SAE International®

ASIL Decomposition: The Good, the Bad, and the Ugly

2013-01-0195
Published
04/08/2013

Joseph G. D'Ambrosio and Rami Debouk
General Motors Company

doi:10.4271/2013-01-0195

ABSTRACT

ASIL decomposition is a method described in the ISO 26262 standard for the assignment of ASILs to redundant requirements. Although ASIL decomposition appears to have similar intent to the hardware fault tolerance concept of IEC 61508-2, ASIL decomposition is not intended to reduce ASIL assignments to hardware elements for random hardware failures, but instead focuses on functions and requirements in the context of systematic failures. Based on our participation in the development of the standard, the method has been applied in different ways in practice, not all of which are fully consistent with the intent of the standard. Two potential reasons that may result in the use of “modified” ASIL algebra include the need of OEMs to partition a system and specify subsystem requirements to suppliers and the need for designers to construct systems bottom up. Constructing systems bottom up has the goal of achieving a target system level ASIL from component elements that have some notion of ASIL already associated with them. In this paper, we examine the origins of ASIL decomposition in the ISO 26262 standard, potential benefits and limitations of the approach, and by examining publications on this subject, how it is currently being applied in industry programs.

INTRODUCTION

The task of decomposing and allocating requirements is a well-known part of systems engineering. A system-level requirement is either directly allocated to one component of the system, or it may be decomposed (split-up) into sub requirements, and these sub requirements allocated to different components. In this latter case, each of the sub requirements supports partial achievement of the system requirement, with the expectation that when each component achieves it sub requirement, the system-level requirement will also be satisfied.

When developing system-level and subsystem level requirements, it is in general desirable to avoid creating redundant requirements, given that the duplication of a requirement can create maintenance problems from a requirements management perspective. If one of the redundant requirements is modified, then all others must be or the requirements will be inconsistent. However, when developing high-integrity systems, redundant requirements can improve the integrity of the system. In this case, a system-level requirement is decomposed into redundant sub requirements, where each of the individual sub requirements directly supports achieving the system requirement. If the redundant requirements are allocated to different components, then if one of the components fails to satisfy its requirement, the other component may still do so, thus improving system integrity.

One question that arises is how to assess the level of integrity provided by redundant requirements. From a safety or assurance case perspective, a system that has multiple ways of achieving a requirement would seem to increase the likelihood that the requirement will be satisfied, than a system that provides only one means of achieving the requirement, and thus there is a desire by the safety analyst to assign a higher integrity level to a system with redundant requirements. The notion of common cause failures needs to be considered, and for redundant requirements to provide higher integrity than a single requirement, there has to be some means to assure independence in the implementation of the redundant requirements.

To assess the value of redundant requirements, a fault model must be established. In general faults could be put into two categories: random hardware failures and systematic failures, where systematic failures may be related to both the hardware and software of a system. To assess the level of integrity of a design with redundant requirements against random hardware failures, a fault tree analysis can be performed. The redundantly implemented requirements will lead to multiple input AND gates in the fault tree. However, for systematic failures, a qualitative algebraic equation, which provides rules for relating the overall integrity achieved to the integrity of the redundant requirements, may be defined and applied. For such an approach, a set of integrity levels can be defined (e.g., A, B, C, & D), and the algebra rules relate how the integrity of redundant requirements are combined to produce higher integrity levels (e.g., A + A = B).

In this paper, we examine how the above issues are addressed by the ISO 26262 standard [1]. ISO 26262 defines Automotive Safety Integrity Levels (ASILs) and an algebraic approach for decomposing safety requirements (ASIL decomposition). The paper is motivated by the author's observation that the practice of ASIL decomposition is not currently consistent within the automotive industry. To help address this issue, we survey and comment on several publications on this subject. The paper is organized as follows: first we provide an introduction to ISO 26262 ASIL decomposition, next we review several related publications, identifying the approaches described and providing comments and a summary example, finally we provide a summary of our findings.

