On the validation of complex systems operating in open contexts

Alexander Poddey*  Tino Brade†  Jan E. Stellet†  Wolfgang Branz†

2019

Abstract

In the recent years, there has been a rush towards highly autonomous systems operating in public environments, such as automated driving of road vehicles, passenger shuttle systems and mobile robots. These systems, operating in unstructured, public real-world environments (the operational design domain can be characterized as open context) per se bear a serious safety risk. The serious safety risk, the complexity of the necessary technical systems, the openness of the operational design domain and the regulatory situation pose a fundamental challenge to the automotive industry. Many different approaches to the validation of autonomous driving functions have been proposed over the course of the last years. However, although partly announced as ‘the solution’ to the validation challenge, many of the praised approaches leave open crucial parts. To illustrate the contributions as well as the limitations of the individual approaches and providing strategies for ‘viable’ validation and approval of such systems, the first part of the paper gives an analysis of the fundamental challenges related to the valid design and operation of complex autonomous systems operating in open contexts. In the second part, we formalize the problem statement and provide algorithms for an iterative development and validation. In the last part we give a high level overview of a practical, holistic development process which we refer to as systematic, system view based approach to validation (in short sys2val) and comment on the contributions from ISO26262 and current state of ISO/PAS 21448 (SOTIF).

1 Introduction

In the recent years, there has been a rush towards highly autonomous systems operating in public environments, such as automated driving of road vehicles, passenger shuttle systems and mobile robots. These systems, operating in unstructured, public real-world environments (the operational design domain can be characterized as open context) per se bear a serious safety risk. The serious safety risk, the complexity of the necessary technical systems, the openness of the operational design domain and the regulatory situation pose a large challenge to the automotive industry.

Due diligence is necessary in development, release and even post release operation, which are all related to validation aspects. Successful, ongoing demonstration of safe operation and strict avoidance of fatal incidents in all day use is necessary for societal acceptance. Otherwise, a ‘winter of autonomous systems’ (II) might come down and the large investments taken e.g. in the automotive industry will not pay-off.

*Bosch Center for Artificial Intelligence, Robert Bosch GmbH, Corporate Research Campus, 71272 Renningen, Germany, alexander.poddey@de.bosch.com
†Vehicle Safety and Assistance Systems, Robert Bosch GmbH, Corporate Research Campus, 71272 Renningen, Germany
Many different approaches to the validation of autonomous driving functions have been proposed over the course of the last years. There have been discussions, among others, about real world driving based statistic approaches (following the state of the art in assisted driving), simulation, formal patterns, partly as silver bullet (one for all solutions), partly in combination. However, although partly announced as ‘the solution’ to the validation challenge, many of the praised approaches leave open crucial parts. In order to be able to illustrate the contributions as well as the limitations of the individual approaches, we give an analysis of the fundamental challenges related to the valid design and operation of complex autonomous systems operating in an open context in the following. Strategies for ‘viable’ validation and approval of such systems can then be discussed, based on this.

In section 2 we introduce basic terms and concepts as well as a detailed presentation of the validation- and approval related necessities for complex systems operating in an open context, summed up as validation challenge. Boldface characters are used for term definitions, whereas italic characters indicate the use of already defined terms within the document. Section 3 formalizes the problem statement and provides algorithms for iterative development and validation. In section 4 we give a high level overview of a practical, holistic development process, which we refer to as systematic, system view based approach to validation (in short sys²val) and comment on the contributions from ISO26262 and current state of ISO/PAS 21448 (SOTIF) for automotive focused applications.

2 Problem statement

2.1 Conception of validity of complex systems in open contexts

Autonomous systems operating in public environments are usually designed to take over typical human tasks like e.g. driving road vehicles. The technical systems, or strictly speaking its developers and distributors, thereby do not only need to fulfill the tasks from a functional point of view. They also need to take over responsibility for safe operation and mitigation of hazardous situations, traditionally incurred by the human operator. This is significantly different from e.g. assisted driving where a quite limited assistance function is continuously supervised by the driver. Over and above, the unstructured real-world operational design domain can be characterized as an open context. It bears infinitely many characteristics, possible interactions and effects, which cannot be expressed formally complete (we refer to this as $\infty$-complexity). Moreover, the context develops in time (it is evolving) with so far unseen characteristics and interactions appearing suddenly, e.g. post release. Due to the complexity of the autonomous system (e.g. compared to an assistance function, see also emergent behavior discussed later on) and the resulting numerous possible interactions, changes in context might lead to undesirable behavior of the system. Both, the complexity of the system and the context therefore require several topics to be addressed in a systematic and holistic approach in order to achieve a valid product. With valid system, we refer to a product which bears no unreasonable risk\footnote{According to ISO26262: Risk judged to be unacceptable in a certain context according to valid societal moral concepts.} to users and the society, and, however subordinated, no unreasonable risk to the manufacturer, which is related to liability issues and costs e.g. due to unreasonable development expenditures. The following list provides an overview of these topics:

- functional safety (e.g. IEC61508, ISO26262)
• safety of the intended functionality (SOTIF)
• safety of machine learning
• safety of use
• security
• product liability
• meeting customers and society’s expectations

As will be discussed in the following, complex systems operating in an open context will never be perfectly valid (i.e. a system which bears absolutely no risk).

In accordance with system-view based approaches (e.g. STAMP \[2\]), all the aspects illustrated in the above listing are part of the high level goal. The negative effect of non-achievement of aspects of the high level goal is referred to as loss. Losses large enough to undermine the achievement of the overall goal are referred to as unacceptable loss. A system possibly leading to unacceptable losses is invalid, as it bears unreasonable risk.

The differentiation between loss and unacceptable loss is related to the (societal) tolerated risk. For the moment, it is sufficient to note that the (societal) tolerated risk is a complex topic and even a moving target, which, among others, depends on attributes like type of cause of loss (e.g. force major cause, random effect - like e.g. random hardware errors - or systematic malfunction). Societal factors play an important role for what might be regarded as tolerated. Societal expectations, influenced by advertising, current jurisprudence and trends however are hard to quantify, which makes arguing about tolerated risk a fuzzy topic. In addition, the validation of such systems can no longer be solved in a ‘once for all time’ approach, as it will be discussed later on in the context of ongoing validation.

At first we will introduce some additional terms and concepts related to the root challenge of validation.

2.2 Basic concepts from development approaches

Over the last decades, the challenges in developing complex systems have been approached in many different ways (we comment on the main approaches in 4.2). In the following, we introduce some basic concepts and terms necessary for the problem statement.

In general, the development of complex systems deals with different layers of abstraction. Established approaches usually apply a quite coarse definition of horizontal layers, such as e.g. system-, functional- and implementation layer, as illustrated in figure 1 as boxes. These horizontal layers each encapsulate several (fine-grained) layers of abstraction (e.g. $i \downarrow i + n$ in the first box of figure 1). Moving down the layers of abstraction corresponds to the vertical direction. The basic idea behind the usage of coarsely defined horizontal layers is to encapsulate the specific necessities of the different layers with well defined interfaces and transitions among them (large arrow in the figure, e.g. handing over a set of requirements from the system- to the functional layer).

Reducing complexity and supporting reuse within a given layer is approached by encapsulation into elements (e.g. system components within the system layer) that interact via well defined interfaces. This is referred to as encapsulation along the horizontal direction. Typically, during
development, there are several transformations taken also along the horizontal direction (indicated as \( i + 1 \rightarrow i + 2 \) in the first box of figure [1]).

Figure 1: Illustration of the concepts of coarse horizontal layers (gray boxes) encapsulating fine-grained layers of abstraction (e.g. \( i \downarrow i + n \) in the vertical direction, and \( i + 1 \rightarrow i + 2 \) in the horizontal direction in the first box).

2.3 Validation triangle

Figure 2 provides an illustration of the three basic aspects related to the overall validation problem (we refer to this as validation triangle).

Figure 2: The validation triangle consists of the three, interdependent aspects purpose (P), context (C) and realization (R), subsumed as the triad T.

In general, providing a valid solution to a given task is related to (depending on the horizontal layer) developing, designing or implementing a realization R (e.g. a technical system) for a purpose P (e.g. autonomous driving on a highway), which needs to be fulfilled in a specific context C (e.g. on German highways at daytime). This three-way dependency is present across all levels of abstraction, e.g. on the highest level as in the above example, down to the implementation of a concrete sub-function or technical component. The technical component as well is built to fulfill an intended purpose (contributing to the main purpose of the technical system) in a given context. We use the shorthand notation T to refer to the interdependent triad of P, R, C.

In the following, we will step-wise expand figure 2 to build up an illustration of the manifold and interrelated aspects of the actual validation problem related to complex systems operating in open contexts.
2.4 Implicit infinite complexity

The fundamental challenge of validation of complex systems operating in an open context is that none of the three aspects of the validation triangle - purpose (P), context (C), and realization (R) - can be expressed in a formally complete manner and, as already discussed above, the expression of one aspect even depends on the remaining two. This is illustrated in figure 3a) as a fuzzy black frame. In the following, the cause of this will be discussed per aspect.

