Analysis of ISO 26262 Compliant Techniques for the Automotive Domain

S. Manoj Kannan[1], Yanja Dajsuren[2], Yaping Luo[1], and Ion Barosan[1]

[1] Eindhoven University of Technology, Eindhoven, The Netherlands
[2] Centrum Wiskunde & Informatica, Amsterdam, The Netherlands
m.k.soundarapandian@student.tue.nl, y.dajsuren@cwi.nl
y.luo2@tue.nl, i.barosan@tue.nl

Abstract. The ISO 26262 standard defines functional safety for automotive E/E systems. Since the publication of the first edition of this standard in 2011, many different safety techniques complying to the ISO 26262 have been developed. However, it is not clear which parts and (sub-) phases of the standard are targeted by these techniques and which objectives of the standard are particularly addressed. Therefore, we carried out a gap analysis to identify gaps between the safety standard objectives of the part 3 till 7 and the existing techniques. In this paper the results of the gap analysis are presented such as we identified that there is a lack of mature tool support for the ASIL sub-phase and a need for a common platform for the entire product development cycle.

Keywords: ISO 26262, vehicle safety, safety standard, gap analysis

1 Introduction

Development of innovative features such as advanced driver assistance systems in modern day automobiles have led to an increased complexity in product development and maintenance. This imposes an increased risk in terms of system failure that could lead to unacceptable hazards. Thus it becomes crucial to ensure functional safety. The ISO 26262 standard [15] defines functional safety for automotive Electric/Electronic (E/E) safety-related systems. Its objective is to address possible hazards caused by the malfunctioning behavior of E/E systems throughout the product development cycle.

Most of the automotive companies have already started using safety analysis, verification and validation techniques to ensure vehicle safety [22]. One of the main objectives of the ISO 26262 is that these techniques should be applied as a standardized methodology for all automobile manufacturers. These techniques focus mainly on traceability which is the ability to track the safety requirements from initial concept design till the production and operation phase. Upon trying to improve the traceability, the researchers seek more techniques for effective product development process.

The introduction of the ISO 26262 functional safety standard provides more specific development processes that help to avoid the hazards and threats in

the development phases. Following steps should be taken to ensure compliance with the standard: a) The manufacturers should adopt the development processes; b) The manufacturers should determine the Automotive Safety Integrity Level (ASIL) for safety-critical systems; c) The manufacturers should satisfy the additional requirements.

The standardization process requires the consistency of methods, languages and tools across all the sub-phases of the software lifecycle as well as system and hardware development phases as stated in the section 5.4.4 of the ISO 26262 Part 6 [15, p. 4]. In recent years, safety related platforms such as OPENCOSS [6] and AutoFOCUS3 [2] have been developed. OPENCOSS provides a common safety certification platform for the railway, avionics and automotive markets. AutoFOCUS3 provides a model-based tool for distributed, reactive, embedded software systems. The consistency can be assured through the availability of a tool that ensures the compatibility within the ISO 26262 (sub-) phases. The automobile manufacturers are challenged in the selection of the optimal techniques to ensure this compatibility which helps to prove the functional safety. This paper focuses on examining the gap between the ISO 26262 standard objectives and state-of-the-art safety related techniques.

The remainder of the paper is organized as follows: In Section 2 we provide background information on the V-model of the ISO 26262 standard. In Section 3, we describe the systematic literature review process and the summary of the papers selected for the analysis. Section 4 presents the gap analysis results and Section 5 discusses the findings. Finally, we present the concluding remarks and some related future works.

2 Background

The safety standard ISO 26262 [15] is an adaptation of the functional safety standard IEC 61508 [14] for automotive E/E systems. Similar to IEC 61508, ISO 26262 is also a risk-based safety standard. It provides a risk-driven safety life-cycle for developing safety-critical systems in the automotive domain.