ASIL DECOMPOSITION IN ISO 26262

ASIL decomposition is the method defined in Clause 5 (requirements decomposition with respect to ASIL tailoring) of Part 9 of ISO 26262 to decompose a given safety requirement into a set of redundant safety requirements to which an ASIL is tailored given the ASIL of the “parent” requirement. In general, the ASIL is assigned to a safety goal and all safety requirements derived from that goal inherit the same ASIL. If requirements are assigned to elements (architectural, hardware or software) that are sufficiently independent, then the ASIL decomposition method can be used to implement the decomposed safety requirements redundantly on these independent elements and that may result in lowering the ASIL of the decomposed requirements. Note here that lowering the ASIL of the decomposed requirements does not imply that the ASIL of the “parent” requirement has been reduced, rather all the decomposed requirements with the lowered ASIL together achieve the ASIL of the “parent” requirement.

While using ASIL decomposition, one needs to be aware of the following:

- ASIL decomposition can be used for functional, technical, hardware, and software requirements

- If the property of independence of elements does not hold, then the decomposed requirements inherit the ASIL of their “parent” requirement.

- Dependent failure analysis [9] is required to prove independence if homogenous redundancy (pure duplication of hardware/software) is used.

ASIL decomposition can be applied to any requirement at any stage of the design process and it can be applied more than once. This is not the same as the hardware fault tolerance concept of IEC 61508-2 where the decomposition is allowed only once. Development of the elements assigned the decomposed requirements at the system, hardware and software levels is performed at a minimum at the decomposed ASIL. However, the evaluation of the hardware architectural metrics for the parent safety goal due to random hardware failures (ISO 26262, Part 5) needs to be performed at the ASIL before decomposition. Consequently, each requirement with a decomposed ASIL shall have the ASIL of its safety goal in between parentheses as a way to track the original ASIL. If ASIL decomposition is performed at the software level, elements implementing the decomposed requirements will need to be checked for independence (including the measures to guarantee independence) at system, hardware and software levels.

The rules for ASIL decomposition are described in Table 1. When performing the ASIL decomposition:

- Any confirmation measures (ISO 26262, Part2) will be applied given the ASIL before decomposition

- Proof of element independence (ISO 26262, Part 9) will be provided

- Any integration activities (ISO 26262, Part 4) will be performed given the ASIL before decomposition

We note that if the ASIL decomposition results in assigning decomposed requirements to the functionality and its safety mechanism then the safety mechanism will be assigned the highest of the decomposed ASILs. If the decomposition rule is one of ASIL X to ASIL X(X) and QM(X) then it is evident that quality management requirements are enough to develop the element to which the functionality is allocated.

We conclude this section by providing an example of ASIL Decomposition abstracted out of the example in [8]. Clause 11.3 of [8] describes a comfort feature whereby the driver can actuate it by pressing a switch while at zero speed. If the vehicle speed is above 15 km/h, the actuation may result in a hazardous situation. As is the case in [8], the feature, its requirements and their ASILs are for illustration purposes and do not reflect any real life feature and its requirements. Let us assume that the requirement before decomposition is:

R. The feature shall be deactivated for vehicle speed greater than 15km/h

Table 1. ASIL Decomposition Rules (Source ISO 26262, Part 9)

ASIL before Decomposition	ASIL after Decomposition
ASIL D Requirement	ASIL C(D) Requirement + ASIL A(D) Requirement Or ASIL B(D) Requirement + ASIL B(D) Requirement Or ASIL D(D) Requirement + QM(D) Requirement
ASIL C Requirement	ASIL B(C) Requirement + ASIL A(C) Requirement Or ASIL C(C) Requirement + QM(C) Requirement
ASIL B Requirement	ASIL A(B) Requirement + ASIL A(B) Requirement Or ASIL B(B) Requirement + QM(B) Requirement
ASIL A Requirement	ASIL A(A) Requirement + QM(A) Requirement

