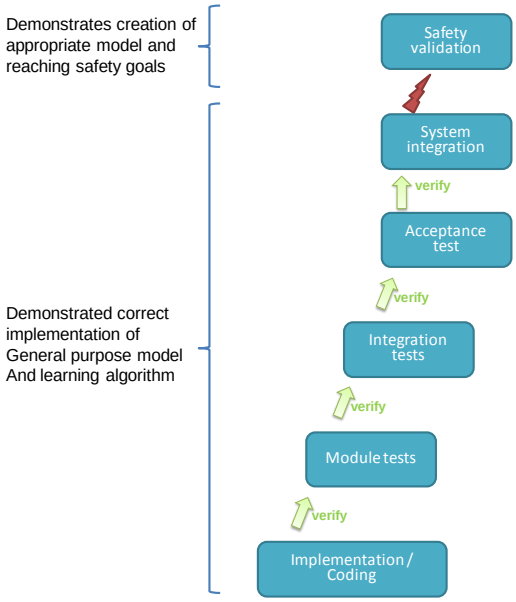

Figure 11: The verification measures only cover generic aspects but leave a gap to the safety goals

The result of this approach is a gap in the refinement chain of the V-Model. The safety goals cannot be directly transferred into functional requirements whose correct implementation directly implies the fulfillment of the safety goals. Only the sample data serve as a partial specification for the software development.

5.3 The impact of insufficiency on the effectiveness of verification

The approach in the ISO 26262 is a systematic refinement and approval of the development work products, including system specification architecture, unit design and implementation. The verification activities are based on the assumption that a mapping exists that lets comprehensively check for compliance between system models at different levels of abstractions. For complete or over simplified models this is the case: for every possible input a valid system reaction is defined and by verification it can be approved that this work product adheres to these defined system reactions. This allows verifying the correctness of every development step. The only (implicit) assumption in the ISO 26262 is that the deductive models that are used to relate to levels, are appropriate (e.g. demonstrated by experience or by proof).

The verification of a development step is done by demonstrating either by review, inspection, testing or even formal proof that the deduction is a homomorphism i.e. both system models create comparable reactions with respect to the mapping of the respective carrier sets i.e. the outputs of one system model can be related with the outputs of the other system model via the deductive model. What is not included in the verification is the appropriateness of grouping elements into an equivalence class. In an abstraction all the grouped elements need to share the same relevant properties. In the running example, this becomes obvious if two real world objects are mapped to an element that is represented by a certain pattern of

edges in a video signal. It may happen that one never will be an obstacle but due to an inappropriate equivalence class it is regarded as one.

The verification measures and tests on each layer in the development lifecycle are used to demonstrate that the corresponding specification is fulfilled. In a scenario with underspecified deductive models, classic module and integration tests can only demonstrate the correct implementation of the general purpose model (i.e. a model that comprises all assumptions) and, if applicable, a chosen learning algorithm. Both, reviews and tests cannot provide or contribute to the evidence of the correctness with respect to the safety goals as long as the validity of a deductive model has not been proven. Figure 11 illustrates the effect of the specification gap on the effectiveness of verification measures and is the pendant to Figure 10.

As an example, if using a neuronal network, it is of little use for demonstrating the correct implementation of the safety goals, if testing each and every untrained neuron. Code coverage criteria require this and indeed, it must be demonstrated that neurons behave like neurons. But no conclusion about the safety goals can be drawn unless it is demonstrated that the combination of the neurons is sufficient to learn about the relations in the system environment. Similar, for design by experiments, a demonstration of correctness by module tests covers only the correct implementation of the chosen parameterized function but to the correct choice of the parameters. Only the validation of the deductive model that is created a posteriori during the adaptation of the parameters affords this.

The initial system model was based only on the example data and does not allow deriving acceptance criteria other than exactly these example data. Technically, this allows only the demonstration of the compliance with the test samples but not with the safety goals in general. As a consequence, underspecified deductive models render any classic verification mechanism largely ineffective with respect to safety goals. The verification methods on the left side of the V only serve to demonstrate the correct implementation of the general purpose model. There is no means to demonstrate the compliance to the safety goal because the safety goals have not been refined into verifiable requirements, yet. Similar, on the right side of the V, the verification measures only serve to demonstrate the correct implementation of those aspects that were part of a system specification.