The ISO 26262 consists of ten parts as shown in Figure 1. Part 1, 2, and Part 8 to 10 are out of the scope of this paper, because Part 3 to Part 7 correspond to the safety life-cycle. The main part of ISO 26262 is structured based upon the V-model, as well as Part 5 and Part 6. Part 3 and Part 7 focus on the vehicle level. The main goal of Part 3 is to identify system hazards and risks through Hazard Analysis and Risk Assessment (HARA), then derive safety goals and Functional Safety Concepts (FSC) from them. Part 4 focuses on the system level. In this part, Technical Safety Requirements (TSR) are derived from FSC. Then system design can be carried out based on TSR. Part 5 and Part 6 focus on the subsystem/component level. In these two parts more detailed safety requirements are derived from TSR. Those safety requirements are assigned to the concrete subsystems or components for implementation.

In the following section, we present state-of-the-art techniques complying to the ISO 26262 standard.

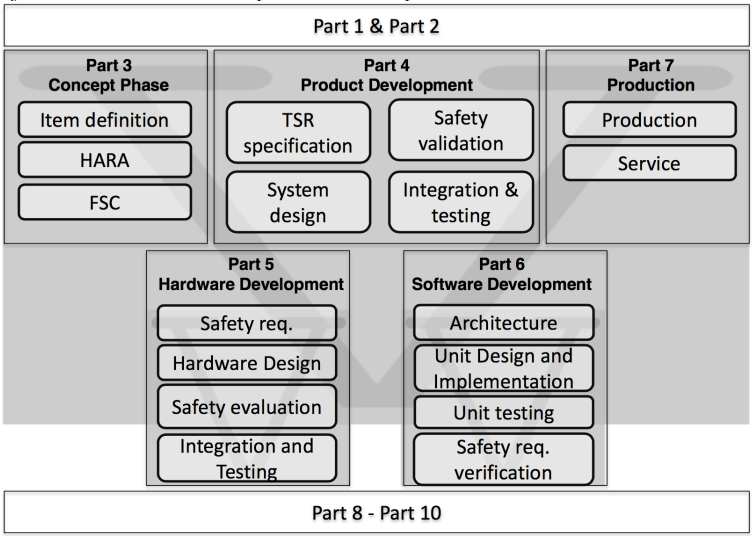

Fig. 1. An overview of the ISO 26262 V-model

3 Systematic Literature Review

We use a Systematic Literature Review (SLR) methodology [16] to obtain state-of-the-art information on the techniques in the area of the ISO 26262 standard. In a comprehensive SLR analysis, documents that contain ISO 26262 related information are analyzed. Sources are collected from various popular resources such as IEEE [5], ACM [1], Springer [8], SAE [7], and FISITA [3]. Scientific journal articles, research papers, and industrial technical reports are considered.

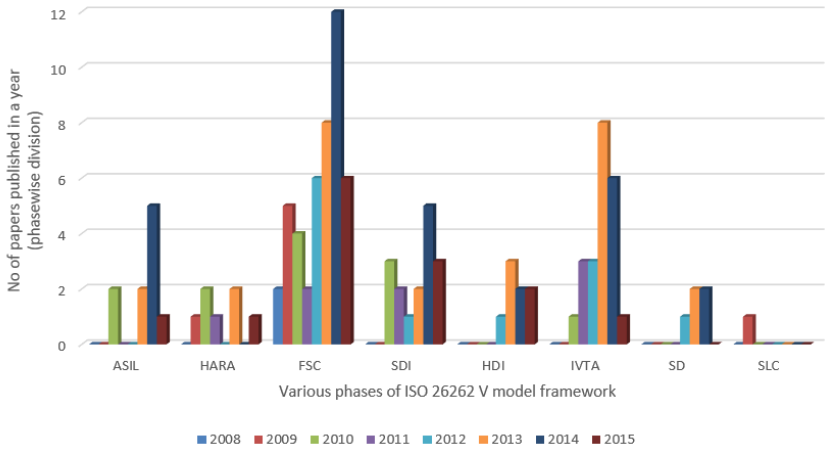

Fig. 2. Trend of publications related to the ISO 26262 over a period of time

Peer-reviewed articles on the topics "ISO 26262" and "vehicle safety", published between 2008 and 2015, are included. We exclude duplicate reports of the same or similar studies as well as white papers are excluded. After the search and inclusion/exclusion processes, we identify 120 unique papers. In our findings, we discover that higher number of papers are published in the concept phase (63 papers) than the development phases (51 papers) *i.e.,* product development, software development, and hardware development phases of the ISO 26262 V-model. The remaining six papers are considered as general publications, since they cover all the phases of the V-model. To further narrow down the search results, citations are used as a key tool to assess the quality of the identified papers. Publications between 2013 and 2015 are included.

