

Model-based Application of ISO 26262: The Hazard Analysis and Risk Assessment

Marion Suerken and Thomas Peikenkamp
OFFIS e.V.

ABSTRACT

New drive concepts as well as advanced driver assistance systems and revised basic functions belong to the main innovation areas in today's automotive industry. Since thereby more and more electric / electronic components are integrated in modern passenger cars, the complexity is steadily increasing. Additionally, a crucial part of the innovations focuses on active and passive safety. Together, this creates tough demands on systems reliability and functionality which influence the development.

Addressing the development process of electric / electronic components for passenger cars, the ISO 26262 "Road vehicles - Functional safety" came into effect in November 2011 ([1]). This standard introduces a safety lifecycle which "encompasses the principal safety activities during the concept phase, product development, production, operation, service and decommissioning" ([1], part 2, p.3) and which can be seen as a guideline that demands a risk-based development approach with seamless traceability.

Already in the beginning of this safety lifecycle, in particular during the concept phase (cf. [1], part 3), a hazard analysis and risk assessment needs to be performed for the item under development. Within this process, hazards are identified, combined with possible operational situations for the car, and relevant outcomes of these combinations, namely the hazardous events, are classified to determine the automotive safety integrity level (ASIL) for the item.

In this paper we address this hazard analysis and risk assessment. We developed a formalized hazard description language that enables a precise formulation of hazardous events based on four potential contributing factors. Moreover, it is shown how safety goals can be derived from these formalized hazardous events. In a further step it is described how the concepts presented in the paper on the one hand permit checking the implemented safety measures with respect to the risk mitigation which is achieved and on the other hand enables the application of model-based safety analyses in order to verify safety characteristics.

CITATION: Suerken, M. and Peikenkamp, T., "Model-based Application of ISO 26262: The Hazard Analysis and Risk Assessment," *SAE Int. J. Passeng. Cars – Electron. Electr. Syst.* 6(1):2013, doi:10.4271/2013-01-0184.

INTRODUCTION

Taking a closer look to the automotive industry shows that more and more electric/electronic components are installed in modern cars. One reason for this is that today's innovations focus on new drive concepts like hybrid vehicles or electric vehicles as well as advanced driver assistance systems which support the driver to fulfill the task of responsible driving. Those advanced driver assistance systems in particular evoke additional hardware and electronic control units which interact with existing electronic control units. The adaptive cruise control, as one example, leads to an additional radar sensor and a control unit which determines the distance to the car running ahead and calculates system reactions which have the form of reducing

or increasing the velocity of the ego-vehicle. This, in turn, needs to be processed by the motor control unit.

As a matter of course, the increasing complexity creates higher demands on reliable systems interacting with their environment. This focuses especially on the functional safety which is defined as the absence of unreasonable risk. In order to achieve functional safety, appropriate analyses need to be carried out during the development process. In November 2011, the ISO 26262 "Road vehicles - Functional Safety" came into effect. This standard for electronic/electric components used in passenger cars is an adaptation of the IEC 61508 for the automotive industry. It addresses the upcoming challenges and introduces a safety lifecycle which "encompasses the principal safety activities during the concept phase, product development, production, operation,

service and decommissioning” ([1], part 2, p.3) and which can be seen as a guideline that demands a risk-based development approach with seamless traceability.

However, not only the demands on functional safety are increased with the higher complexity. Also the demands on the development process as such are getting tougher since more complex systems need to be adjusted with respect to their interaction. Aiming to fulfill the demands, model-based design / development is getting more and more importance in the automotive industry. The main advantages of this approach are the following:

- It supports “front loading” by providing models for all typical development artifacts, at the same time enabling (partially) automated checks of properties of these models and their interrelationships.
- It usually provides the “separation of concerns” idiom in the sense that typically for each concern a particular model can be defined. Here also the integration of concerns can be addressed, allowing - for instance - to relate the functional design to the technical design level.
- Requirement engineering becomes an integrated discipline, since the subjects of requirements become explicit as (parts of) models. Verification and Validation (V&V) activities can be focused to locate and correct errors as early as the subjects of requirements become available.
- Reuse as well as replacement of system parts can be efficiently addressed with contract-based specification methods, increasing the efficiency of reuse on the V&V side as well as minimizing the effect of replacement.