5.4 Appraisal of the impact

By applying the ISO 26262 we must not exclude functional performance and insufficiencies from the scope and therefore shall consider it in the 2nd edition of the ISO 26262. The above discussion demonstrates that insufficiencies are present in almost every function but are especially immanent for advanced driver systems because of the complexity of the used models. If one excludes the functional performance and the insufficiencies from the scope of the ISO 26262, to be consequent this must be done for the Airbag, the ESP etc. as well. As a result the ISO 26262 would be largely useless and inapplicable.

Deviations from the safety goal are no longer systematic errors in the literal sense from the viewpoint of the software design: any implementation that deals with the example data correctly is basically correct. However, from the viewpoint of the safety goals, there is indeed a systematic error: the deductive model may be chosen inappropriate or the sample data is insufficient such to result in violations of safety goals. Therefore, tests demonstrating the ap-

propriateness of the models must correspond to the level of system specification and safety goals (i.e. system validation). In order to support the understanding, we give a short summary for the comparison of both scenarios:

Classic situation (deductive model completely available a priory):

A deductive model exists that is known to be appropriate for the purpose and is used to define a system specification. The system specification can be verified to comply with the safety goals along the lines of the deductive model. The methods in the ISO 26262 ensure that the system specification, respectively the subsequent system models are implemented correctly.

Advanced driver assistance systems (underspecified models):

No complete deductive model exists a priory. Only sample data is given. A deductive model is generated during the development process by using a general purpose system model and applying the example data. The generated system model is correct by construction with respect to the example data but still needs to be proven to be adequate with respect to the safety goals. This includes a definition of the term 'adequate'.

The notion of adequacy is crucial. Adequacy means that the probability for residual errors in the deductive model is at an acceptable rate. Most deductive models (regardless if initially underspecified or not) have a certain amount of oversimplification or even errors. Even mature, a priori known and analyzed models may cause safety goals to be violated very easily. The example of the airbag incidence demonstrates that models, even if they are available a priori and used to generate complete system specifications are subject to insufficiencies. The facts that the model was known a priori, used in a complete specification and implemented correctly (at least not an error in the implementation caused the accident) did not prevent the accident.

Note that in a systems development it is possible to defer the decisions about a systems behavior to the late implementation. It is a matter of engineering to iterate the development and to make changes to the specification and/or implementation. However, with respect to the safety goals and their implementation, a system specification is required to be complete. However, this may be the case even though a deduction of technical (functional) requirements incorporates insufficiencies. As a result, an implementation is provable correct while subjectively violating safety goals because the basis of the proof is an insufficient deductive model.

6 Options for treating insufficient models

Systems based on complete models allow a verification of the implementation by executing the verification measures defined in the ISO 26262. But, as pointed out above, insufficient models break the line of arguments for an appropriate implementation. In this case, the correctness of the actual implementation of the deduced system specification can still be proven by established verification methods and tests. But since the actual system behavior essentially relies on the environment model that provides the basis for this system specification, insufficiencies in this deduction model might cause risks as critical as implementation faults or random hardware failures (or even more critical). Thus, the validity of the environment

model needs to be considered on its own. But unfortunately at this the ISO 26262 does not help: Although a failure rate on system level including systematic faults is mentioned (cf. ISO 26262-3, Annex B1), it does not provide any guidance to handle such problems. In particular, a quantitative consideration of failure rates, which to our conviction is indicated in this regard, only is required for random hardware failures.

6.1 Validation of system and deductive models

In fact, there is a subtle difference between validating the deductive model and an actual (sub) system model based on it. Consider an algorithm (being one part of a complex DAS) that classifies vehicles in 'cars' and 'others' based on object information like size, moving trajectory etc. extracted from a video sequence. (We do not consider the actual object recognition which is an even more challenging task.) Now, we might answer our question concerning validation in two ways. First, validating the deductive model means analyzing whether the decision 'car' based on size, trajectory, etc. is always correct. Second, we might validate the implemented algorithm itself by simulations based on real or synthetic data, i.e. we validate the system model. In the end, both approaches lead to the same result. They only differ in the way how to achieve this.

The latter might be ensured by a black box test of the subsystem considered. Carrying this approach to extremes we end up with the black box validation of the complete system. That is, evaluating the behavior of the complete system on a sufficiently large set of real data and calculating a significant total failure rate. In the first case we furthermore need to verify that the implementation of the algorithm is correct which might be obsolete in the second case, if the significance of the black box tests can be demonstrated.

In both cases the crucial point is that the validation must not be based on idealized data but on the actual output data of the preceding subsystems. This includes possible inaccuracies due to measurement errors and insufficient deduction models (in practice, the algorithm will never see idealized data!).