In the case of concept and product development phases, more than half of the papers have been cited at least once and number of papers cited more than five are 18. Figure 2 shows the trend of papers published in each sub-phases from the selected sources. It can be inferred that the focus of the papers are more on the improvement of FSC (Functional Safety Concepts) in the conceptual phase and IVTA (Integration,Validation,Testing and Assessment) in the development phase. This shows the following observations:

Table 1. Mapping between safety related techniques and safety phases

PHASES	SUB PHASES	TECHNIQUES	IMPORTANT FEATURES	REF
CONCEPT PHASE	ASIL	SRGM (Safety Reliability Growth Model)	* Consider all risk factor apart from development method.	[10]
		HIP-HoPs	*First Automatic ASIL allocation tool. *Applicable for complex large scale systems. *Consumes more processing time. *Provides less options for ASIL decomposition.	[17]
		Systems of Linear Equations	*Easy to implement and consumes less processing time. *Not proven to be applicable for complex systems. *Provides all possible solutions based on Cost Optimization for ASIL decomposition.	[9]
		Exact Solver	*Provides all possible solutions based on Cost Optimization for ASIL decomposition.	[15]
PRODUCT DEVELOPMENT PHASE	Specification of Requirements	IBM Rhapsody	*Ensures requirement traceability throughout the product life cycle.	[3]
		Papyrus	*Ensures requirement traceability and also can perform safety analysis.	[14]
	Safety Validation	GSN (Goal Structuring Notation)	*Reduce cost and time during certification process. *It also helps to reuse the models. *Act as basis for systematic functional safety requirements.	[8]
		SmartTestGen	*Integrates different test generation techniques and covers maximum test cases to ensure safety.	[18]
		Time Usage Model (TUM)	*Describe time and data dependencies of the system to be tested.	[19]
HARDWARE DEVELOPMENT PHASE	Safety evaluation	UML complaint meta model	*Provides evaluation of preliminary hardware architecture. *It supports the design process as the complete hardware design including safety evaluation can be performed in one model based environment.	[7]
SOFTWARE DEVELOPMENT PHASE	Architecture and Implementation	Simulink Model Generator and Safety Driver Generator	*This tool enhancement is capable of generating Simulink models to support application software development. *Also for configuring and generating safety drivers for initialization, runtime testing and error handling.	[14]

– More additional standardized procedures have been implemented from the IEC 61508 standard on the conceptual and development phases where automobile manufacturers required clear process for implementation.
– Engineers and researchers were involved in the development of methodologies to ensure safety compliance of the system at these phases.

The summary of the selected papers mapped to the standard phases is presented in Table 1. Following section presents the gap analysis results between the ISO 26262 standard and the techniques identified from the selected papers.

4 Gap Analysis

A gap analysis helps to understand the shortcoming of existing approaches suggested by literatures. The gap analysis is carried out between the ISO 26262 objectives of the Part 3 till Part 7 sub-phases.

Table 2. Schematic illustration of the gap between objectives of the ISO 26262 *concept phase* and the respective techniques from the literature

ISO 26262		**Standard Objective**	**Techniques Developed**	**Gap Analysis**
Phase	**Sub phases**			
Concept Phase	*Item Definition*	Key objective is to support an adequate understanding of the item so that the activities in subsequent phases can be performed.	Need further analysis to understand the method used for item definition.	Need further analysis to understand the gap.
	HARA	To Identify and categorize the hazards and formulate the safety goals. Then derive the safety requirements from safety goals and allocate them to the architectural elements	Separate methods are available to identify hazards. Whereas different methods are used to allocate the safety goals to the architectural elements.	Need for enhanced tool that integrates the HARA analysis, derivation of safety requirements and safety goal allocation to the architectural elements.
	ASIL Allocation and Decomposition	Effective ASIL Allocation and Decomposition to reduce the complexity and the development cost of the design.	Various methods are developed that provides optimized possible combination of solutions for decomposition. Most of these methods takes more processing time.	Only few methods are suitable for complex large scale systems and provides all possible solutions for decomposition. Need effort to reduce the processing time.
	Functional Safety Concept	To derive the functional safety requirements and allocate them to the architectural elements of the item.	Enhanced architecture description language techniques are developed that helps for allocation and reduce ambiguity.	Need further analysis to understand the gap.

