orig-research
10.1109/MS.2013.118

ISO 26262, a functional-safety standard, uses Automotive Safety Integrity Levels (ASILs) to assign safety requirements to automotive-system elements. System designers initially assign ASILs to system-level hazards and then allocate them to elements of the refined system architecture. Through ASIL decomposition, designers can divide a function's safety requirements among multiple components. However, in practice, manual ASIL decomposition is difficult and produces varying results. To overcome this problem, a new tool automates ASIL allocation and decomposition. It supports the system and software engineering life cycle by enabling users to efficiently allocate safety requirements regarding systematic failures in the design of critical embedded computer systems. The tool is applicable to industries with a similar concept of safety integrity levels.

functional safety, ISO 26262, ASIL decomposition, ASIL automation, ASIL, Automotive Safety Integrity Levels, HiP-HOPS

New Perspectives on Software Quality

Assisted Assignment of Automotive Safety Requirements

Luís da Silva Azevedo, David Parker, Martin Walker, and Yiannis Papadopoulos, University of Hull

Rui Esteves Araújo, INESC TEC

A new tool automates Automotive Safety Integrity Levels' allocation and decomposition. It supports the system and software engineering life cycle by letting users allocate safety requirements regarding systematic failures in the design of critical embedded computer systems. The tool is applicable to industries with a similar concept of safety integrity levels.

ISO 26262 is a functional-safety standard that defines a detailed safety life cycle for developing electrical and electronic systems for passenger cars. The automotive industry has already been using safety analysis and validation and verification techniques to ensure vehicle safety. However, ISO 26262 means that these techniques must now be applied as part of a standardized, industry-wide methodology. One major focus of the standard is traceability: the ability to consistently track how safety requirements are being refined and met from the initial concept development to the final production and operation.

Traditionally, safety requirements have been expressed quantitatively, using maximum target probabilities for system failures that shouldn't be exceeded. However, techniques for quantifying the probability of software failure are immature. Instead, we can use *Automotive Safety Integrity Levels* (ASILs) to represent safety requirements' stringency regarding software and systematic failures in general. ASILs are an adaptation, for the passenger car industry, of the Safety Integrity Level (SIL) defined in IEC 61508. They range from A (least strict) to D (strictest), with QM representing no special safety requirements (normal quality management controls are applied).

We've developed a tool that supports ISO 26262 by employing ASILs to help its users manage top-down allocation and verification of safety requirements regarding systematic failure—including software failure—during the design of complex systems. It will also likely support any similar future standards derived from ISO 26262 in other application domains and thus generally support the design of safety-critical systems in their systems-engineering life cycle.

Automotive Safety Integrity Levels

Designers initially encounter ASILs during concept development. Once they've identified the system's major functions (propulsion, braking, and so on), ISO 26262 requires a hazard analysis. The designers must examine the different functions to determine how they could malfunction in various scenarios (for example, bad weather or busy roads) and what hazards could arise as a result. On the basis of each hazard's severity, controllability, and **exposure, i.e., the probability that the environmental factors of the hazard are present,** , they assign an ASIL to the hazard. Finally, they define safety goals for each hazard, which serve as the system's high-level safety requirements. During the architecture development, they must show how the design fulfills and refines the original safety requirements. (For example, an analysis could show that any components that could contribute to a given hazard can meet the requirements defined by the hazard's ASIL[1]).

ISO 26262 introduces ASIL decomposition, which allows a safety-critical system to meet a particular ASIL target without all its components having to meet that target. If a particular hazard can be caused only by two independent, heterogeneous components failing, the two components can share the responsibility for meeting that hazard's ASIL, rather than each having to meet the full ASIL.

To allow this decomposition, ISO 26262 defines a simple integer algebra. Each ASIL is equivalent to a number: QM = 0, A = 1, B = 2, C = 3, and D = 4. So, two components that share responsibility for meeting ASIL D might individually only be required to meet ASIL B because it produces the same ASIL value: 2 + 2 = 4. This is important because higher ASILs inevitably mean higher costs. To meet more stringent safety requirements, more safety measures must be in place, more work must be performed, and higher-quality components will likely be needed. Therefore, a component's ASIL could significantly affect both development and production costs. So, being able to efficiently allocate ASILs is important so that we can meet the safety requirements without being unnecessarily stringent or expensive.