6.2 Possible arguments

In facing the challenges resulting from insufficient models in connection with the ISO 26262 different options are at hand.

First one might simply try to ignore the insufficiencies. But this seems to be the worst alternative. Although insufficiencies might be minimized by careful system specification, they are unavoidable for certain kinds of systems, for instance for many surround sensor based DAS. And, as pointed out above, these insufficiencies might cause risks which must not be disregarded a priori without further considerations.

Since the ISO 26262 is not providing any measures to treat such insufficiencies, a second option could be to basically do whatever one wants. That is, everybody facing such a problem develops an individual solution. This might be the only possible way in case of entirely new technologies but is dangerous in the long term. In particular it does not help in a case of liability and does not allow a comparing judgment about created products. Furthermore for

the cooperation of suppliers and OEM a transparent market with defined (quality) standards is useful.

Third, one might try to use a different standard which is more appropriate for the problem considered. For instance the IEC 61496-4-2 concerning safety of machinery using vision based protective devices defines an approach for safety related recognition of persons in a production environment. But the problem is that these approaches cannot be applied to automotive directly. At least the authors are currently not aware of any standard to be applied here with satisfactory result.

Finally another option, which is basically a combination of the last two, is justified by the fact that the application domain is still the automotive domain and therefore, the relevant standard should be adhered as far as possible. The subset of the specification that is defined clearly is in the scope of the ISO 26262. Concerning insufficiencies the work hypothesis is: If it is possible to demonstrate the appropriateness and effectiveness of safety measures and if based on these the residual risk is proven to be sufficiently small, it is acceptable in principle to deal with insufficient models. The methods to achieve this have to be geared to approaches from different standards and consistent with the treatment of hardware failures in the ISO 26262. In the following we will outline such an approach.

6.3 A possible approach based on the ISO 26262

As pointed out above, a documented evidence of conformity of a safety goal must not be based only on the verification of the implementation against the technical system model. It also has to include the validation of the deductive model or of the implemented system against the functional system model. Both can only be successful in a limited scope because any insufficient model only allows a validation with tests, which is limited by definition. Thus the acceptance of a residual risk is necessary and forms the basis of the further considerations. Note in this connection that, although in the ISO 26262 the concept of residual risk in a quantitative manner only occurs in connection of hardware failure measures, in the underlying IEC 61508 it is used in a much wider context, for instance in connection with the ALARP principle, cf. 61508-5, Annex A, B and Ch. 2.5 in [6].

An extension of the standard approach according to the ISO 26262 (outlined in section 2) including insufficient models might be based on a provision for these insufficiencies in a quantitative manner. That is, factors with negative impact in view of the safety goals have to be identified and according to their effect localized and classified. Constraints, i.e. possible sources of failures in the deduction models, are to be described by quantitative models. These models are to be validated by tests with reasonable significance level to ensure that violations of the safety goal are sufficiently improbable. This implies in particular that within these tests failures have to be accepted to a certain degree.

A framework for validating systems based on insufficient models might be formed by third party assessment. In doing so, evaluating an algorithm in general is neither possible nor reasonable. But a process for testing sensors and algorithms can be evaluated. Such a standard for testing needs to impose requirements or might even put restrictions on the technology. The testing also has to include consideration of the insufficiencies of the deduction model and algorithmic characteristics. Here it is necessary to have quantitative measures describ-

ing the limitations of the tests. That is, the system responsible is demanded to provide evidence that the probability for situations, where freedom from faults might not be ensured, is sufficiently small. In determining what 'sufficiently small' means, in the first instance the system responsible should be demanded for defining a quantitative measure describing an accepted rate of violations of the safety goal on total system level. But certainly it makes sense to aim for a uniform perception supported by a consortium of suppliers and OEMs in the same manner as hardware failure rates stated in the ISO 26262. For both, the approach described in [7] might for instance serve as a basis.

If sensor techniques would be standardized, the testing could be done by means of a standardized test suite. (Such a standard test suite must be sensor technology specific since the insufficiencies essentially depend on this.) In that case a simplified approval would be possible. But the crucial point is that the techniques for instance for DAS are quite new, i.e. far from standardization or essential aspects are intellectual property. Accounting for this and ensuring further development it is necessary to initially restrict on measures for the systems engineering instead of standardized test suites. Recall in this connection that the ISO 26262 demands for safety by design. The conformance of the safety goal is ensured by requirements on the development process. Here a generalization in our context is to replace safety by design, which basically means eliminating the insufficiencies, by ensuring that the residual risk caused by insufficiencies is sufficiently small. But the actual definition of the tests is a sole liability of the system responsible. This includes in particular assuring the cogency of the tests. To do so, it is necessary to analyze the deduction model and identify and assess the insufficiencies. A process for testing sensors, algorithms and the deductive models, respectively, matching the system engineering requirements needs to be defined and the requirements for the analysis of insufficiencies and their documentation have to be derived. Finally, the liability of the system responsible also includes the qualification of the test methods and the implementation of a sufficient field observation that required by the ISO 26262.