4.1 Gap Analysis for the Concept Phase

Table 2 summarizes the finding of a gap analysis for the *concept phase*. In the area of Hazard Analysis and Risk Assessment (HARA), various techniques are available to identify and categorize the hazards. Techniques suggested by the literature elucidate the way of estimating the hazard parameters (*i.e.*, severity, exposure and controllability) and help to formulate the safety goals. After identifying the safety goals, safety requirements can be derived for each goals. Literature provides more options for writing the requirement by different notations [10]. Once the requirements are elicited, they are allocated to the relevant architectural elements. This is performed using various architecture description languages such as EAST-ADL [17] and AADL [9].

Though existing techniques fulfill the objectives given for HARA in the standard, more techniques are required to achieve this effort. There is no standard common method or tool suggested by literature for meeting this objective. This is found to be one of the gaps by contrasting the standard objectives and literature approaches. A gap analysis for other sub-phases of the concept phase

Table 3. Schematic illustration of the gap between standard objectives and techniques in the area of *product development* phase

ISO 26262		**Standard Objective**	**Techniques Developed**	**Gap Analysis**
Phase	**Sub phases**			
Product Development Phase	*Technical Safety Requirement Specification*	To verify whether TSR comply with the FSR. To manage the system requirements with complete traceability across the product life cycle.	Fewer tools like IBM Rational Team Concert, PTC Integrity, Papyrus are developed for requirement specification to improve the traceability. But the detailed semantic traceability for each sub phase has not been explored.	New Opportunity that integrates the requirement specification with Item Integration, testing and validation.
	System Design	To develop the system design. To verify the system design and the technical safety concept comply with the TSR specification	As mentioned above, to verify the design and concepts compilation with the specification, fewer tools are developed that also ensures the traceability.	Need for enhanced tool that integrates both design and verification process together.
	Safety validation	To ensure all the safety cases generated in the concept phase are validated.	Separate tools like Medini Analysis are developed to ensure safety validation. All these tools Depends on the input of requirement management.	Need for tool that integrates the requirement management with the safety validation.
	Integration and Testing	To test compliance with each safety requirement and to verify the system design covering those requirements.	Methods are available that ensures the requirement traceability and verify the system design compliance.	Need for tool that combines all the sub phases of the product development.

i.e., Item Definition, Functional Safety Concept, and ASIL [13, 19, 20, 12, 18] is presented in the Table 2.

4.2 Gap Analysis for the Product Development Phase

From the gap analysis of the product development phase, it is observed that there are few tools [23, 21] suggested by literature and industrial technical report for requirement specification. These tools support only for specific sub-phases and there are more opportunities to integrate these tools with testing and validation tools [4, 17]. By this integration, it becomes more sophisticated to perform all the activities of a phase using single technique. This also gives clear way of understanding the standard norms to the developers and verifying it by testers using same platform. The finding of this gap analysis can be found in the Table 3 on the previous page.

Table 4. Schematic illustration of the gap between standard objectives and techniques in the area of *software development* phase

ISO 26262		**Standard Objective**	**Techniques Developed**	**Gap Analysis**
Phase	sub phases			
Software Development Phase	*Architecture*	To develop and verify the architectural design that realizes the software safety requirements.	Several methods such as GSN (Goal Structuring Notation) are used to reduce the developing Cost and time. This also helps for verification with the safety requirements.	Need tools for integrating the architectural design and verification the safety Requirements with the elements.
	Implementation	To specify and implement the software units identifies as specified in accordance with software design and the associated software safety requirements.	Need further analysis to understand the methods used for implementation process.	Need further analysis to understand the gap.
	Unit testing	To demonstrate the software units fulfil the software unit design specification and do not contain undesired functionality.	Need further analysis to understand the methods used for unit testing.	Need further analysis to understand the gap.
	Safety Requirement Verification	To demonstrate that the embedded software fulfils the software safety requirements	As mentioned in the previous phase, to verify the safety requirements with the software, fewer tools are developed that also ensures the traceability.	Need for enhanced tool that integrates both design and verification process together.

