文章编号：0253-374X(2022)S1-0247-05

DOI：10. 11908/j. issn. 0253-374x. 22-SSS1-23751

基于ISO 26262安全要求的汽车电子电气架构优化模型

Dorsa ZAHERI，Hans-Christian REUSS

（斯图加特大学 汽车工程学院(IFS)，斯图加特 70569，德国）

摘要：在开发汽车电子电气(E/E)架构时，考虑安全要求是实现未来技术(如自动驾驶)的先决条件之一。按照ISO 26262标准，安全分析必须在产品开发生命周期的早期阶段进行，以检测设计缺陷并采取行动改善设计。本文提出了一种基于模型的方法，在汽车E/E架构的设计阶段解决符合ISO 26262的安全要求；同时，基于这些要求，提取了一组与安全相关的约束条件，并通过整数线性规划(ILP)模型将这些约束用于E/E架构的优化。

关键词：汽车电子电气；电子电气架构；优化模型；安全要求；ISO 26262标准

中图分类号：U462　　**文献标志码**：A

Considering Safety Requirements Based on ISO 26262 in Model-Based Optimization of Automotive Electrical/Electronic Architectures

Dorsa ZAHERI, Hans-Christian REUSS

(Institute of Automotive Engineering (IFS), University of Stuttgart, 70569 Stuttgart, Germany)

Abstract: Considering safety requirements while developing electrical and electronic (E/E) architectures is a prerequisite for the realization of future technologies such as autonomous driving. Following the ISO 26262 standard, safety analyses have to be conducted in the early phase of the development lifecycle in order to detect design flaws and take actions to improve the design. This paper presents a model-based approach for addressing safety requirements conforming to ISO 26262 during the design phase of automotive E/E architectures. Based on the requirements, a set of safety-related constraints is extracted, which can be used in an integer linear programming (ILP) model to optimize E/E architectures.

Key words: functional safety; E/E architecture; optimization; model-based development

The number of functions and complexity in E/E architectures are increasing due to the transition to Advanced Driver Assistance Systems (ADAS) and autonomous vehicles. Future vehicles are expected to have a centralized architecture in which several high-performance general-purpose Electronic Control Units (ECU) control multiple functions[1]. The new requirements rising out of these technological innovations lead to an increase in the design complexity of automotive E/E architectures. Safety is one of the key requirements that must be considered during the design phase of future vehicles. Currently, model-based development approaches are drawing the attention of car manufacturers and suppliers as a solution to master design complexity. Therefore, integrating safety concepts into model-based E/E architecture design plays a crucial role to overcome the aforementioned challenges[2-3].

ISO 26262 "Road vehicles—Functional safety" is an adaption of the functional safety standard IEC 61508 for the automotive domain[4]. The safety life cycle according to ISO 26262 influences all phases of vehicle development. Until now, ensuring ISO 26262 compliance is a time-consuming process that is mostly done manually. In this paper, we present a generic framework for model-based optimization of automotive E/E architectures based on safety

收稿日期：2022-08-15

第一作者：Dorsa ZAHERI (1992—)，女，理学硕士，主要研究方向为汽车机电一体化。E-mail：dorsa. mohammad-zaheri@ifs. uni-stuttgart. de

通信作者：Hans-Christian REUSS (1959—)，男，教授，工学博士，主要研究方向为汽车机电控制。

E-mail：hans-christian. reuss@ifs. uni-stuttgart. de

constraints. There are several studies that present model-based approaches to optimize E/E architectures concerning various attributes, including cost, weight, and power consumption[5-7]. However, only a few studies take reliability and safety aspects into account. In **Ref.** [8], the authors present an approach for optimizing E/E architectures based on reliability. The authors in **Ref.** [9-10] took a step further and considered both automotive safety integrity level (ASIL) and reliability requirements. In **Ref.** [11-12] ILP formulations are presented to optimize architecture topology and resource allocation in a central computing platform, respectively. Although these studies focused on safety attributes, there are still other safety-related requirements, such as timing requirements, that have yet to be considered. Therefore, we aim to extract constraints from safety requirements in accordance with ISO 26262. These constraints can be added to an ILP optimization model in order to generate a safe architecture. This may bring us one step closer to automating E/E architecture design.

Fig. 1 illustrates our approach to integrating ISO 26262 safety requirements into the development process of E/E architectures. The development process follows the well-known V-model. In this paper, we are focusing only on the design process, which means the left branch of the V-model.