Concerning the specific implementation of the validation process a validation suite providing confidence in the created model is needed. In this context, for each technology a set of standard tests and test environments might be defined similar to IEC 61496-4-2, which are to be expanded in each particular case. The test cases in the validation suite shall be significant i.e. cover all potentially dangerous inputs (dependent on sensor principle) and the number of tests must satisfy statistical matters including the level of confidence. Based on a statistical evaluation of the tests an estimation of violations of the safety goal must be made using analytical and statistical methods. Of course, here we do aim to a rigorous proof which is obviously impossible. All considerations and conclusions are carried out in all conscience at the time of SOP.

To clarify the issue, consider a video-based DAS. Assume object recognition relevant to safety in the sense that, if some further conditions are complied, a false positive object leads to a violation of the safety goal. Now, a validation of the system according to the approach stated above might consist in the following steps. The starting point is the identification of factors with negative effects concerning the object recognition with the video sensor at hand. The significance of the test suite reflects in completeness of the factors and the number of tests, i.e. in this connection the time duration in their frequency of occurrence. The test suite consists in a careful combination of different factors, i.e. a design of experiments. Combina-

tions of factors with sufficiently small frequency of occurrence can be neglected. Evaluations of the video sensor with this test suite leads to a false positive rate concerning object recognition. Taking into account the further conditions necessary for a violation of the safety goal, other parts of the signal processing are also to be described by statistical models for instance using similar approaches or simulations. Based on this, one might estimate the rate of violations on total system level including a level of confidence.

Finally, the estimated number/rate of violations of the safety goal is to be compared with an accepted rate of violations derived from the quantitative measure on total system level named above. Violations of the estimated rates are a violation of design goals. After release, the violations in the field must be monitored and affirmed according to the ISO 26262.

6.4 Example: System and deductive model, insufficiencies and validation

To clarify the concepts presented so far consider the example of an emergency braking system. The system is to trigger an emergency brake if and only if a collision with another vehicle is unavoidable (or the situation is sufficiently hazardous). This system bears the risk of causing collisions with following traffic. The corresponding top-level safety requirement is ’avoid triggering the brake if no unavoidable collision impends‘. In fact different safety requirements are possible but for the sake of simplicity we restrict on this.

The stated safety requirement is still an informal statement whereof the technical system specification, i.e. the (technical) system model, has to be derived from. To do so, first decisions concerning the technical realization, for instance the basic sensor technology, are to be made. In this example we assume to base our system on a radar sensor only.

Now, on a functional level a situation is represented by the locations and velocities of vehicles in the surrounding. In view of the chosen sensor technology vehicles (including velocities) are represented by radar locations and an actual situation is defined by locations, velocities and perhaps information about roadside like guard rails etc. This corresponds to the estimation- and interpretation-part of the signal processing (here we skip the details to separate these two). The decision whether a collision is unavoidable or not is based on a criticality measure valuating the existence of evasive drive paths. This depends on assumptions on the possible motions of the vehicles and corresponds to the prediction-part of the signal processing. The entirety of these assumptions forms the deduction (or environment) model and the obtained mapping from ’real world‘ to triggering the brake is the (technical) system model.

Obviously this deduction model is insufficient. This already becomes evident by assuming vehicles to be represented by certain radar reflexes, which in fact is an example of an insufficient model. Here completeness of the model would mean to have a one-to-one correspondence between real-world scenarios with vehicles and radar reflexes, which due to the infinite multiplicity is impossible to capture. Thus we are led to define a relation based only on a specific sample, i.e. the deduction model is underspecified. Furthermore we have to accept to have an oversimplification since we might not ensure to distinguish between any two objects based on the radar signals. The prediction is also insufficient: Only based on physical constraints, the decision that the collision is unavoidable would be only possible several 100 ms in advance. To get a reasonable benefit we have to put assumptions for instance on

reaction times, conditions of the tires, behavior of other vehicles, perhaps traffic lanes etc. In any case these insufficiencies might cause mistakes, in the sense of false positives as well as false negatives. In our case the latter might violate the safety goal. This is where the line of arguments breaks such that a verification of an actual implementation is not sufficient but a validation is necessary.