4.3 Gap Analysis for the Software Development Phase

Similar to the system architecture level, more techniques are used for the software level [11]. Some of the common architecture description languages are EAST-ADL [17] and AADL [9] which help to reduce the development cost and time. In addition, such techniques provide a way to make the verification of safety requirements easier. But there is no tool available that integrates both architectural design and safety verification together. This is found to be one of the gap. Table 4 on the previous page shows the gap analysis performed for the software development phase.

4.4 Gap Analysis for the Hardware Development Phase

In the case of hardware development phase, only few literatures are published about the development required for the evaluation of safety violation. These literatures provide techniques mainly to support two claims. One is hardware architectural metrics and second is evaluation of safety goal violations. Techniques like UML based meta-model [9] support for design process and help to

Table 5. Schematic illustration of the gap between standard objectives and techniques in the area of *hardware development* phase

ISO 26262		**Standard Objective**	**Techniques Developed**	**Gap Analysis**
Phase	Sub phases			
Hardware Development Phase	*Safety requirement*	To demonstrate that the hardware fulfils the hardware safety requirements	As mentioned in the product development phase, to verify the safety requirements with the components, fewer tools are developed that also ensures the traceability.	Need for enhanced tool that integrates both design and verification process together.
	Design	To develop and verify the architectural design that realizes the hardware safety requirements.	Need more analysis to understand the techniques used for hardware design and ensuring the safety requirements compliance.	Need further analysis to understand the gap.
	Safety evaluation	To demonstrate the compliance of the design with the safety metrics.	Need more analysis to understand the evaluation techniques used along with the hardware architectural metrics.	Need further analysis to understand the gap.
	Testing	To demonstrate the hardware components fulfil the hardware design specification and do not contain undesired functionality.	Need more analysis to understand the hardware testing procedures.	Need further analysis to understand the gap.

perform safety evaluation in a unified model based environment. The findings of the gap analysis for the hardware development phase are shown in the Table 5. Following section discusses the main results of the gap analysis.

5 Discussion

Based on the gap analysis, the shortcoming and challenges of the techniques suggested by literature while fulfilling the standard objectives are found. In the concept phase, gap analysis identified the lack of mature techniques that provide wider possible solutions for ASIL decomposition. It showcases the opportunity for integrating various techniques within the phase. For product development phase, gap analysis shows similar results. There are tools used for each sub phases of the product development but there is no common platform where all sub phase activities can be performed. This tool integration could facilitate the understanding and correct interpretation of the standard norms.

For the software and hardware development phase, same type of architecture description languages, such as EAST-ADL and AADL, are used. But there is a lack of common platform that supports both design and safety evaluations.

6 Conclusion and Future Work

Since the ISO 26262 standard does not specify which techniques to be applied in fulfilling the safety requirements, variety of techniques are developed for each phase of the ISO 26262 standard. However, a general overview of existing and emerging ISO 26262 related techniques is lacking. Therefore, in this paper, we carried out a gap analysis to identify the challenges and future trends to fulfill the ISO 26262 (part 3 to Part 7) safety objectives. We identified that the focus of research techniques is for the concept and product development phases. However, more techniques are needed for fulfilling the objectives of the software and hardware phases.

As a future work, we plan to conduct similar study on the remaining phases of the ISO 26262 and develop a method for the software and hardware development phases. Furthermore, our analysis focused on the research results rather than the practical application of the standard. This requires further survey on the gap between research results and the practical applicability of the standard to reflect the actual situation in the automotive industry.

References

1. ACM. `http://dl.acm.org/`. Accessed: 2015-04-24.
2. AutoFOCUS3. `http://af3.fortiss.org/research/`. Accessed: 2015-09-16.
3. FISITA. `http://www.fisita.com/publications/papers`. Accessed: 2015-04-24.
4. IBM Rhapsody. `http://www-03.ibm.com/software/products/en/ratidoor`. Accessed: 2015-06-02.