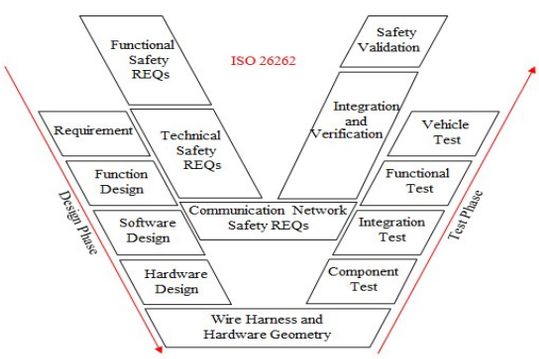

Fig. 1 Integrating ISO 26262 safety requirements into the V-model

At the beginning of the development process, functional and non-functional requirements are documented. The next level, i. e., the function design level, deals with the functions of the vehicle and their interactions. As shown in **Fig.** 1, functional safety requirements according to ISO 26262 are identified during these phases. Based on the hazard analysis and risk assessment, required ASIL levels for each function or a set of functions are determined. Another important aspect in the development of current E/E architectures is timing. Many functions have certain timing constraints. Currently, most automotive functions are distributed functions. At the function design level, timing requirements for function chains, which refer to end-to-end timing constraints, are identified. These are safety-related requirements that must be fulfilled[13].

At the next levels, software architecture and hardware topology are designed, respectively. The allocation of functions/software components on hardware nodes is one of the factors that influences the quality of the designed system. According to ISO 26262, safety requirements should be assured when mapping software to hardware components. Consequently, the deployment process is getting even harder when considering the conflicting constraints and the growth in the complexity of architectures[7]. Therefore, using an optimization algorithm can be helpful to solve this issue and automate this task. An exemplary mapping of the functional network model to a component network model is depicted in **Fig.** 2.

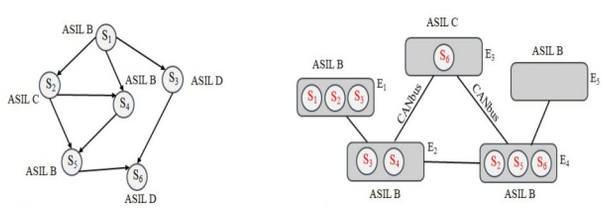

Fig. 1 Exemplary function architecture model; hardware architecture model and a deployment candidate

1 System modeling

This section introduces the parameters of our function and hardware model. These parameters are the input of the optimization algorithm and are summarized in **Tab.** 1. Our model is inspired by the approach in **Ref.** [5].

1.1 Hardware component specifications

The hardware model consists of a few general-purpose ECUs and communication buses. Although the real architectures contain sensors and actuators, we are not defining them in our model. This is because the mapping of functions to sensors and actuators is not meaningful. However, their effect on the deployment process is considered as localization constraint, which is explained in the next section. As can be seen in Table 1, we denote a set of ECUs as $E=\{E_1,\cdots,E_n\}$. The ASIL level that E_k supports is considered as A_k. The number of ECU cores and the size of its memory, expressed in MB (megabyte), are defined as C_{Ek} and M_{Ek}, respectively. Our model supports two communication buses, CAN and Ethernet. The data transmission rates of the buses are defined as dr_{CAN} and dr_{ETH}. Since the failure rates of the buses are lower than those of the ECUs, we refrain from considering their values in reliability analysis.

1.2 Software components specifications

The function model consists of some unmodifiable software components with defined specifications and the connections between them. As shown in **Tab**. 1, we denote a set of software components as $S=\{S_1,\cdots,S_m\}$. The amount of data being transferred from function S_i to S_j is considered as ds (S_i, S_j). As already mentioned, for each software component S_i an ASIL level L_i is assigned during software design phase. We defined the Worst-Case Execution Time (WCET) of S_i running on E_k and ASIL L_h as WCET (i, k, h). Some software components may be executed periodically. Hence, we considered the variable F_{ij} to express the frequency of S_i with regard to S_j.