On the one hand, as already discussed above, unstructured real-world operational design domains (open contexts) bear infinitely many possible interactions and effects, they are \(\infty\)-complex. The purpose, on the other hand, is based on implicit expectations, which also cannot be expressed formally complete. Over and above, statements regarding a certain purpose depend on the related context of application and possibly even on the characteristics of the chosen realization and therefore most often are based on implicit assumptions about these aspects of the validation triangle.

Regarding the purpose, we differentiate between the aimed purpose (implicitly expected), which is necessarily vague, and the intended purpose, relating to explicitly expressed expectations (e.g. a specification). We apply here the meaning of ‘intended’ as ‘explicitly expressed’ as applied in the context of ISO26262 & SOTIF in connection with intended functionality (I)\(^2\) (behavior specified for an item, system, or element excluding safety mechanisms) and intended behavior (I)\(^3\) (specified behavior of the intended functionality including interaction with items). The difference between intended (explicitly expressed) and aimed (implicitly expected) can be illustrated as the difference between what has been stated and what was actually meant. One of the challenges related to complex systems in open contexts is that the developer, the customer or even the society might be able to express an expectation e.g. about appropriate behavior of the system for a given, specific setting (related to a specific realization in a specific context). However, due to the \(\infty\)-complexity, it is non-trivial or rather impossible to identify all relevant settings. The explicit expectations therefore are usually expressed quite abstract leaving open room for (setting specific) interpretation. The same explicit but abstract statement might even lead to quite different interpretations, depending on different concretizations of the setting.

Vice versa, the necessary mapping of the \(\infty\)-complex context onto a reduced subset of ‘expected to be relevant’ context (valid projection onto a finite representation) strongly depends on the related specific purpose and realization. For example, the presence and characteristics of metallic structures might be relevant for radar-based realizations, whereas thermic radiation might be irrelevant, as long as no corresponding sensing system is applied, or possible irritation of the realization due to thermic radiation might happen. This illustrates that the representativeness (i.e. a subset of something accurately reflects the larger super-set), which is a basic precondition for statistic argumentation, cannot easily be achieved and is impossible to achieve for a complex system in evolving open contexts without a valid analysis of all aspects of the validation triangle (this is referred to as representativeness challenge).

Last but not least, the same field of tension is present between the expected and true/effective characteristics and behavior of a realization. The properties of complex systems - independent of the effect of the operation in an open context - are emergent. I.e. the effective properties and behavior of a realization is a result of the complex interplay of the components and can therefore, again, not easily be expressed formally complete. We refer to this as emergent characteristics.

\(^2\)Terms defined by ISO26262 / SOTIF are annotated by (I) in order to explicitly indicate this circumstance.

\(^3\)Don’t be irritated by the apparent circular dependency between the two definitions via the term behavior. Specified behavior in the definition of the intended functionality is meant as ‘what was specified regarding the functionality’.
and emergent behavior. This is a second aspect, on top of the complex interaction between the realization and the context. As a consequence, an explicitly expressed expected behavior of a realization (related to the intended behavior (I) of SOTIF) might be different from the effective behavior as well as different from the effectively necessary behavior – necessary to meet the purpose. We will introduce a special notation for the expected and effective aspects of the triad in section 2.6.

2.5 Subspace of perfectly valid solutions \( \tilde{T} \)

As discussed in the previous section, the space of possible triad combinations \( T \) is infinitely large and can not explicitly be expressed. However, in the sense of a thought experiment, suppose an omniscient observer could classify any possible triad either as perfectly valid or not. By this, he could thereby explicitly determine the decision boundary, which reduces both, the relevant hyper volume in \( T \)-space to a finite volume and the implicitly regarding the decision boundary to zero. The subspace of perfectly valid triads will be denoted by the tilde notation \( \tilde{T} \).

Having an explicit description of the decision boundary would reduce the validation problem effectively to a verification task (i.e. checking a given triad \( T \) against a fully explicit decision boundary for \( \tilde{T} \)). However, the description of the decision boundary will necessarily be highly complex, as illustrated in figure 3 by the ragged curve. In reality, the decision boundary could neither be determined nor would the explicit complexity be manageable. In anticipation of section 2.9 due to this, the development goal is targeted at a valid system (i.e. a system bearing no unreasonable loss, denoted by \( \tilde{T} + \text{loss} \)), not a perfectly valid system (\( \tilde{T} \)).
Figure 3: The implicit $\infty$-complex space of possible triad combinations (fuzzy black frame in a), the subspace of perfectly valid solutions $\tilde{T}$ and the effect of alternative deductions, aiming at approximating $\tilde{T}$ (illustrated in b-e). For further details see section 2.8.
2.6 Model of reality $\tilde{T}$

From section 2.4, it becomes clear that developing complex systems for open contexts necessarily deals with more or less simplified representations of the $\infty$-complex reality (including mutual interaction) for all three aspects (purpose, context and realization) across all levels of abstraction. According to the most general definition, we refer to this representation of reality as model. Just as argued above, a model can never be generally complete & correct (i.e. valid) in every aspect. It can only be sufficiently complete & correct for a certain purpose in a given context. Specifically, we are interested in models of valid systems (i.e. models of valid triads).

Our work is based on [3], who analysed the role of models in the context of ISO26262. We generalize this to the triangle of validation and hence to the more general task of validation of complex systems operating in an open context.

Building a model is a form of deduction (sometimes also referred to as concretizations), which might happen implicitly e.g. in the form of a mental model unconsciously applied by the developer or explicitly e.g. in the form of a formal model, a written requirement or even written code. It is important to note that all statements, reasoning and arguing about any aspect of the validation triangle is, as emphasized by the previous sections, necessarily related to a (more or less reduced) model of the reality.

We denote this model of reality by the 'bar' notation, e.g. $\tilde{T}$ for the models related to the aspects of the validation triangle, specifically $\tilde{P}$, $\tilde{R}$ and $\tilde{C}$. Due to the $\infty$-complexity, the true characteristics, in the following denoted by the dagger notation, e.g. $T^\dagger$, can not be explicitly expressed at any level of abstraction. In addition we include the interdependence of the aspects into the notation, so that:

- For a given realization $R$, we are able to think, argue, reason about the related properties and behavior only in representations of reality, i.e. models $\bar{R}(\bar{C}, \bar{P})$, which are not necessarily consistent with the true characteristics and behavior $R^\dagger(C^\dagger, P^\dagger)$.

- Regarding the context $C$, the expected to be relevant part $\bar{C}(\bar{R}, \bar{P})$ might not necessarily be consistent with the effectively relevant one $C^\dagger(R^\dagger, P^\dagger)$.

- The mapping of the aimed purpose $P$ in form of $\bar{P}(\bar{C}, \bar{R})$ might not necessarily meet the effectively achieved purpose $P^\dagger(C^\dagger, R^\dagger)$.

In figure 3, we indicate the consecutively derived models of reality $\tilde{T}$, corresponding to the development process, as dashed lines. The related effective triangle $T^\dagger$ as dash-dot-dagger line.

2.7 Deductive gap, validation and evidence

Building a model is a form of deduction. Different types of deductions appear all along the process from the initial formulation of a product idea to the implementation of a concrete product. An example from the top-most level, as discussed above, would be textually writing down the intended purpose, the relevant aspects of the context and the characteristics of the realization (e.g. as set of requirements and constraints). Other examples would e.g. be the semantic transformation of

\[^4\text{Note that we apply the term deduction of model for both, deductive and inductive model building. The effect of inductive model building (i.e. deriving a model based on specific observations of reality), namely a gap between the resulting model and the reality, as well as the related necessary validation is comparable to the deductive gap.} \]
textual representations (e.g. requirement transformation like decomposition and refinement), transformation of textual representations to more formal / testable representations, system- or functional decomposition, model reduction and transition from specification to concrete implementation.

The unavoidable deductions however are at the heart of the validation problem. Following from the above discussion, it is clear that every model building is necessarily based on explicit and most often even implicit assumptions. In other words, the deduced representation corresponds to the initial representation only under certain assumptions. If these assumptions are not justified (even temporarily), the deduced representation is invalid, and the derived realization provides insufficient characteristics compared to both, the intended and aimed purpose of the initial representation for the relevant context. In other words, there possibly exists a gap between the initial- and the effect of the deduced representation. We therefore refer to this as deductive gap and insufficiency of the deducted representation.