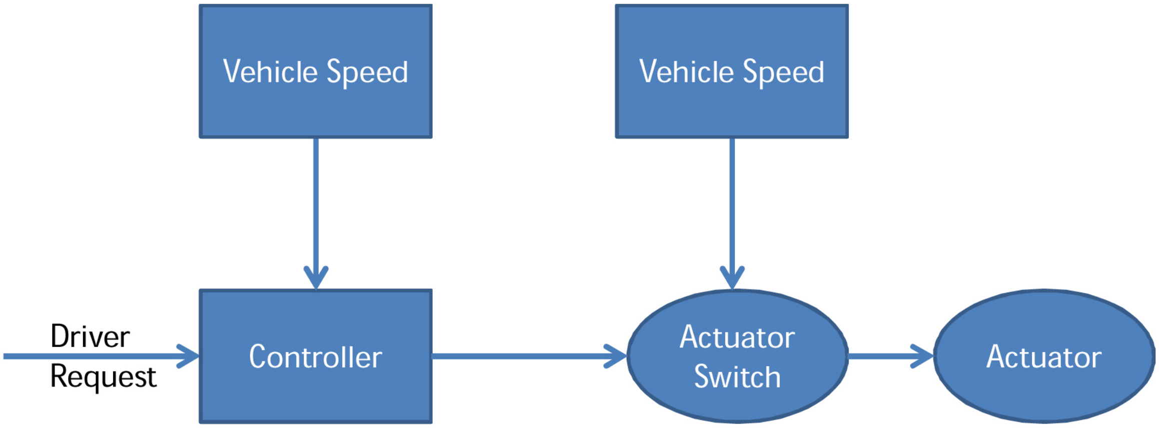

Figure 1. Feature Architecture

And the above requirement is an ASIL C requirement. Given the architecture described below in Figure 1, one can decompose the above requirement into two redundant requirements as follows:

R1. Controller shall not send an activation command for vehicle speeds greater than 15km/h

R2. Actuator switch shall open when vehicle speed is greater than 15km/h

Obviously the controller and switch are independent elements receiving the vehicle speed from independent sources. This proof of element independence allows the application of ASIL Decomposition to Requirements R1 and R2 and they can be assigned decomposed ASILs from Table 2 as follows:

Table 2. ASIL Decomposition for Requirement R

Requirement R1	Requirement R2
ASIL C (C)	QM (C)
ASIL B (C)	ASIL A (C)
ASIL A (C)	ASIL B (C)
QM (C)	ASIL C (C)

ASIL DECOMPOSITION INTERPRETATION AND PRACTICE

ASIL Decomposition has been studied, applied and reported out by many researchers and practitioners since the ISO 26262 Draft International Standard was released in 2009. In addition to the papers and presentations published and/or presented at conferences and workshops, many companies that provide consultancy in the functional safety arena have provided their own interpretation of ASIL Decomposition. We have reviewed a few of these papers and presentations and the following summarizes some of the main findings.

ASIL Decomposition Concept and Applicability

The concept is understood as not being a new one as IEC 61508 [2] Part 2 introduced a similar concept targeting hardware redundancy and that is recognized by all functional safety experts. However, the way the concept has been applied differs from one practitioner to another given their interpretation of the concept. Many have understood the concept as one to be used in allowing the exploration of options in designing an architecture.

In [3], the authors clearly state that ASIL Decomposition is a qualitative concept addressing systematic issues and not hardware failures. Namely, [3] states that the concept is used as part of the architecture design when decomposing requirements and allocating them to architecture elements and does not deal with hardware reliability and trying to achieve a quantitative target. Such statements are supported by [7] where it is clearly stated that ASIL Decomposition deals with systematic requirements and that hardware targets remain at the level of the top level safety requirement (the safety goal in ISO 26262 terminology).

At first, a more general statement on ASIL Decomposition is presented in [6] where the authors make the claim that ASIL decomposition can be performed on the system, hardware and software levels. In case the system, hardware and software levels are used to denote the architectural elements then that statement is in violation of the intent of ISO 26262 ASIL Decomposition. However, it seems that [6] is indirectly following the interpretation of [3] and [7] for ASIL Decomposition since a caution is made that the hardware architectural metrics shall not be affected by ASIL Decomposition. If architectural metrics are not affected then no hardware reliability issues due to random hardware failures are targeted.