Tab.1 Hardware and software architecture parameters

Hardware components	
$E=\{E_1,\cdots,E_n\}$	Set of all ECUs
C_{Ek}	Number of CPU cores of E_k
M_{Ek}	Memory of E_k
λ_k	Failure rate of E_k
A_k	ASIL level of E_k
dr_{CAN}, dr_{ETH}	Data transmission rate of CAN and Ethernet
Software components	
$S=\{S_1,\cdots,S_m\}$	Set of all Software components
C_{Si}	Required CPU cores of S_i
M_{Si}	Required memory of S_i
L_i	ASIL level of S_i
WCET(i,k,h)	WCET of S_i on E_k and ASIL L_h
F_{ij}	Frequency of S_i

2 Extraction of safety constraints

This section explains the formulated requirements, which can be used as constraints for an ILP optimization problem. According to ISO 26262, four ASIL levels, from ASIL A to D, are defined to represent the stringency of safety requirements. ASIL A represents the least and ASIL D dictates the most stringent requirement. In the future, most of the software components are expected to be safety-critical, which means ASIL C or D. However, there are a few available ECUs that can support high ASIL levels. Therefore, action must be taken to enable the mapping of such software functions to these ECUs, while verifying ASIL compatibility. ISO 26262 introduces ASIL decomposition technique to reduce the required ASIL level of a software component by dividing it into multiple redundant components, each with a lower ASIL value[4]. In order to verify ASIL Compatibility, following constraint is defined:

$$\forall E_k\in E,\ \forall S_i\in S:\quad Y_{ik}\cdot L_i\leqslant A_k \tag{1}$$

In the above constraint, L_i is an integer value between 1 and 4, which $L_i=1$ represents ASIL A and $L_i=4$ represents ASIL D. We assume $Y_{ik}=1$, if S_i is mapped to E_k. If the above constraint is not satisfied, an ASIL decomposition, similar to the approach in **Ref**. [9], should be performed. In this case, another constraint should be defined to prevent the execution of redundant software components on the same ECU. Another requirement that should be satisfied is timing constraint. When designing a software architecture, an end-to-end timing requirement can be defined for a software component chain. Constraint (2) ensures that the end-to-end timing requirement T for the function chain $S'\subseteq S$ is fulfilled.

$$\forall E_k\in E,\ \forall S_i\in S':\quad \sum \mathrm{Max}\left(Y_{ik}\cdot \mathrm{WCET}(i,k,h)\right)+t_{com}\leqslant T \tag{2}$$

Due to the redundancy caused by ASIL decomposition, there might be different paths for the defined functional chain. Therefore, we consider the maximum reaction time for a functional chain to guarantee the fulfillment of the timing requirement. In the above constraint, t_{com} refers to the data transfer time between corresponding ECUs and can be estimated using equations (3) and (4). We assume that the communication buses between ECUs are identified in the hardware architecture. We store the ECUs that only have a CAN interface in E'.

$$\forall E_k \in E,\ \forall (S_i, S_j) \in S' \mid Y_{ik} = Y_{jk} = 1: \quad t_{com} = 0 \tag{3}$$

$$\forall E_k \in E',\ \forall (S_i, S_j) \in S': \quad t_{com} = \sum_{(S_i, S_j) \in S'} [(Y_{ik} + Y_{jk}) \cdot \frac{\mathrm{ds}(S_i, S_j)}{\mathrm{dr}_{CAN}} + (1 - (Y_{ik} + Y_{jk})) \cdot \frac{\mathrm{ds}(S_i, S_j)}{\mathrm{dr}_{ETH}}] \tag{4}$$

The reliability of S_i running on E_k with L_h is [9]:

$$R(S_i, E_k, L_h) = \mathrm{e}^{-\lambda_k \mathrm{WCET}(i,k,h)} \tag{5}$$

Equation (6) can be used to ensure the fulfillment of reliability constraint for the whole system. This equation is valid only for systems without redundancy. If performing ASIL decomposition and consequently adding redundant components is needed, then the approach introduced by **Ref**. [9] can be used to calculate reliability of the system.

$$\prod_{S_i \in S:\ Y_{ik}=1} R(S_i, E_k, L_h) \geqslant R_{\mathrm{req}} \tag{6}$$

In order to verify that ECUs provide sufficient CPU cores and memory for software components which run on them, constraints (7) and (8) can be used[5].

$$\forall E_k \in E: \quad \sum_{S_i \in S} Y_{ik} C_{S_i} \leqslant C_{E_k} \tag{7}$$

$$\sum_{S_i \in S} Y_{ik} M_{S_i} \leqslant M_{E_k} \tag{8}$$

A localization constraint, equation (9), can be defined to prevent deploying a software component on a particular ECU. For example, $\mathrm{loc}(S_2) = E_1$ means that S_2 should not be executed on E_1.

$$\forall E_k \in E,\ \forall S_i \in S \mid E_k \in \mathrm{loc}(S_i): \quad Y_{ik} = 0 \tag{9}$$