The complex multi-dependency of the related aspects, as well as the impossibility to formally complete express them is characteristic for the validation problem. In contrast to that, verification is related to checking for differences between, at least in principle, formally complete expressible representations. The adverse effect of the deductive gap is quite problematic. Due to the necessary deductions, the system might operate as specified (the specification itself is the result of several deductions) and the implemented system might be well verified against this specification, but, nevertheless, it might not succeed to fulfill neither the intended nor aimed purpose of the initial representation, showing malfunctioning behavior leading to unsafe operation. In contrast to e.g. random hardware errors, the effect of the deductive gap is systematic, possibly leading to a systematically harmful system, which is intolerable. SOTIF partly addresses this, as discussed in section 4.3.

In addition, the for complex systems unavoidable difference between the expected (\(\overline{T}\)) and the effective (\(T^{\dagger}\)), related to the deductive gap, might have a strongly adverse effect on the validation activity (i.e. hypothesis checking). This is because usually, the same projections of reality underlying \(T\) are typically applied during the derivation of validation tasks and analysis of results. This leads to the fact that the results of a hypothesis check, e.g. a test, apparently support the hypothesis, although it is, in fact, wrong. This has a much more adverse effect than insufficient or missing evidence (related to an unfounded argument, see also [4]). We therefore refer to this effect as misleading argument.

For some of the many different types of gaps resulting from different deduction types mentioned above, specific names are used. (see e.g. [3]). We will introduce here only two frequently used terms, namely semantic gap, which relates to the gap following from the deduction of linguistic representations as well as specification gap relating to the gap between the effect (\(T^{\dagger}\)) of a concrete specification (\(\overline{T}\)) (subsuming several steps of deduction, including the transition from the \(\infty\)-complexity to a finite representation) and the implicitly aimed (i.e. a valid solution, \(\overline{T}\)).

Over the course of the last years, there have also been proposed different terms for the adverse effect of the deductive gap (i.e. the insufficiency), e.g. functional insufficiency (\(I_{F}\)) (working title in the SOTIF context), functional deficiency and performance limitation (\(I_{F}\)). The proposed terms are partly misleading as they might be understood as related only to a sub-part of the actual problem, e.g. functional insufficiency might be interpreted as related only to the effect

---

5Term used in the context of Safety Of The Intended Functionality (SOTIF), in particular in the definition of SOTIF as 'absence of unreasonable risk due to hazards resulting from functional insufficiencies of the intended functionality or from reasonably foreseeable misuse by persons'. However functional insufficiency as term is currently not defined in the terms and definition of the recent state of the ISO/PAS 21448:2018

6SOTIF ISO/PAS 21448 :2018: insufficiencies in the implementation of the intended functionality
of the deductive gap on the functional level (neglecting the remaining horizontal layers illustrated in section \[2.2\]).

The **validation challenge** is closely related to the many necessary deductions applied during development: for every step, possibly leading to deductive gaps,

- all underlying assumptions need to be made explicit
- per assumption, evidence needs to be provided for its validity

**Evidence** is generated from meeting of expectations. Hypotheses need to be formulated and their legitimacy needs to be demonstrated, e.g. by formal approaches, simulation and real world observation. Due to the complex open world problem, hypothesis checking based on the developed model of reality $\tilde{T}$ needs to be done carefully, as applying the same approximations in derivation and interpretation of hypothesis checks might lead to a self-fulfilling prophecy and hence to a wrongly passed test, providing a misleading argument for the validity of the deduction (see the following section for a detailed discussion).

### 2.8 Development, iterative deduction and redesign

Figure 3b illustrates a typical situation during development after $i$ steps of deduction, achieved by any approach (layered design, V-model guided etc.). The deduction towards $\tilde{T}_i$ (bold arrow 1) reduces the $\infty-$complex and $\infty-$voluminous space of possibilities to a more or less abstract (implying more or less implicit), however finite representation indicated by the dashed line. The related explicit complexity is much lower than that of the (actually practically not determinable) perfectly valid $\tilde{T}$ - indicated by the simple geometry of $\tilde{T}$. In a nutshell, the development process can be understood as iterative approaching a good enough (i.e. valid) approximation of a perfectly valid $\tilde{T}$, without knowing $\tilde{T}$ explicitly (see also the conception of validity in section 2.1 and the discussion related to $\text{cond1}$ in section 3.2).

As discussed in section 2.6, the deduced triad $\tilde{T}$ might not necessarily fully correctly map the true characteristics $T^\dagger$. This deviation is depicted as deviation of the dash-dot-dagger line from the dashed line for parts of the model $\tilde{T}$ in the lower part of figures 3b-f. The validity of the developed system however is only related to the deviation between the effective ($T^\dagger$) and the necessary ($\tilde{T}$). Due to this deviation between $T^\dagger$ and $\tilde{T}$, four different source types of loss might be inherent to the resulting system, namely *loss* (small bolts) or *unacceptable loss* (striped large bolts), each related to either unnecessarily included subspace (e.g. unacceptable loss included in upper left of figure 3b) or neglected relevant subspace (right hand side of figure 3b), relative to $\tilde{T}$.

An important aspect of any reasonable development approach is encapsulation, which means that a further deduction from $\tilde{T}_i$ is based on $\tilde{T}_i$ solely. This however implies that any further deduction $\tilde{T}_i \downarrow \tilde{T}_{i+1}$, as illustrated in figure 3b (bold arrow 2), can not reintroduce an already neglected subspace. Therefore, an already inherent risk for the manifestation of loss due to neglected relevant subspace can not be cured by further deduction from $\tilde{T}_i$. Only a redesign on (at least) level $i$, related to a deduction $\tilde{T}_{i-1} \downarrow \tilde{T}_j$ (bold arrow 3), might solve the missing relevant subspace issue as show in figure 3d. The necessary level for redesign corresponds to the level at which the relevant subspace got lost during deduction. After redesigning, deduction can be continued, as illustrated by $\tilde{T}_j \downarrow \tilde{T}_{j+1}$ (bold arrow 4). Finally, a valid result might be achieved (i.e. a good enough approximation of $\tilde{T}$ such, that no unacceptable loss might occur from the operation of the system. This is related to the development goal, discussed in the following section.
In conclusion, this makes clear that validation aspects are present on every layer of abstraction, more specifically, validation is strongly related to every step of deduction. These deductions occur frequently, also within an encapsulated coarse horizontal layer (such as e.g. system, functional and implementation layer). Per deduction, a sound validation argumentation is necessary, to prevent major redesign expenditures later on. Aspects missed in early phases of development are particularly problematic. Validation therefore needs to be addressed seriously, right from the beginning of development.

2.9 The development goal

From the previous sections, it becomes clear that complexity and interdependence are the root causes for the manifestation of loss. With respect to complexity, one can differentiate two aspects, namely that related to the hyper-volume of the solution space and, on the other hand, the characteristics of the decision boundary enclosing the hyper-volume. Starting from a high level goal, the complexity is mainly implicit and related to the possibly infinite volume of the principal solution space. The development process maps this infinite complexity starting point onto a finite representation, which has initially more abstract (implicit) shares and is continuously driven towards a more explicit approximation of the perfectly valid solution. Thereby, the considered solution space volume all in all is reduced. However, the explicit complexity (related to the decision boundary enclosing this subspace) increases. Even though explicit, this complexity however also contributes to the possible deviation between the mental/explicit model the developer has (the expected characteristics) and the true effective characteristics of a built system (i.e. $T_{\text{eff}} - T = \Delta_{\text{model-real}}(\text{complexity})$).

The development goal therefore is not a fully explicit expression of the perfectly valid solution ($\tilde{T}$), but rather an optimum between the implicit and explicit complexity such, that the possible manifestation of loss is reduced on a robust basis. More specifically, the system shall be free from unacceptable loss. With robust basis, we refer to the fact that, for a complex system in an open context, a formal proof of the validity of the underlying assumptions is impossible. Robustness against deviation from what is expected (based on $\bar{T}$) and effective operation of the system ($T_{\text{eff}}$) needs to be designed into the system. In addition, validity of assumptions and related validation argumentation needs to be continuously checked post release. This is referred to as ongoing validation.

In conclusion, the iterative deduction applied during development - e.g. working through the horizontal layers of abstraction from the high level goal to the final build product - serves four basic purposes:

1. deducting valid systems at the given layer of abstraction, preventing major design iterations later

2. iterative reduction of implicit complexity

3. keeping the related explicit complexity manageable

4. providing metrics for hypothesis checking and evidence generation on every layer for pre-release validation and post-release checking and maintaining of validity. We refer to the derived metrics at the heart of ongoing validation as assumption monitors.
3 Problem formalization

3.1 Full partition of solution space

The relation among the development goal (i.e. a valid solution, see section 2.9), the expected ($\bar{T}$) and the effective ($T^\dagger$), see section 2.6, as well as the role of assumption monitoring (discussed in section 2.7, 2.8 and 2.9) is visualized in figure 4.