It is worth noting here that [3] makes the statement that an ASIL can be lowered. Although the ASIL of the decomposed requirement may be lowered, the ASIL of the safety goal remains the same and hence no absolute lowering of the ASIL is achieved. On the other hand, if our understanding is incorrect, that statement may be inconsistent with the ASIL Decomposition rules and requirements.

Overlapping Requirements and ASIL Decomposition

Many functional safety experts agree that ASIL Decomposition is applied to requirements and not the elements building up the architecture. Even though [3] states that ASIL Decomposition introduces functional or heterogeneous redundancy through architectural design elements, we believe the authors of [3] are discussing the redundancy of the requirements allocated to different architectural elements. A similar statement is also made in [6] and [7]. In [6] and [7], the authors discuss that a suitable partitioning of safety requirements into redundant safety requirements can be used on a combination of independent elements. The key here is that the elements need to be independent and that requires some type of analyses to prove that a failure of one of the elements does not lead in isolation to the violation of the safety goal. Moreover, the decomposed requirements must comply with the “parent” requirement, that is, these requirements are all one and the same.

An interesting approach to decomposing requirements is the one presented in [4] where a fault tree analysis (FTA) like approach is used. In decomposing the requirements, independence of systems/subsystems is demonstrated through AND gates and the rules of Table 1 are applied. Even though [4] presents a fault tree representation of the systems and their building subsystems and depicts the corresponding ASIL Decomposition, our understanding is that the decomposed ASILs are assigned to the requirements of these systems and not the systems and subsystems themselves. If that understanding is incorrect, the proposed approach may be questionable and is in fact in violation of the intent of the ISO 26262 ASIL Decomposition. If the intention of the proposed approach is to benefit from requirements decomposition then such an approach may be valid.

Benefits of Requirements Decomposition

As ASIL Decomposition deals with requirements decomposition, such decomposition can be used to lower the ASIL of the safety requirements assigned to a specific element, by benefiting from the existing redundant elements in an architecture, which have already been added to help meet the hardware architectural metrics. Specifically, in many systems implementing safety mechanisms for random hardware failures (developed in accordance with ISO 26262 in one element) will be enough to meet the requirements of the standard for several elements of the design. Such a technique is one explanation for the decomposition rule of ASIL X to ASIL X(X) and QM(X) and has been discussed earlier.

ASIL Decomposition as a Top Down vs. Bottom Up Approach

ASIL Decomposition is easily interpreted as a top down approach. A safety goal is decomposed into requirements and sub-requirements. These requirements are assigned lowered ASILs in the case where redundant elements to which they are allocated to are independent and do not interfere with each other. The description in [4] of an FTA like ASIL Decomposition supports that argument whereby requirements are decomposed following the branches of the fault tree to different components. It is our understanding that independence of the components or architectural elements can be proven or demonstrated though the type of gates used: namely, the use of AND gates imply that the branches are independent and hence the components or architectural elements are indeed independent and that validates the correctness of the ASIL Decomposition. Another top down application of ASIL Decomposition is presented in [5] where the authors discuss an example of a brake by-wire system and present the safety requirements from the vehicle to system and subsystem levels along with the ASIL Decomposition of the requirements at the system and subsystems levels.