Constraint (10) ensures that an ECU has sufficient CPU capacity to execute its tasks. In order to satisfy CPU utilization constraint, it must be ensured that the CPU utilization doesn't exceed its threshold value (U_t).

$$\forall E_k \in E: \quad \frac{1}{C_{E_k}} \sum_{S_i \in S} Y_{ik} \cdot \mathrm{WCET}(i,k,h) \cdot F_{ij} \leqslant U_t \tag{10}$$

In addition to the above constraints, an objective goal such as minimizing cost can be added to an optimization problem. Minimizing cost can be formulated as follows:

$$\min \sum_{S_i \in S} \mathrm{cost}(S_i, L_h)$$

Our proposed workflow is depicted in **Fig**. 3. The output of the optimization algorithm is a cost-effective safe candidate for the deployment problem.

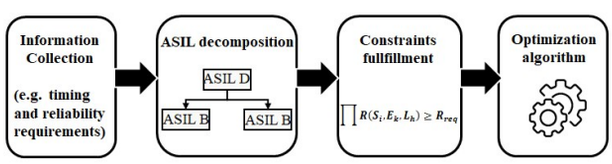

Fig.3 Workflow of the proposed approach

3 Conclusion and future work

In this paper, a model-based approach for function mapping and E/E architecture optimization based on safety constraints for future centralized architectures is presented. We derived safety requirements that are not application-specific from ISO 26262 and formulated them as ILP-based constraints. By adding these constraints and an optimization objective such as cost to an optimization algorithm, achieving a cost-effective safe architecture is possible. We consider the presented work as a first step and we are aware that our approach is far from complete. In future, we will apply this model to AMPL tool and a solver (CPLEX or Gurobi), to find the best solution for our ILP optimization problem.

References:

[1] ASKARIPOOR H, FARZANEH M, KNOLL A. E/E architecture synthesis: Challenges and technologies [J]. Electronics Journal, 2022.

[2] STARON M. Automotive Software Architecture Views and Why we need a new one—Safety view[C]//Workshop CARS-

critical automotive applications.[S.l.]:[s.n.], 2016.

[3] HAMMER M, MASCHOTTA R, ZIMMERMANN A. Model-driven application development for evaluation and optimization of automotive E/E-architectures [C]//IEEE International Conference on Recent Advances in Systems Science and Engineering (RASSE). [S.l.]: IEEE, 2021.

[4] International Organization for Standardization. Road Vehicles—Functional safety:ISO 26262[S].2011.

[5] KAMPMANN A, LUEER M, KOWALEWSKI S, et al. Optimization-based resource allocation for an automotive service-oriented software architecture [C]//IEEE Intelligent Vehicles Symposium (IV22), 2022.

[6] KUGELE S, PUCEA G. Model-based optimization of automotive E/E-architectures [C]//Proceedings of the 6th International Workshop on Constraints in Software Testing, Verification, and Analysis. [S.l.]:[s.n.], 2014.

[7] A. Aleti A, L. Grunske L, I. Meedeniya I, et al. Let the ants deploy your software—An ACO based deployment optimisation strategy [C]//IEEE/ACM International Conference on Automated Software Engineering, 2009.

[8] MEEDENIYA I, BUHNOVA B, ALETI A. Reliability driven deployment optimization for embedded systems[J]. Journal of Systems and Software, 2011, 84(5): 835.

[9] XIE G, CHEN Y, LIU Y, ET AL. Minimizing development cost with reliability goal for automotive functional safety during design phase[J]. IEEE Transactions on reliability, 2018, 67(1):196.

[10] XIE G, WU W, ZENG G, et al. Risk assessment and development cost optimization in software defined vehicles[J]. IEEE Transactions on Intelligent Transportation Systems, 2021.

[11] ASKARIPOOR H, FARZANEH M, KNOLL A. A model-based approach to facilitate design of homogeneous redundant E/E architectures [C]//IEEE Intelligent Transportation Systems Conference (ITSC). [S.l.]: IEEE, 2021.

[12] PAN F, LIN J, RICKERT M, et al. Resource allocation in software-defined vehicles: ILP model formulation and solver evaluation[C]//IEEE Conference on Intelligent Transportation Systems (ITSC), [S.l.]: IEEE, 2022.

[13] AUTOSAR. Recommended methods and practices for timing analysis and design within the AUTOSAR development process[M]. 2018. http://www.autosar.org.