We apply the following notation: labeled edges refer to the complete related circle area; we have:

- Valid solutions. I.e. perfectly valid $\bar{T}$ plus the set of solutions bearing possible loss which is below unacceptable loss, indicated by the black (diffuse) edge. This is the development goal (see section 2.9). The diffuse style relates to the fuzziness of the discrimination between loss and unacceptable loss (it depends on attributes like type of cause of loss and societal factors as discussed in section 2.1).

- The expected $\bar{T}$ on the lower right of the diagram. It relates to all the mental and explicit models (see section 2.6).

- The effective ($T^\dagger$) on the lower left (see section 2.6).

Labeled areas refer to the related circular ring area, here the area labeled by $\Delta \bar{T}_{mon}$ referring to any deviation from the expected ($\bar{T}$), which is covered by assumption monitors ($\Delta \bar{T}_{mon}$).

As discussed in connection with the deductive gap (section 2.7), iterative deduction and redesign (section 2.8) and ongoing validation (section 2.9), assumptions and related hypothesis checking play an important role for validation. The assumption-related measures (like monitoring) are an inevitable add-on to the models ($\bar{T}$), intentionally extrinsic to $\bar{T}$, in order to overcome the misleading argument problematic. Intentionally extrinsic means that we aim for assumption handling not prone to the same limitations than the model. Making assumptions explicit and monitoring them prevents the misleading argument and allows for incidents triggering specific rework for improvement during development and post-release ongoing validation. The aim is to maximize this area such, that the effective ($T^\dagger$) is fully covered.

Numbers refer to the subareas of circular intersections (the area having one effective color from overlays). Area 1, for example, indicates the subset of solutions, which effectively are as expected and valid (overlay of $T^\dagger$, $\bar{T}$ and $\bar{T}+\text{loss}$). This, in fact, is the area that we aim to maximize.

Table 1 explains all subareas referenced in figure 4.

\footnote{With incident we refer to any event leading to a monitor indicating a deviation. Sound monitoring allows identifying such deviations, even before any loss manifests.}
| Id | Description |
|----|-------------|
| 1  | The space of solutions being effectively as expected and valid. This is the area we aim to maximize. |
| 2  | The space of solutions being effectively as expected, however invalid, e.g. due to invalid assumption. Due to the congruence of the expected ($\bar{T}$) and the effective ($T^\dagger$), the deviation can be found via dedicated verification and validation (v&v) efforts based on $\bar{T}$. |
| 3  | The effective deviates from the expected and is (by chance) valid. Due to the deviation between the effective ($T^\dagger$) and the expected ($\bar{T}$), the deviations are not accessible via dedicated v&v efforts based on $\bar{T}$. However, the deviation is accessible via assumption monitoring. |
| 4  | The effective deviates from the expected and is invalid. Due to the deviation between the effective ($T^\dagger$) and the expected ($\bar{T}$), the deviations are not accessible via dedicated v&v efforts based on $\bar{T}$. However, the deviation is accessible via assumption monitoring. The monitors allow to identify already smaller incidents. The assumption related efforts allow to trigger specific rework for improvement. |
| 5  | The effective deviates from the expected and is invalid. Due to the deviation between the effective ($T^\dagger$) and the expected ($\bar{T}$), the deviations are not accessible via dedicated v&v efforts based on $\bar{T}$. In addition, the deviation is not accessible via assumption monitoring. Invalidity can only be discovered by manifestation of unacceptable loss; this however can not be explained based on the present models. Such an event triggers major rework. |
| 6  | The effective deviates from the expected and is (by chance) valid. Due to the deviation between the effective ($T^\dagger$) and the expected ($\bar{T}$), the deviations are not accessible via dedicated v&v efforts based on $\bar{T}$. No assumption monitoring. This is the region related to the misleading argument problematic as the wrongful expectation might apparently be supported by dedicated v&v efforts based on $\bar{T}$. |
| 7  | Valid design space not used. This might be inconvenient as in principle available design space is not used for optimization of product properties, or even be problematic (from a validation point of vie), as implicitly expected properties of the product are not achieved. |
| 8  | Invalid model, however never effectively reached part of the solution space. |
| 9  | In principle monitored deviation from the expected (however invalid). The effective deviates from the expected and never reaches this part of the solution space $\rightarrow$ never effectively triggered assumption monitor. |
| 10 | The expected would be valid, however the effective deviates and never reaches this part of the solution space. |
| 11 | In principle monitored deviation from the expected (valid), however the effective deviates from the expected and never reaches this part of the solution space $\rightarrow$ never effectively triggered assumption monitor. |

Table 1: Description of the subareas of the intersection between the development goal (i.e. a valid solution, see section 2.9), the expected ($\bar{T}$) and the effective ($T^\dagger$, see section 2.6), as well as the role of assumption monitoring (ongoing validation) referenced in figure 4.
Figure 4: The relation between the development goal (i.e. a valid solution, see section 2.9), the expected ($\bar{T}$) and the effective ($T^\dagger$, see section 2.6), as well as the role of assumption monitoring (ongoing validation) with referenced subareas, explained in table 1.

The area 6 relating to solutions that are prone to the misleading argument problematic is especially critical in the region close to the fuzzy boundary between loss and unacceptable loss. Even when conscientiously following the societal discussion and in principle being prepared to respond to shifts, a (by chance) valid effective solution of type 6 may become unnoticed invalid (at least until an unacceptable loss occurs), as the model $\bar{T}$ spuriously relates the solution to a different solution space which is not affected by the shift.

In conclusion, with respect to the areas illustrated in figure 4, a development process should address the following aspects:

1. Maximize area 1. This is the ultimate goal, which needs to be supported in initial design, during development and post release in the sense of continuous improvement. All the following aspects contribute to this.

2. Provide support for identifying the space of valid solutions (circle $\tilde{T}$+loss), especially addressing the diffuse boundary between loss and unacceptable loss (black diffuse edge).

3. Provide support for derivation of valid models (circle $\bar{T}$), necessarily accompanied by support in identifying and validation of underlying assumptions, and maximizing circular ring $\Delta\bar{T}_{mon}$.

4. Provide support for robust design: large however expected deviations from the nominal should be taken into account in development. I.e. don't design a system mainly operating close to the fuzzy boundary.

5. Derive sound dedicated v&v contributions based on $\bar{T}$ and underlying assumptions (addressing area 2).

6. Be prepared to the unexpected. I.e. maximize $\Delta\bar{T}_{mon}$ (in 3.) and establish the possibility to identify already minor incidents and trigger specific rework for improvement. This is beyond...
robustness and related to what Taleb refers to as anti-fragility - a system under stress needs to survive and get better [5, 6, 7]. Especially maximize area 3 and 4, thereby minimizing 5 and 6.

7. Support design freedom by using the available design space (i.e. minimize 7). In a nutshell, the process should allow addressing the above mentioned aspects and identifying design drivers, while preventing a too early focusing on certain implementations (which would effectively reduce the available solution space). As an example, a system level hazard and risk analysis allows identifying design drivers in an early stage of development.

3.2 Formalized algorithm for iterative development and validation

Initial remark

In this section, formalized algorithms (and the necessary ingredients) for the development and iterative validation of complex systems, operating in open contexts, are provided. Due to the complexity of the validation issue, this formalization is regarded necessary in order to ensure that none of the many subtle details are neglected in development process design.

The reader being mainly interested in the broad mindset might skip the formalized algorithms and head on for section 4, after ensuring he is aware of the fact that cond1, given below, can practically only be enforced indirectly (via eq. (2) and cond2). All other basic aspects have already been discussed in the foregoing sections, and a high level overview of a practical process is given in section 4.

Figure 5 provides a high level overview of the complex interdependence and mutually recursive flow through the algorithms.

Definitions and ingredients

A representation is a set of statements representing the possibly ω-complex reality. A representation might be built from a wide range of elements, e.g. abstract textual statements or even an expansion of reality in an explicit basis sets in the mathematical sense.

Any given representation therefore has a certain abstraction level whereas higher abstraction levels are related to more implicity, leaving room e.g. for deviating interpretation (i.e. alternative concretizations).

Transformation of representation to another representation is indicated by $\bar{T}_i \downarrow \bar{T}_{i+1}$

The actual goal during development is to achieve a sufficiently explicit representation of the relevant part of reality (e.g. the aspects of the validation triangle) in order to be able to argue about the validity of the achieved solution (see section 2.9 for details). More formally, for complex systems operating in an open context, the condition is

$$cond1 : (\bar{T}_{suff}^{\text{robust}} T^\top) \land (T^\top \in \{\bar{T} + \text{loss} \}^{\text{robust}})$$

(1)
I.e. the model of reality $\bar{T}$ needs to sufficiently well approximate the effective $T^\dagger$, this approximation needs to be robust against stress, and the effective needs to be in the subset of valid solutions $\{(\bar{T} + \text{loss})_{\text{robust}}\}$ which robustly stay valid even under stress. In addition, possible deviations need to be recognized for antifragility (i.e. a system under stress needs to get aware of this and means to survive and get better need to be prepared), see assumption monitors in the following.