The approach of model-based design is accompanied with a proper requirement engineering which enables a precise formulation of the requirements assigned to a system, component or functionality. Moreover, an integrated approach of requirement formulation on all levels facilitates the analysis if one level of abstraction is correctly pictured by the refinement.

Within this paper one particular aspect of the safety lifecycle introduced in ISO 26262 is examined, namely the Hazard Analysis and Risk Assessment. In a first step other work related to this topic is presented in order to give an overview about already existing approaches. Subsequent to that, the related standards ISO 26262 (Road Vehicles - Functional Safety) and ISO / IEC 29148 (Requirements Engineering) are described briefly. After presenting the key aspects needed in a model in the section hazard analysis and risk assessment the process required by ISO 26262 is described. Subsequent to that, the item definition which is also required by ISO 26262 is used to introduce the example of a steering system used in this paper. Based on that the hazard description language, the classification of hazardous events and the derivation of safety goals are described in single subchapters. In an additional subchapter it is outlined how the concepts available in EAST-ADL can be used to represent hazardous events in a model. Moreover, an outlook

on further research work including the adaption of already existing analysis techniques which are enabled to be applied with using the concepts described in this paper is given. Finally, a conclusion and discussion about the topic is presented.

RELATED WORK

Subsequent to the introduction to the paper in a first step the presented work will be set in the context of related work. In the following subchapters, the related work is presented in a brief overview.

Automotive Generic Hazard List

The Automotive Generic Hazard List [2] has been defined on the basis of the approach of hazard lists used in military as well as railway applications and provides a means to structurally identify hazardous events. Currently, the list is specialized for the analysis of Advanced Driver Assistance Systems (ADAS).

In general the AGHL has the form of a matrix in which variables assigned to the system environment can be associated with variables of the classes operation and accident. In [Figure 1](#) an excerpt from an AGHL is shown.

Within the presented matrix, the hazardous events, which are combinations of the three factors system environment, operation, and accident, are labeled with a number. For every number stated in this matrix, there need to be a textual description in an additional document. This description might look like this:

possibility for collision with another vehicle due to undesired traffic routing and wrong setpoint value [2]

This paper focuses more on the precise formulation of hazardous events. Thus, the use of the generic hazard list provides a useful complementary tool for the initial identification of hazards.

		Operation						Unfall			
		Schutzoperation & Funktionen						Kollision		Unerwünschtes Abkommen von der Fahrspur	Andere Unfälle
		Information		Vergleich	Intervention						
		Sollwert	Istwert	Soll- mit Istwert	Geschwindigkeitskontrolle	Richtungslenkung	mit FZG	nicht mit FZG		Feuer	Natur
Infrastruktur	Trassierung	x					1.1	2.1	3.1		
			x				1.2	2.2	3.2		
				x			1.3	2.3	3.3		
					x		1.4	2.4	3.4		
						x	1.5	2.5	3.5		
							1.6	2.6	3.6		
	Fahrbahnbeschaffenheit	x					1.7	2.7	3.7		
			x				1.8	2.8	3.8		
				x			1.9	2.9	3.9		
					x		1.10	2.10	3.10		
						x	1.11	2.11	3.11		
		x					1.12	2.12	3.12		

Figure 1. Excerpt from AGHL, picture taken from [2]