5. IEEE. `http://ieeexplore.ieee.org/`. Accessed: 2015-04-27.
6. OPENCOSS. `http://www.opencoss-project.eu/`. Accessed: 2015-09-16.
7. SAE. `http://digitallibrary.sae.org/`. Accessed: 2015-04-24.
8. Springer. `http://link.springer.com/`. Accessed: 2015-04-27.
9. N. Adler, S. Otten, P. Cuenot, and K. Müller-Glaser. Performing safety evaluation on detailed hardware level according to ISO 26262. *SAE International journal of passenger cars-electronic and electrical systems*, 6(2013-01-0182):102–113, 2013.
10. D. Chen, R. Johansson, H. Lönn, Y. Papadopoulos, A. Sandberg, F. Törner, and M. Törngren. Modelling support for design of safety-critical automotive embedded systems. In *Computer Safety, Reliability,& Security*, pages 72–85. Springer, 2008.
11. Y. Dajsuren, M. G. van den Brand, A. Serebrenik, and R. Huisman. Automotive ADLs: A study on enforcing consistency through multiple architectural levels. In *ACM SIGSOFT Conference on Quality of Software Architectures (QoSA)*, pages 71–80. ACM, 2012.
12. M. S. Dhouibi, J.-M. Perquis, L. Saintis, and M. Barreau. Automatic Decomposition and Allocation of Safety Integrity Level Using System of Linear Equations. *Complex Syst*, pages 1–5, 2014.
13. T. Fujiwara, J. M. Estevez, Y. Satoh, and S. Yamada. A Calculation Method for Software Safety Integrity Level. In *Proceedings of the 1st Workshop on Critical Automotive applications: Robustness & Safety*, pages 31–34. ACM, 2010.
14. IEC. Functional Safety of Electrical/electronic /programmable Electronic Safety-related Systems. IEC 26262, International Electrotechnical Commission, 2009.
15. International Standardization Organization. ISO 26262: Road Vehicles - Functional safety, International Organization for Standardization. 2011.
16. B. Kitchenham. Procedures for Performing Systematic Reviews. *Keele, UK, Keele University*, 33(2004):1–26, 2004.
17. R. Mader, G. Griessnig, E. Armengaud, A. Leitner, C. Kreiner, Q. Bourrouilh, C. Steger, and R. Weiss. A bridge from system to software development for safety-critical automotive embedded systems. In *Software Engineering and Advanced Applications (SEAA)*, pages 75–79. IEEE, 2012.
18. A. Murashkin, L. S. Azevedo, J. Guo, E. Zulkoski, J. H. Liang, K. Czarnecki, and D. Parker. Automated Decomposition and Allocation of Automotive Safety Integrity Levels Using Exact Solvers. *SAE International Journal of Passenger Cars-Electronic and Electrical Systems*, 8(2015-01-0156):70–78, 2015.
19. Y. Papadopoulos, M. Walker, M.-O. Reiser, M. Weber, D. Chen, M. Törngren, D. Servat, A. Abele, F. Stappert, H. Lonn, et al. Automatic Allocation of Safety Integrity Levels. In *Proceedings of the 1st workshop on critical automotive applications: robustness & safety*, pages 7–10. ACM, 2010.
20. D. Parker, M. Walker, L. S. Azevedo, Y. Papadopoulos, and R. E. Araújo. Automatic Decomposition and Allocation of Safety Integrity Levels Using a Penalty-Based Genetic Algorithm. pages 449–459, 2013.
21. P. Peranandam, S. Raviram, M. Satpathy, A. Yeolekar, A. Gadkari, and S. Ramesh. An integrated test generation tool for enhanced coverage of Simulink/Stateflow models. In *Design,Automation & Test in Europe Conference & Exhibition (DATE)*, pages 308–311. IEEE, 2012.
22. A. Saberi, Y. Luo, F. Cichosz, M. van den Brand, and S. Janseny. An Approach for Functional Safety Improvement of an Existing Automotive System. In *8th Annual IEEE System Conference*, pages 277–282, 2015.
23. S. Siegl, K.-S. Hielscher, R. German, and C. Berger. Formal specification and systematic model-driven testing of embedded automotive systems. In *Design, Automation & Test in Europe Conference & Exhibition*, pages 1–6. IEEE, 2011.