Neither the effective $T^\dagger$, nor the valid $\bar{T} + \text{loss}$ are directly accessible (as discussed in the foregoing sections). Therefore, achievement of $\text{cond}1$ can not directly be argued / examined. Instead, development and argumentation about the system is based on representations of reality, which consists of models of reality $\bar{T}$ related to the aspects of the validation triangle, accompanied by further elements as introduced in the following. $\text{Cond}1$ is then indirectly enforced via the complex, however accessible condition $\text{cond}2$ given below.

The high level goal is the most abstract representation of the implicit, $\infty$-complex product idea (which initially is free from representation). Following system-view based approaches (e.g. STAMP [2]), a sufficiently complete set of high level unacceptable losses can be derived from the high level goal. Based on this, a set of high level constraints $\{Cr\}$ can be formulated, which constrain the system away from states which might, under certain conditions, lead to the manifestation of the determined unacceptable losses.

The initial formation of representation of the implicit and $\infty$-complex reality related to the product idea, just as every following transformation of representation, usually involves a set of assumptions $\{A\}$. These underlying assumptions need to be elicited by a conscientious process and the sufficiently completeness and validity needs to be argued (see evidence-based argumentation below, which is required also for other elements, such as $\{Cr\}$).

The basis for the indirect approach to condition $\text{cond}1$, in a nutshell, is formed by two main contributions, namely the arguable validity of underlying assumptions and the enforcement of (arguable valid) constraints $\{Cr\}$. Each contribution per se can never be perfect, but could in principle ensure the validity of the system. This approach therefore forms a mutual reinforcing double barrier against unacceptable losses.

As discussed in the foregoing sections (namely sections 2.7 - 3.1), an arguably sufficiently complete set of monitors needs to be derived in order to implement robustness and antifragility. The already introduced assumption monitors $\{\Delta T_{\text{mon}}\}$ need, as argued in the foregoing passages, to be supplemented by satisfaction monitors for constraints $\{Cr_{\text{mon}}\}$.

Robustness and antifragility is achieved by derivation of recovery- and degradation strategies in case a monitor indicates the system being close to or already in violation of the related requirement. Recovery thereby is related to a situation in which normal operation can be re-achieved, whereas degradation relates to a degradation of the systems functionality in order to prevent manifestation.

---

8Suppose all underlying assumptions would be valid. Then the model of reality, reasoning about the system and its context and the derived argumentation for safe operation would be valid. Hence manifestation of unacceptable losses would be prevented. In the case some assumptions might get invalidated, the valid enforcement of the constraints $\{Cr\}$ would still prevent the manifestation.
of unacceptable loss (e.g. transition to safe stop).

\[
\{\text{recDegStrat}\} := \{\text{recovStrat}\} \cup \{\text{degStrat}\}
\]

with assumptions- and constraints related contributions:

\[
\{\text{recovStrat}\} := \{\text{recovStrat}^{\Delta T_{mon}}\} \cup \{\text{recovStrat}^{Cr_{mon}}\}
\]
\[
\{\text{degStrat}\} := \{\text{degStrat}^{\Delta T_{mon}}\} \cup \{\text{degStrat}^{Cr_{mon}}\}
\]

The sufficient representation of the aspects of the validation triangle \(\bar{T}\), as well as the sufficient completeness of the set of constraints \(\{Cr\}\), elicited assumptions \(\{A\}\) and derived assumption monitors \(\{\Delta T_{mon}\}\) need to be argued, based on evidence. As discussed above, the argumentation needs to address aspects such as application of appropriate processes (e.g. for elicitation), sufficient completeness of sets (e.g. the set of assumptions \(A\)) and aspects of robustness and antifragility. We subsume all these aspects of argumentation in the operator \(\triangledown\) and write, e.g. for the set of evidence based arguments related to constraints \(\{evArg^{Cr}\}\):

\[
\{evArg^{Cr}\} := \{Cr\} \triangledown \{evArg\}
\]

The complete set of evidence based arguments is denoted as \(\{evArg\}\), with

\[
\{evArg\} := \{evArg^{\bar{T}}\} \cup \{evArg^{Cr}\} \cup \{evArg^{A}\} \cup \{evArg^{\Delta T_{mon}}\} \cup \{evArg^{Cr_{mon}}\} \cup \{evArg^{\text{recDegStrat}}\}
\]

The argumentation is relative to the representation from which it has been deducted, or reality in the case of the highest level deduction.\(^9\)

There needs to be defined a set of quality criteria \(\{Qc\}\) for the evidence based argumentation allowing to rate the state of the argumentation (i.e. insufficient or sufficient). As discussed in section 2.8, validity of a certain state of development should be ensured (i.e. \(\{evArg\}\) complies with \(\{Qc\}\)) before applying further transformations. Note that the quality criteria for the combination of the diverse and fragmented evidence contributions to an argumentation is closely related to the societal expectations discussed in the foregoing sections. Derivation of \(\{Qc\}\) therefore is a complex task.

For preconditions, we use the notation \(pre\).

**Indirectly enforcing validity**

To sum up, a representation at level \(i\) is given by

\[
\{Rep_i\} := \{\bar{T}_i\} \cup \{Cr_i\} \cup \{A_i\} \cup \{\Delta T_{mon,i}\} \cup \{Cr_{mon,i}\} \cup \{\text{recDegStrat}_i\} \cup \{evArg_i\} \cup \{Qc_i\} \quad (2)
\]

and condition \(cond1\) is enforced by:

\[
\text{cond2} : \forall i \in \{0 \ldots \{Rep\}\} \exists \{Rep_i\}\{evArg_i\}\text{complies with}\{Qc_i\}
\]

which can be stated as: for all underlying representations, the representations are complete according to \(\exists\) and the evidence based argumentation complies with the given quality criteria.

---

\(^9\)Argumentation relative to the representation deduced from follows the well established divide and conquer approach. It is not manageable to argue about every transformation relative to the full implicit reality. Instead, each representation is argued to be valid and following transformations can then be argued relative to this (arguably valid) reference.
Overview of the flow through the algorithms

Figure 5 provides a high level overview of the complex interdependence and mutually recursive flow through the algorithms.

After successful high level initialization (providing a self consistent highest level representation \{Rep_0\}_s.c), a sequence of iterative, intended refinements (subsequent application of algo3a) is undertaken such that an appropriate set of representations (across multiple level of abstractions, according to eq. (2)) is achieved and cond2 (eq. (3)) is satisfied. Algorithm 2 thereby iteratively drives a representation to self-consistency, which may require transformations of the representation (algorithm 3b). Vice versa, a transformed representation needs, just as well, be brought to self-consistency. Therefore, algorithm 2 and 3b are mutual dependent, with possibly several alternating recursions becoming necessary.

Algorithm 1 - high level initialization

Algo1: (initial product idea) → {Rep_0}_s.c.

1. start from initial product idea, derive the high level goal
2. establish general quality criteria for the underlying processes and validity of individual aspects \{Qc\}
3. from the high level goal derive a candidate representation \( \{ \text{Rep}_{\text{cand}} \} \):

(a) carry out initial design domain and functional analysis, set up open model \( \{ \bar{T}_0 \} = \{ \bar{C}_0 \} \cup \{ \bar{P}_0 \} \cup \{ \bar{R}_0 \} \)

(b) apply conscientious process to elicitate related \( \{ A_0 \} \)

(c) carry out analysis of high level unacceptable losses, derive \( \{ \text{Cr}_0 \} \), possibly extending \( \{ A_0 \} \), note this in \( \{ \text{ex}A_0 \} \)

(d) derive \( \{ \Delta \bar{T}_{\text{mon},0} \} \) from \( \{ A_0 \} \), possibly extending \( \{ A_0 \} \), note this in \( \{ \text{ex}A_0 \} \)

(e) derive \( \{ \text{Cr}_{\text{mon},0} \} \) from \( \{ \text{Cr}_0 \} \), possibly extending \( \{ A_0 \} \), note this in \( \{ \text{ex}A_0 \} \)

(f) derive \( \{ \text{recovStrat}_{\Delta \bar{T}_{\text{mon},0}} \} \) and \( \{ \text{degStrat}_{\Delta \bar{T}_{\text{mon},0}} \} \), possibly extending \( \{ A_0 \} \), note this in \( \{ \text{ex}A_0 \} \)

(g) derive \( \{ \text{recovStrat}_{\text{Cr}_{\text{mon},0}} \} \) and \( \{ \text{degStrat}_{\text{Cr}_{\text{mon},0}} \} \), possibly extending \( \{ A_0 \} \), note this in \( \{ \text{ex}A_0 \} \)