Requirement Specification Language

Stating requirements is one of the essential tasks in developing safety related systems. These requirements constitute an interface between different groups of people, i.e. customers, engineers, project managers and many more. Typically requirements are stored as natural language text which has the disadvantage that ambiguities of the sentences can cause a huge amount of requirement changes in early as well as in later development phases. To reduce the costs incurred by these ambiguities, formal languages have been used for several years to ensure that requirements have a well-defined semantic interpretation. Starting with the theoretical groundwork these languages have been tuned to more comprehensive versions suitable for applications in engineering domains. A typical representative of these languages is the Property Specification Language [3] developed by Accelera which is targeted to the formal specification of hardware. It has been developed in order to overcome the shortcoming of ambiguous and unverifiable requirements formulated in natural language. With the help of such formal language the design engineer is able to formulate requirements with respect to the behavior of the design (also internal behavior) as well as about the environment in which the design shall operate using a concise syntax with clear semantics underlying. In addition, since it is an extension of the standard temporal logics all defined properties can be translated to LTL or CTL, respectively. The advantages which come in hand with using this specification language are that by using this mathematically precise, formal language it gets on the one hand easier for the design engineer to write down verifiable requirements. On the other hand that it enables the verification engineer to validate via simulation or formal verification like for example model-checking.

In recent years such languages have been further tuned to reduce the hurdle of using them in real life requirement specifications. The main idea is to provide unambiguous patterns of natural language text that can be translated to one of the formalisms mentioned above. These patterns consist of static text elements and attributes being filled in by the requirements engineer. Each pattern has a well-defined semantic in order to ensure a consistent interpretation of the written system specification across all project participants. On the one hand this limits the possibilities of writing a requirement; on the other hand it prevents misunderstandings regarding the interpretation of the sentences. To gain a set of unambiguous requirements this limitation is necessary. However, writing requirements shall still be possible in an intuitive way.

To gain a high degree of intuitivism the language consists of only a few constructs that can be easily remembered to give the requirement engineer the possibility to fully understand the requirement specification language (RSL), which has been developed within the CESAR project [4], and choose the right pattern that fits the properties he wants to

demand on the system. There are patterns for each of the following categories available [4]:

- **Functional Patterns:** These Patterns express functional requirements of the system. This includes the relationship between events, handling of conditions and invariants and the possibility to define intervals in which the requirements or parts of them are valid.
- **Probability Patterns:** (Not only) in the ISO 26262 it is necessary to express quantitative failure or hazard characteristics. Since there are various partly redundant forms of expressing probabilities that are used quite ambiguous in common speech, these patterns guide the requirements engineer to express his needs.
- **Safety Related Patterns:** Only a small part of safety related requirements can be covered with probability patterns. The major part of the requirements outlines relationships between system components. These patterns enable the specification of single point of failures or other failure and hazard dependencies between different components.
- **Timing Patterns:** These patterns can be used to describe real-time behavior of systems. This includes periodic and aperiodic activations, jitter or delay.
- **Architecture Patterns:** These patterns specify the existence of architecture elements like components, events or connections between components.
- **Mapping Patterns:** These patterns can be used to express requirements on the mapping from the functional design perspective to the physical design perspective.

Patterns are noted in a form with optional parts that are not necessarily instantiated. A functional pattern describing the causality between two events looks like this:

whenever *request* **occurs** *response* **occurs** [*during interval*].

Phrases in square brackets are considered optional, bold elements are static key-words of the pattern and italic printed elements represent attributes that have to be filled out by the requirements engineer. When filling the attributes there are no general restrictions on the names.

As a key element of every requirement engineering methodology it must be possible to steer the architectural design and to allocate derived requirements to the architecture. In the following section we show how to exploit an automotive-specific architecture description framework. However, for other purpose it is sufficient to have a framework that allows to create an architecture out of self-contained components and distinguishes between functional and technical viewpoints.