Some practitioners have considered the ASIL Decomposition as a bottom-up approach. In our understanding however, this is not the intent of ISO 26262 as demonstrated in Clause 5.4 of Part 9 of ISO 26262. That approach may be explained by the need of designers to construct systems bottom up, from existing design elements. More specifically, if a manufacturer provides subsystem safety requirements for a supplier, then the supplier needs to meet these requirements and hence achieve the target system level ASIL from component elements that have some notion of ASIL associated with them. Having some knowledge about the component elements and their associated ASILs and factoring that into meeting a target system level ASIL truly constitutes a bottom up approach for ASIL Decomposition. The authors in [6] discuss such an approach when designing basic software. As explained in [6], the basic software needs to meet an ASIL D and since it is so complex, it cannot be developed to meet that ASIL independent of other mechanisms. Consequently, some application software and a safety layer are developed meeting ASIL D requirements and alongside a QM basic software each component is allocated to independent architectural elements to meet the top ASIL D requirement. Another demonstration of the bottom up approach is also presented in [3] although not in as obvious manner as presented in [6]. In [3], the authors discuss an automotive example whereby a safety goal of an ASIL D is defined. All components in the system are assigned requirements to meet ASIL D integrity to start with. However, analyses at the component level are performed to check whether any faults in these components can lead directly to the violation of the safety goal or not. If there is no violation of the safety goal, the ASIL of these component requirements are lowered given one of the rules of Table 1. Even though inheriting the ASIL of the safety goal is considered a top down approach, the true ASIL Decomposition as presented in [3] was achieved in a bottom up approach.

Safety Element out of Context (SEooC)

_Per ISO 26262 Part 10 [8], A SEooC is a safety-related element which is not developed for a specific system in the context of a particular vehicle. Therefore such an element needs to be defined and analyzed without a vehicle level safety goal (and its corresponding ASIL) in mind. In doing so, assumptions are made at the component level and requirements are developed that can meet a given safety integrity level. Once these elements need to be integrated within the vehicle, the assumptions made need to be checked for correctness and whether the vehicle safety goals and safety requirements can be indeed met given the development of the SEooC. Even though SEooC does not require or use an ASIL Decomposition, such an approach, in our opinion, can be seen as a bottom up application of an ASIL Decomposition concept.

THE GOOD, THE BAD, AND THE UGLY

In this section, we provide a summary example that highlights issues identified in this paper.

Consider two cases of design requirements based on the example presented earlier.

Case 1 (“The Good”) is as follows

- System
- Requirement: The feature shall be deactivated for vehicle speed greater than 15km/h. ASIL C
- Controller
- Requirement: Controller shall not send an activation command for vehicle speeds greater than 15km/h ASIL A(C)
- Actuator Switch
- Requirement: Actuator Switch shall open when vehicle speed is greater than 15km/n ASIL B(C)

Case 2 (“The Bad and The Ugly”) is

- Controller ASIL A
- Actuator Switch ASIL B

Case 1 follows ISO 26262 methodology for ASIL decomposition. The Controller development must satisfy ASIL A requirements, the Actuator Switch development must satisfy ASIL B, and a random hardware analysis for the overall system must satisfy ASIL C requirements with respect to the ASIL C safety goal “The feature shall be deactivated for vehicle speed greater than 15km/h." If the Controller development is assigned to one supplier, and the Actuator Switch to another, each supplier knows the appropriate development process requirements for their component, and both must work together with the system integrator to ensure the random hardware analysis requirements are satisfied for the top safety goal (ASIL C).

Case 2 does not follow ISO 26262 methodology. Although the development of each component will in general be done to the appropriate requirements (ASIL A for the Controller, ASIL B for the Actuator Switch), Case 2 implies that there are no random hardware targets for the Controller (ASIL A has no target), and only recommended (not required) target for the Actuator Switch. There is no guarantee that if these components satisfy these ASIL targets for random hardware failures, that the overall system will satisfy the ASIL C target for the overall safety goal. Analysis to assess random hardware targets depends on understanding which component failure modes are safe and which are not with respect to the overall safety goal of the system. Case 2 does provide information to make this distinction. There is also a possibility that an unexpected dependency between the Controller and Actuator Switch would be missed (analysis is not required for the Controller).

When working with suppliers, it may indeed be necessary to provide the supplier with specific guidance on how they should proceed with their design, such that when their component is integrated into the overall system, the system itself will achieve the required ASIL target for random hardware failure requirements. This can be achieved by using a systems engineering budgeting approach, assigning an ASIL to guide the development of the component (addressing systematic errors), and a specific target for random hardware failures, derived by allocating the top level failure target among the components of the system. ISO 26262 Part 4 requirement 7.4.4.4 addresses this approach [1].