(h) refine \( \{ Q_c \} \) if necessary: \( \{ Q_c \} \downarrow \{ Q_{c0} \} \), possibly extending \( \{ A_0 \} \), note this in \( \{ \text{ex}A_0 \} \)

(i) compile evidence based argumentation \( \{ \text{ev}Arg_0 \} \), possibly extending \( \{ A_0 \} \), note this in \( \{ \text{ex}A_0 \} \), related to:

1. derivation process and individual validity of the aspects of the validation triangle

\[
\{ \bar{T}_0 \} \prec \{ \text{ev}Arg \} = \{ \bar{C}_0 \} \prec \{ \text{ev}Arg \}|_{\{ \bar{P}_0 \}, \{ \bar{R}_0 \}}
\cup \{ \bar{P}_0 \} \prec \{ \text{ev}Arg \}|_{\{ \bar{C}_0 \}, \{ \bar{R}_0 \}}
\cup \{ \bar{R}_0 \} \prec \{ \text{ev}Arg \}|_{\{ \bar{C}_0 \}, \{ \bar{P}_0 \}}
\]

ii. elicitation process and individual validity of underlying assumptions \( \{ A_0 \} \prec \{ \text{ev}Arg \} \)

iii. derivation process and individual validity of assumption monitors \( \{ \Delta \bar{T}_{\text{mon},0} \} \prec \{ \text{ev}Arg \} \)

iv. derivation process and individual validity of constraints \( \{ \text{Cr}_0 \} \prec \{ \text{ev}Arg \} \)

v. derivation process and individual validity of satisfaction monitors for constraints \( \{ \text{Cr}_{\text{mon},0} \} \)

vi. derivation process and individual validity of \( \{ \text{recDegStrat}_{\text{Cr}_{\text{mon},0}} \} \prec \{ \text{ev}Arg \} \)

(j) iteratively enhance argumentation quality until \( \{ \text{ev}Arg_0 \} \) complies with \( \{ Q_{c0} \} \); if iteration fails, exit with failure

(k) enter self-consistency loop (algo2) based on the candidate representation for the highest abstraction level \( \{ \text{Rep}_{\text{cand}} \} \) and \( \{ \text{ex}A_0 \} \), which gives:

\( \{ \text{Rep}_0 \} \) s.c. = Algo2(\( \{ \text{Rep}_{\text{cand}} \}, \{ \text{ex}A_0 \})

Algorithm 2 - recursive, self consistent iteration of representations

As it becomes clear from the listing of the high level initialization algorithm, the elements of the representation are mutually dependent. Specifically, the list of \( \{ \text{ex}A_i \} \) grows from algo1 3a) to algo1 3k), with the prior items not being consistent with the complete list of assumptions after a singular pass. Therefore, all elements of the candidate representation need to be checked and possibly extended with respect to all elements in \( \{ \text{ex}A_i \} \). This can extend \( \{ \text{Rep}_{\text{cand}} \} \) and introduce a new
set of \(\{exA_i\}\). Therefore, another cycle of checking the updated \(\{Rep_{icand}\}\) against the updated \(\{exA_i\}\) needs to be started until \(\{exA_i\} = \emptyset\), which implies self-consistency of the representation:

\[
\{Rep_{icand}\} \rightarrow \{Rep_i\} \text{s.c.}
\]

---

Algo2: \(\{\{Rep_{icand}\}, \{exA_i\}\} \rightarrow \{Rep_i\} \text{s.c.}\)

1. set \(\{Rep_{cand}\} = \{Rep_{icand}\}\)
2. set \(\{exA\} = \{exA_i\}\)
3. set \(\{exA_i\} = \emptyset\)
4. For \(j = 1, \ldots, |\{exA\}|\):
   
   (a) check all elements of \(\{\bar{T}_{icand}\}\) with respect to \(exA^j \in \{exA\}\). If affected, transform the element using algo3b such that the resulting representation is consistent with \(exA^j\). I.e.
   
   i. set affected element as focused element for next transformation \(\{f_{in}\}\)
   
   ii. \(\{Rep_{cand}\} = \text{Algo3b}\left(\{Rep_{cand}\}, \{f_{in}\}, exA^j\right)\)

   This possibly updates \(\{Rep_{cand}\}\).

   (b) check and transform \(\{A_{icand}\}\) accordingly; this possibly extends \(\{exA_i\}\) (which was set empty in step algo2 3.)

   (c) check and transform \(\{recDegStrat_{icand}\}, \{Cr_{icand}\}, \{\Delta \bar{T}_{mon,icand}\}, \{Cr_{mon,icand}\}, \{Qc_{icand}\}\)

   and \(\{evArg_{icand}\}\) accordingly, possibly extending \(\{A_{cand}\}\) and \(\{exA_i\}\)

   (d) check and transform \(\{evArg_{icand}\}\) accordingly (applying the sequence alg1.3.i), possibly extending \(\{A_{cand}\}\) and \(\{exA_i\}\)

   (e) iteratively enhance argumentation quality until \(\{evArg_{cand}\}\) complies with \(\{Qc_{cand}\}\); if iteration fails, exit with failure.

   This sequence results in an updated \(\{Rep_{cand}\}\) with respect to \(exA^j\) being the basis for \(j + 1\) and a possibly non-empty \(\{exA_i\}\).

5. The above for loop results in an updated \(\{Rep_{cand}\}\) with respect to \(\{exA\}\) being the basis for the next self consistency cycle and a possibly non-empty \(\{exA_i\}\).

If \(\{exA_i\} = \emptyset\) return \(\{Rep_i\} \text{s.c.} = \{Rep_{cand}\}\)

else set \(\{Rep_{icand}\} = \{Rep_{cand}\}\) and re-start algo2

---

Algorithm 3a - intended transformation of representations

---

Algo3a: \(\{Rep_i\} \text{s.c.} \rightarrow \{Rep_{i+1}\} \text{s.c.}\)
1. \( \text{pre} : \{ \text{evArg}_i \} \text{ complies with} \{ Qc_i \} \land \{ \text{Rep}_i \} \text{ s.c.} \)

From \( \{ T_i \} = \{ C_i \} \cup \{ P_i \} \cup \{ R_i \} \) select one or several aspects as focused elements for next transformation:

\[ \{ f_i \} \subseteq \{ T_i \} \]

2. \( \{ \text{Rep}_{i+1} \} = \text{Algo3b}(\{ \text{Rep}_i \}, \{ f_i \}, \emptyset) \)

3. return \( \{ \text{Rep}_{i+1} \} \text{ s.c.} = \{ \text{Rep}_{i+1} \} \)

### 3.2.1 Algorithm 3b - transformation of representations under condition

\text{Algo3b}: (\{ \text{Rep}_{in} \}, \{ f_{in} \}, \{ \text{exCond}_{in} \}) \rightarrow \{ \text{Rep}_{out} \}

1. set \( \{ \text{exA}_i \} = \emptyset \)

2. For \( j = 1, \ldots, |\{ f_{in} \}| : \)

   (a) derive transformed \textit{representation} of focused element, all other aspects held fixed, such that extra consistency conditions \( \{ \text{exCond}_{in} \} \) are fulfilled:

   \[ f_{in}^j \downarrow \{ f_{in+1}^j \} \big| \big( \{ \text{Rep}_{in} \} \setminus f_{in}^j = \text{const.} \big) \land (\{ \text{Rep}_{in} \} \setminus f_{in}^j \text{ consistend to} \{ \text{exCond}_{i} \} ) \big) \]

   The set notation indicates the fact that the transformation of one element \( f_{i+1}^j \) might lead to a set of elements \( \{ f_{i+1}^j \} \). If impossible, exit with failure.

   (b) set \( \{ \text{Rep}_{out} \} = (\{ \text{Rep}_{in} \} \setminus f_{in}^j) \cup \{ f_{in+1}^j \} \)

   (c) extend \( \{ \text{Rep}_{out} \} \) by

   i. apply conscientious process to elicitate transformation related \( \{ A \} \)

   ii. derive \( \{ \text{recDegStrat} \} \), possibly extending \( \{ A \} \), note this in \( \{ \text{exA} \} \)

   iii. derive \( \{ \Delta T_{mon} \} \) from \( \{ A \} \), possibly extending \( \{ A \} \), note this in \( \{ \text{exA} \} \)

   iv. extend evidence based argumentation \( \{ \text{evArg} \} \), applying the sequence alg1 3.i, possibly extending \( \{ A \} \), note this in \( \{ \text{exA} \} \) following

   v. iteratively enhance argumentation quality until \( \{ \text{evArg} \} \text{ complies with} \{ Qc \} \); if iteration fails, exit with failure.