In the example considered, for instance, coke cans or gully covers might be taken for cars (which might rely on the fact that radar sensors considered does not capture the object height) and lead to triggering the brake. Note that, although the specific example might suggest the problem to be solved by using a different sensor, it remains in general. You see this the latest by going to extremes by considering a dummy cars used in development: The are usually constructed such that they 'look like' a usual vehicle for the sensor, but there is no damage in case of a collision. In any case, assuming the optimization of the deduction model already to be exhausted, we have two possibilities to face this problem. First, we might exclude these cases explicitly, for instance by restricting the function on moving objects. This, in fact, is a modification of the functional system model (cf. ROC curve) and not in the scope of this paper. Second, if statistical considerations show that the probability of situations where the safety goal is violated is sufficiently small, the gap might be tolerated. To do so we could estimate the probability of a situation having a can within a correct angle, critical distance and relative velocity etc. Or we could simply evaluate the correct vehicle recognition of a proper system implementation on a sufficiently large data set and calculate the failure rate. The first corresponds to a validation the deduction model, the second to a validation of (an actual implementation of) the system model. In the first case the verification of the correct implementation is still necessary.

For an approval of the system the total residual risk caused by all insufficiencies needs to be sufficiently small in an appropriate sense. This might be achieved by doing this validation step for all main sources of failure combined with a rigorous argumentation that the remaining gap (for instance caused by second order faults) is again sufficiently small.

7 Conclusion

Functional insufficiency is a general challenge and may affect the quality of all system developments that rely on models as measures to counteract complexity. We presented an overview of the use of models in a systems development and described some basic model properties in order to introduce the reader to the challenge of functional insufficiency.

We argued that functional insufficiency potentially affects functional safety: e.g. driver assistance systems with their need to estimate, interpret and predict driving situations are generally complex and need models to capture this functionality. Any of these models may be subject to functional insufficiencies as it is an inherent property of models to apply simplifications. The topic of insufficiencies is not new to engineering as the airbag example demonstrates but in the face of even more sophisticated functions it becomes more and more important. Depending on the models used to verbalize safety goals or safety requirements already those may suffer from the described insufficiency. As a result, a system development may be completely correct with respect to the requirements but may fail to satisfy the safety goals, though. In addition, by incomplete knowledge about the reality, approximations are

created during the development that needs to be proved sufficiently in order to provide an acceptable level of safety.

These considerations lead to the discussion of the accepted level of safety and residual failures of the system that are not only caused by hardware faults but caused by inevitable insufficiencies. Only by accepting that an implementation of a system may fail to reach a safety goal because of insufficiencies allows to discuss further measures that are required in order to achieve functional safety. We proposed a basic set of requirements for such an extended setting in order to initiate and drive a discussion of this topic in a broad community.

8 Literature

[1] ISO 26262 (all parts). *International Standard Road vehicles — Functional safety*, 2011

[2] IEC 61508 (all parts). *Functional Safety of Electrical / Electronic / Programmable Electronic Safety-Related Systems (E/E/PES)*. 2nd edition, 2010

[3] ISO/FDIS 26262-10. *Final Draft International Standard Road vehicles — Functional safety. Part 10 'Guideline on ISO 26262'*, 2012

[4] Federal Court (Bundesgerichtshof). *Case as of June 16th 2009 – VI ZR 107/08*, http://dejure.org/dienste/vernetzung/rechtsprechung?Text=VI%20ZR%20107/08

[5] C.L. Heitmeyer, J. McLean. *Abstract Requirements Specification: A New Approach and Its Application*. IEEE Transactions on Software Engineering (SE-9 Issue:5), 1983

[6] P. Löw, R. Pabst, E. Petry. *Funktionale Sicherheit in der Praxis. Anwendung von DIN EN 61508 und ISO/DIS 26262 bei der Entwicklung von Serienprodukten*. dpunkt Verlag, 2010

[7] S. Ebel, U. Wilhelm, A. Grimm, U. Sailer. *Wie sicher ist sicher genug? Anforderungen an die funktionale Unzulänglichkeit von Fahrerassistenzsystemen in Anlehnung an das gesellschaftlich akzeptierte Risiko*. 6. Workshop Fahrerassistenzsysteme, 2009