When designing a system based upon existing design elements, it may indeed by very useful to use a notation similar to Case 2 when evaluating different design alternatives. This bottom up approach provides a simple method to consider how elements with different ASILs may be used to construct an overall system. However, it is recommended that in parallel a top-down analysis be considered to help confirm that the assumptions of the bottom up analysis are justified.

SUMMARY AND RECOMMENDATIONS

In this paper, we summarized the theory behind the method of ASIL Decomposition as defined in ISO 26262 and presented how the method is being interpreted and implemented by practitioners in the field of functional safety. Given the handful of papers and presentations we reviewed, it is obvious that the method is being applied in different ways. We would like to single out two issues we believe are quite important: ASIL Decomposition applicability and top down vs. bottom up application of the decomposition.

- ASIL Decomposition per ISO 26262 is a method not intended to justify reduced ASIL assignments to hardware elements for random hardware failures, but instead focuses on functions and requirements in the context of systematic failures. As ASIL Decomposition deals with requirements decomposition, such decomposition can in some cases be enabled by the existing redundant elements in an architecture that have been already added to help meet the hardware architectural metrics. However, the method itself is intended to provide rules for relating the achieved integrity of a requirement given the integrity of its decomposed redundant requirements. The authors consequently recommend that ASIL Decomposition be used only when requirements are being refined and allocated to different independent components.

- Although bottom up approaches may be needed for development of systems based upon pre-existing design elements, ASILs are inherently associated with requirements, and when a design utilizes ASIL decomposition, it is critical for the final safety case that the top-down ASIL requirement decomposition be confirmed. The authors recommend that design teams explicitly consider the (expected) top-down requirements decomposition even when the system is being designed bottom up, where design elements have pre-assigned ASILs. This will support constructing the final safety case while confirming that the ASIL requirements decomposition matches the ASILs associated with the design elements.

REFERENCES

1. ISO 26262, "Functional Safety - Road Vehicles", November 2011

2. IEC 61508, "Functional safety of electrical/electronic/ programmable electronic safety-related systems", 2nd Edition, 2010

3. Piovesan Andrea and Favaro John, "Experience with ISO 26262 ASIL Decomposition", Presentation at Automotive SPIN, Milano Italy, February 2011

4. Webinar Mahindra Satyam, "Automotive Functional Safety with ISO 26262", January 2012 (http://www.mahindrasatyam.com/events/documents/ISO26262_webinar_presentation.pdf)

5. Cheon J.S., Kim J.S., Jeon J.H., Lee S.M., "Brake By Wire Functional Safety Concept Design for ISO/DIS 26262", Paper 2011-01-2357, SAE World Congress, April 2011

6. Kalmbach J., Wenzel T., and Fassl M., "Recipe for Safe Software", Vector Technical Article, November 2010

7. Ward David, “The uses and abuses of ASIL decomposition in ISO 26262”, Presentation at System Safety Conference, Edinburgh UK, October 2012

8. ISO 26262 Part 10 (FDIS) “Functional Safety - Road Vehicles: Guideline on ISO 26262”, July 2012

9. Ericson Clifton II, “Hazard Analysis Techniques for System Safety”, Wiley, 2005

CONTACT INFORMATION

Joseph D'Ambrosio and Rami Debouk are with General Motors Research and Development. They can be reached at joseph.dambrosio@gm.com and rami.debouk@gm.com

The Engineering Meetings Board has approved this paper for publication. It has successfully completed SAE's peer review process under the supervision of the session organizer. This process requires a minimum of three (3) reviews by industry experts.

ISSN 0148-7191

Positions and opinions advanced in this paper are those of the author(s) and not necessarily those of SAE. The author is solely responsible for the content of the paper.

SAE Customer Service:
Tel: 877-606-7323 (inside USA and Canada)
Tel: 724-776-4970 (outside USA)
Fax: 724-776-0790
Email: CustomerService@sae.org
SAE Web Address: http://www.sae.org
Printed in USA

SAE International®