   (d) This results in an updated \( \{ \text{Rep}_{out} \} \)

   (e) If \( \{ \text{exA}_i \} = \emptyset \) exit

else

   i. set \( \{ \text{Rep}_{icand} \} = \{ \text{Rep}_{out} \} \)

   ii. set \( \{ \text{exA}_i \} = \{ \text{exA} \} \)

   iii. \( \{ \text{Rep}_{out} \} = \text{Algo2}(\{ \text{Rep}_{out} \}, \{ \text{exA} \}) \)

   iv. return \( \{ \text{Rep}_{out} \} \)

21
On exit with failure

Exit with failure might happen at algo1 3.j, algo2 4.e, algo3b 2.a and algo3b 2.c.v. The reason is that the underlying representation can not be brought to consistent validity, which means that an alternative transformation on the current level needs to be found. If this is still not possible, one needs to iteratively step up one layer at a time (↑ \{Rep_{j-1}\}) and try to find a working transformation on that layer. Finding an alternative transformation on a layer close to the failed layer (most preferentially on the same layer) is beneficial, as stepping up one layer in each case renders already invested validation related results from \{Rep_{j-1}\} ↓ \{Rep_{j}\} unused. Typically, for alternative transformations, only a part of these validation results can be reused.

4 Development process, contributions and standards

4.1 Overview

Based on the discussion in the foregoing chapters, we briefly sketch a holistic development process which we refer to as systematic, system view based approach to validation, in short \textit{sys}^2\textit{val}. A detailed presentation however is out of the scope of this paper and might be provided later on. A discussion of published approaches by others, based on our presentation of the validation challenge, will be published later on.

\textit{Holistic} in this context means that the process needs to address the whole process (all aspects of the listing in section 3.1), from initial framing of intention (\textit{high level goal}) over design and implementation to post-release operation, supporting evidence generation from fragmentary, manifold sources (such as simulation, real world driving, etc.), ongoing validation and continuous improvement.

\textit{Sys}^2\textit{val}, just as our general mindset, is strongly influenced by the work of Rasmussen and Leveson [8, 9, 2]. We build on this strong basis and extend it to enhance traceability, completeness- & validation argumentation and applicability for complex systems operating in open contexts.

On the highest level, a holistic process should support framing of the intention (i.e. high level goal definition), taking in to account all relevant aspects (such as system safety, safety of the intended functionality, functional safety, security, product liability, etc.) listed in section 2.1.

Next, just as in Leveson’s approach, the ’what and where should it be done’ (high level purpose and operational design domain analysis) is initially separated from the aspect of ’what should not happen in order to not compromise the high level goal’. This analysis is effect based (i.e. based on high level accidents / unacceptable losses). It necessarily is - as far as possible - independent from specific solution approaches and details of implementation. The aim is to provide general applicable constraints - rated by a related risk level, to the following design and implementation without reducing the accessible solution space right from the start (maximize area 7, point 7 in the listing at end of section 3.1). On the high level of abstraction, an arguable complete set of hazardous system states and related constraints can be determined, which is easily comprehensible. The derived constraints contribute to the practically only indirect enforcing of \textit{cond1} via eq. 3 and \textit{cond2}, as discussed in section 3.2.

One of the basic aspects of \textit{Sys}^2\textit{val} is to apply what we refer to as \textit{open model approach}, which is a model approach coping with the complexity of open contexts, allowing for iterative return to the model building, validation and refinement without the need to fully explicit modeling of e.g. the operational design domain (which would be impossible in an open context). As discussed in
the foregoing sections (see the discussion of figure 3 in section 2 and algo1 3.i and ,b; algo2 4.e and algo3b 2.c.v in section 3.2), providing validation argumentation on each layer of abstraction (in each step of iterative refinement) is a basic necessity. The high level purpose-, operational design domain- and constraint analysis forms the first level of concretization (open model iteration), based on the high level goal.

The high level operational design domain analysis and the high level system hazards then form the basis for the system level hazard and risk analysis. The system h&r provides hazardous event characterization and relation to the constraints and thereby identifies design drivers without reducing design freedom (see 7. in the listing in section 3.1).

Following this, the necessarily creative act of system design can be approached in iterative, mutual dependent concretization of all aspects (T(R, C, P)). This is done by applying valid deductive steps, addressing the deductive gap, as discussed in 2.7 and providing input for robustness and anti-fragility (point 3 to 6 in the listing at end of section 3.1) by following the algorithms stated in section 3.2, triggering necessary refinement of the open models and supporting ongoing validation.

With respect to preexisting components (e.g. radar or lidar sensors), we suggest what we refer to as contribution analysis: the possibly to be used component from a lower level of abstraction might be analyzed regarding the underlying assumptions, possible insufficiencies and consequences for validation etc. The open models can then be refined with respect to the related aspects (e.g. reflectivity coefficients, presence of metallic structures in the operational design domain, etc.) and design decisions can be taken on a higher level of abstraction, based on this. The contribution analysis steps are an important input for design for validation and allow steering high level design from a perspective of 'what can be validated and what can be built'. This is necessary, as it makes no sense to design a system purely top down, finally ending up with a solution that can either not be build or hardly validated.

4.2 Contributions to the holistic evidence generation

We comment only briefly on the contributions to the holistic evidence generation cycle necessary for the validation of complex systems operating in an open context (related to algo1 3.i and algo1 3.j, algo2 4.e and algo3b 2.c.v in section 3.2). The main contributions are from

- system understanding
- simulation
- real world observation
- continuous observation (pre- and post-release e.g. in the style of a control and observation center)
- continuous feedback to system understanding

It is important to note that all individual contributions can never be used as silver bullet (i.e. singular approach for the complete validation challenge). Due to the characteristics of the problem elaborated in the foregoing sections, every possible contribution per se can never be complete. For complex systems operating in an open context, the individual contributions necessarily need to be mutually reinforced by combination with the others. Only closing the whole cycle in a well balanced form across all aspects listed above will be appropriate to address the validation challenge.
However, the result will only be as good as the individual contributions. In other words, excessively investing in singular aspects (e.g. simulation) and neglecting others will be inappropriate.

As an example, we comment on obvious, however problematic statistics based silver bullet approaches in the sense of ‘driving X hours or miles’ to demonstrate the validity of the system, which have initially been discussed for autonomous driving, but never seriously been applied (as silver bullet). As should have become clear so far, the characteristics of the validation challenge, especially the representativeness challenge and the problems related to extremely rare, but systematic manifestation of unacceptable losses, render such an approach fundamentally intractable. Nevertheless, magic numbers X could be proposed and even reduced in size by superficial statistic arguments, e.g. about data fusion. Besides the fact that, due to the aforementioned reasons already the starting point of this argumentation, namely the magic number X, would be basically questionable, a conscientious analysis shows that the basic problem of X being large would even be irreducible for realistic systems within a silver bullet approach. For example, a perfect independence of fused data cannot ad-hoc be argued. However, demonstrating even a certain level of independence already poses a problem even larger than X (see [10], especially section 5 for a detailed discussion). Statistic arguments however, when being part of the holistic evidence cycle, have an important contribution to validation.

The general process sketched in section 4.1 and formalized in section 3.2 forms a framework to handle the fragmentary, iteratively refined knowledge about the operational design domain, the system design (including design alternatives), and implementation (i.e. it addresses C, P and R as well as the necessary validation argumentation across all levels). This provides an adequate understanding of the open context total system (e.g. ego vehicle and relevant part of the surrounding), the interactions and mutual dependencies, as well as potential insufficiencies (i.e. assumptions being temporarily or permanently invalid). From this, necessary evidence contributions and hence validation and verification activities can be derived. System understanding therefore is one basic pillar of the holistic evidence generation cycle. It addresses subareas 1-4, 7-11 and the fuzzy edge of $(\bar{T}+\text{loss})$ in figure 4 and prepares metrics for related evidence generation.

Another contribution comes from simulation. A full simulation of open context systems is intractable. However, it has a strong contribution in developing understanding on certain levels e.g. sensor insufficiencies or behavior simulation. In addition, simulation is well suited for scenario analysis (what if) and therefore can provide evidence for the support of the worked out robust and anti-fragile design, metrics, recovery and degradation strategies etc. With other words, it strengthens the arguments formed in the system understanding pillar.

Real world observation, on test tracks or real context, with a small number of ego entities or connectivity based approaches across larger fleets (pre- and post-release), on the other hand, can provide contributions about aspects which can not easily be formally analyzed in detail (system understanding) or simulated. This inability to simulate might be due to sheer complexity being intractable (e.g. full ray-tracing for sensors) or aspects unwittingly missing in $\bar{T}$. Real world observation therefore is the only means able to address area 5 and 6 in figure 4. On the other hand, the representativeness challenge makes it hard to representatively sample the open context for all events relevant for the complex system.

Therefore, being prepared for the unexpected and a continuous observation in the sense of ongoing validation (based on system understanding) and continuous feedback to system understanding complete the holistic cycle. The feedback allows to continuously confirming or refining the system understanding, possibly triggering rework for improvement (restarting next cycle iteration), as well as to strengthen the evidence for the validity of the system.
4.3 Contribution of from ISO26262 and SOTIF

With respect to ISO26262 and SOTIF, one needs to distinguish between what is requested (i.e. which problem is posed) and which part of the posed problem is provided support for.