Architecture Modeling with EAST-ADL

EAST-ADL is an architecture description language that has been developed in various projects jointly by automotive OEMs, suppliers and tool vendors. The objective thereby is to



define an architecture description language tailored to the needs of the automotive industry. (cf. [5]). Similar to the concept of architecture modeling, also in EAST-ADL different levels of abstraction are introduced:

- Vehicle level,
- Analysis level,
- Design level,
- Implementation level, and
- Operational level.

With the hierarchical modeling concept which develops through the different abstraction levels the complexity of systems can be controlled more easily. (cf. [5])

Within the realm of the formalization of hazardous events especially the error modeling in connection with the function modeling is of interest. In addition to the nominal design model which can be captured in the different levels of abstraction EAST-ADL provides the possibility to have one or more error models in parallel which describe the abnormal behavior of the target architecture. The error model also contains special fault/failure ports which can be mapped to the respective ports of the system architecture. (cf. [6])

RELEVANT STANDARDS

For the proposed methodology we are mainly building on two international standards, namely the ISO 26262 (Road vehicles - Functional safety) and the ISO/IEC/IEEE 29148 (Systems and Software Engineering - Life Cycle Processes - Requirements Engineering). Both standards will be briefly introduced in the following subsections.

ISO 26262: Road Vehicles - Functional Safety

As already mentioned the ISO 26262 [1] came into effect in November 2011. The standard, in general, addresses the development of electric/electronic systems installed in passenger cars with a maximum gross mass of 3500 kg. The standard is subdivided in ten parts, whereat the first part determines the vocabulary and the last part, part 10, is an informative guideline about how to apply the ISO 26262. The parts framed by the ones mentioned have the following titles:

- Part 2: Management of Functional Safety
- Part 3: Concept Phase
- Part 4: Product Development - System Level
- Part 5: Product Development - Hardware Level
- Part 6: Product Development - Software Level
- Part 7: Production and Operation
- Part 8: Supporting Processes
- Part 9: Automotive Safety Integrity Level (ASIL)- oriented and Safety-oriented Analyses

Within the eight normative parts (excluding the vocabulary, which is normative but contains no direct requirements, and the guideline of part 10 which is informative) the management of functional safety from development to decommissioning is regulated. In order to manage functional safety, ISO 26262 provides an approach for a safety lifecycle which can be tailored to the actual product development. Moreover, it defines confirmation measures which include safety audit, review and assessment for the phases of the lifecycle.

The hazard analysis and risk assessment which is targeted with the development of a hazard description language is described in the Concept Phase (Part 3). A detailed description of the requirements to this topic is given in section *hazard analysis and risk assessment*.

ISO/IEC/IEEE 29148: Systems and Software Engineering - Life Cycle Processes - Requirements Engineering

As already mentioned in section *requirement specification language* stating requirements is one of the essential tasks in developing safety related systems and - as suggested in the following sections - we recommend to use the same principles when stating the hazards and hazardous events. These principles are given in the ISO/IEC/IEEE 29148 targeting the requirements engineering and “provides a unified treatment of the processes and products involved in engineering requirements throughout the life cycle of systems and software” [7]. Within these standard properties of a “good” requirement and the application of requirements processes in the lifecycle are described. Moreover, the general requirements engineering process as well as the management of activities related to requirements are accompanied with guidance on application. Although being dedicated to system and software, the standard provides a good overview on how requirements should be formulated in general. The characteristics of individual requirements are specified as necessary, implementation free, unambiguous, consistent, complete, singular, feasible, traceable, and verifiable. The concrete meaning of this attributes for the specification of requirements can be read in [7].

Focusing on the expression of hazardous events, all characteristics are also important, however, the reason why is slightly different. In the following the description for the aspects in the realm of stating hazardous events is presented.

- **Necessary:** The hazardous event defines a potential source of harm. If deleted or left out, this would lead to an unknown source of harm which is not handled.
- **Implementation free:** Hazardous events need to be defined subsequent to the item definition. Since at this level only the functionality as well as the limitations and boundaries are known, the formulation of a hazardous event should be independent from implementation