ISO26262 and SOTIF both are focused only on a part of the unacceptable losses, namely harm of persons related to E/E systems. All the other aspects of a valid product discussed in section 2.1 are out of scope.

An adequate understanding of the necessary functionality (of the item) and the interaction with the environment and other items is requested by ISO26262 (see. e.g. ISO26262 section 3.5), but regarded as to come from external source. In other words, an adequate system understanding and item definition (related to $T$) is required, but generation is not supported. There is a good support for addressing systematic development as well as hardware failures, with failure (I) defined as termination of the ability of an element, to perform a function as required. This is related to the crossover between area 1 and 2 in figure 4. Broadly speaking, given an adequate item definition, everything which follows therefrom and can be addressed by verification (well defined starting point and iterative crossover to well defined concretizations) is supported. For non-complex systems in a defined context, this might be appropriate. However, for complex systems operating in an open context, as argued in the foregoing sections, a strong support especially for validation issues would be mandatory.

ISO26262 request a safety validation (I) - assurance, based on examination and tests, that the safety goals are sufficient and have been achieved. In addition, functional safety (I) is understood as absence of unreasonable risk (i.e. related to harm of persons) due to hazards caused by malfunctioning behavior of E/E systems, with malfunctioning behavior (I) defined as failure or unintended behavior of an item with respect to its design intent. Unintended behavior ($T^\dagger \leftrightarrow T$) with respect to design intent ($T^\dagger \leftrightarrow T^\dagger + \text{loss}$) is related to area 5 in figure 4. Therefore, ISO26262 in fact already poses the full problem, however, as already stated above, does not provide support for addressing all related aspects.

SOTIF is an approach to overcome the limitations of ISO26262 (currently for driver assistance systems). The PAS (in preparation) states:

> For some systems, which rely on sensing the external or internal environment, there can be potentially hazardous (related to harm of persons) behavior caused by the intended functionality or performance limitation of a system that is free from the faults addressed in ISO26262.

Hence, safety of the intended functionality (I) is defined as absence of unreasonable risk (related to harm of persons) due to hazards resulting from functional insufficiencies of the intended functionality or from reasonably foreseeable misuse by persons.

As already discussed in the problem statement (section 2), SOTIF’s functional insufficiency relates to what we more generally refer to as insufficiency, fundamentally related to invalid assumptions. The SOTIF process - in contrast to the ISO26262 explicitly addresses the iterative refinement of the functional- and system specification with respect to insufficiencies (i.e. invalid, possibly implicit assumptions). See e.g. ISO/PAS 21448’s figure 9, flowchart of the activities, appended for reference as figure 6 in the appendix. The step from activity 'start' to 'functional and system specification' relates to $\text{sys}^2\text{val}$’s high level goal definition followed by the valid deductive step supported 'well controlled concretization' of the aspects $P, R$ and $C$ of the validation triangle (i.e. iterative deduction of $\tilde{T}$, including contribution analysis steps). Section 5 of the PAS provides
a listing of elements which should be part of the resulting specification. The sys² val approach sketched above provides support for the generation of these and further elements necessary for complex systems operating in open contexts.

When over-viewing the process of the PAS activities as depicted in the flowchart, it is important to note that the focus actually is on insufficiencies (i.e. invalid assumptions) and analysis is based on system understanding. Triggering events should be regarded as subordinated (with respect to insufficiencies). They are representative events stimulating the related insufficiencies and hence possibly leading to harm of persons. With subordinated we refer to the fact that the fundamental basis is formed by analysis of insufficiencies and not by triggering events. A real world observation focused, triggering event based silver bullet like approach would be inappropriate, at least for complex systems operating in open contexts. In the sense of the holistic evidence generation cycle (section 4.2), real world observation and collection of triggering events in order to enrich and refine system understanding however is an important part of a larger overall approach.

With this in mind we can regard area 1 of the flow chart as generation of a collection of representative triggering events (the actual aim is to achieve a good assumption coverage) and in this sense being related to the activities targeted at enlarging area 1 and the assumption monitoring circular ring, especially area 4 of our figure. However, we, once again, refer to the representativeness challenge. Assuming triggering event representativeness would be possible, the PAS flowcharts area 2, especially the reiteration of the SOTIF process via 'functional modification', relates to activities intended to transform the system such that subarea of our area 4 is converted to area 1 in figure. A strong support in doing so, however, is currently not part of the PAS.

And finally, PAS flowcharts area 3, especially the reiteration via 'functional modification', relates to activities intended to transform the system such that subarea of our area 5 is converted to area 1 in figure. A strong support in doing so, however, is currently not part of the PAS.

It needs to be noted that the ISO/PAS 21448 is strongly influenced by the state of the art approach for assistance systems, whereby statistic methods play an important role. In fact, the ISO/PAS 21448 currently is aimed solely for the application of level 2 (driver assistance) systems. Preparation for an extension towards level 3+ is currently in discussion.

Based on our problem statement and comparing the PAS activities to the necessities discussed in the foregoing sections, it becomes clear that the current draft of the ISO/PAS 21448 leaves open a gap with respect to complex systems operating in an open context. In addition to the discussed aspects in this section, ongoing validation and 'being prepared for continuous improvement' (antifragility) play an important role for more complex systems validation and should not be neglected when extending ISO/PAS 21448.

5 Conclusion

A fundamental understanding of the validation challenge related to design and operation of complex systems in open contexts is mandatory, in order to be able to provide strategies for 'viable' validation and approval of such systems. We gave a detailed problem statement, formalization thereof and formalized algorithms for iterative development and validation, based on this. Due to the complexity of the validation issue, this formalization is regarded mandatory in order to ensure that none of the many subtle details are neglected in development process design. In addition, we provide a high level overview of a practical, holistic development process (sys²val) and comment on the contributions from ISO26262 and current state of ISO/PAS 21448 (SOTIF).
From a practical point of view, basic challenges arise from managing the large number of fragmentary and iteratively growing knowledge (see all the interrelated aspects being part of the representation on every level of abstraction \( \{ \text{Rep}_i \} \), eq. 2). The sheer mass and complex interrelation of the knowledge fragments (besides the complex elicitation and evidence generation processes) poses demands on knowledge engineering approaches and -tools, which is well beyond the state of the art.

Another basic challenge is due to the necessity of integrating fragmentary and manifold evidences to an overall approval argumentation on the one hand, and the actual validation goal \((\dot{T} + \text{loss})\) being a moving target strongly related to societal factors. As of today, a unified theory of evidence integration (e.g. integrating probabilistic and systematic aspects) is not available. This however would allow to formalize the state of - and guide the only just started societal discussion, and hence provide formalized and widely accepted quality criteria \( \{ \text{Qc} \} \) (see section 3.2).

Solving these challenges is necessary to successfully demonstrate safe operation and strict avoidance of fatal incidents in all day use, which is mandatory to ensure societal acceptance of highly autonomous systems operating in open contexts, on the long run and preventing a ‘winter of autonomous systems’.

6 References

References

[1] Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua: On a Formal Model of Safe and Scalable Self-driving Cars. arXiv preprint arXiv:1708.06374 (2017).

[2] Nancy Leveson: Engineering a safer world: Systems thinking applied to safety Mit Press (2011).

[3] B. Spanfelner, D. Richter, S. Ebel, U. Wilhelm, and C. Branz, W. Patz: Challenges in applying the ISO 26262 for driver assistance systems. (2014).

[4] Eetu Heikkilä, Risto Tuominen, Risto Tiusanen, Jakub Montewka, and Pentti Kujala: Safety Qualification Process for an Autonomous Ship Prototype—a Goal-based Safety Case Approach. Marine Navigation, 365–370 (2017).

[5] Nassim Nicholas Taleb: Anti-fragile: How to Live in a World We Don’t Understand volume 3 Allen Lane London, UK (2012).

[6] Nassim Nicholas Taleb and Raphaël Douady: Mathematical definition, mapping, and detection of (anti) fragility. Quantitative Finance 13(11), 1677–1689 (2013).

[7] Nassim Nicholas Taleb: Silent risk: Lectures on fat tails,(anti) fragility, and asymmetric exposures. Available at SSRN (2014).

[8] Jens Rasmussen: Skills, rules, and knowledge; signals, signs, and symbols, and other distinctions in human performance models. Systems, Man and Cybernetics, IEEE Transactions on (3), 257–266 (1983).

[9] Jens Rasmussen, Annelise Mark Pejtersen, and Len P Goodstein: Cognitive systems engineering. (1994).
7 Appendix

Figure 6: Flowchart of the ISO/PAS 21448 activities (figure 9 in the draft version of ISO/PAS; changes to the final publication might apply.)