To manage the complexity of storing all the information the standard requires, automotive companies increasingly use model-based design techniques, particularly integrated modeling platforms and architectural description languages (ADLs) such as the automotive-focused EAST-ADL.[2] These approaches help consolidate much of the information in a single model (often containing multiple layers). They also serve as useful platforms for the analyses that ISO 26262 requires such as failure modes and effects analysis (FMEA).[3] However, many aspects of the ISO 26262 process can be difficult or laborious to perform manually. ASIL decomposition requires detailed knowledge of how the system elements are interrelated and how failures can propagate between them, and can generate many possible ASIL allocations for the system.[4] Some level of automation would therefore greatly benefit developers.

Our technique lets us automate both decomposition and allocation of ASILs.[5] This lets system designers rapidly understand how the system's parts relate to top-level ASIL requirements and helps them evaluate possible ASIL distributions across system components. Furthermore, because our tool is hierarchical, we can apply it recursively to a supply chain, helping to support the layered process of system development. For example, designers might determine an initial allocation of ASILs to system elements at the functional level and then provide these as goals or constraints to component suppliers during hardware development. Those suppliers can apply the same techniques to decompose the component-level ASILs across the different elements of the component subarchitecture. This ensures that the final product still meets the overall ASIL requirements that the system design set.

Allocating and Decomposing Safety Requirements

We developed automatic ASIL decomposition and allocation as an extension to HiP-HOPS (*hi*erarchically *p*erformed *h*azard *o*rigin and *p*ropagation *s*tudies), an advanced safety analysis and design optimization tool used at several large automotive companies.[6,7] HiP-HOPS belongs to the state-of-the-art of semiformal and formal analysis techniques that partly automate safety analysis.[7] It employs an architectural system model that has been annotated with component failure logic. This logic describes how combinations of internal failure modes and failures received at a component's inputs can cause failures at the component's outputs.

The architectural model contains information about the system's topology and how the components are connected.

From this annotated model, HiP-HOPS can combine component failure descriptions to automatically synthesize fault trees. These trees describe how each system hazard can be caused by the propagation and combination of failures through the system architecture. The tool analyzes these trees to find their minimal cut sets, which are the smallest combinations of failures necessary to cause a particular hazard.

HiP-HOPS's system error model takes the form of a system topology and logical networks describing the propagation of failures through that topology. It can serve other purposes than fault tree analysis and FMEA. Users can also employ this knowledge to automatically decompose ASILs that are initially assigned to hazards across the system architecture. So, they can assign lower ASILs (or even QM) to components that don't contribute to failure or contribute only indirectly in conjunction with other components. The users then allocate the original ASILs to only those components that directly cause system hazards. This produces a more economical allocation of ASILs across the system architecture.

We can determine how and where this decomposition occurs on the basis of the logic of the minimal cut sets. Cut sets are presented as the sum of products. So, for example, if hazard H1 is caused by a failure of component *X* (F_X) or failures of components *Y* and *Z* combined (F_Y, F_Z), the cut sets are

- F_X, and
- F_Y AND F_Z.

F_X is a single-point failure; it causes the hazard by itself. In contrast, (F_Y AND F_Z) is a dual-point failure; two failures must occur in conjunction to cause the hazard.

Each cut set inherits the ASIL of the hazard it causes. If a cut set is a single-point failure, that failure will inherit the corresponding ASIL. If the cut set is a multiple-point failure, the failures can share the ASIL according to the ASIL algebra. So, if H1 has ASIL D, we have these options:

- F_Y = QM, F_Z = ASIL D;
- F_Y = ASIL A, F_Z = ASIL C;
- F_Y = ASIL B, F_Z = ASIL B;
- F_Y = ASIL C, F_Z = ASIL A; and
- F_Y = ASIL D, F_Z = QM.

F_X always has ASIL D because it's a single-point failure. F_Y and F_Z could have higher ASILs (for example, both could have ASIL D), but that would be an unnecessarily expensive way to meet the safety requirements. Our tool assigns the ASILs directly to components' internal failures. When a component contains multiple failures, users can further refine requirements. However, if a component allocation is required, they can apply a heuristic a posteriori (for example, by selecting the highest ASIL of a component's failures).

If multiple hazards exist, a given component failure could contribute to more than one hazard. In that case, the failure becomes subject to multiple, potentially conflicting constraints. For example, if in addition to H1 we have H2, which has ASIL B and whose only cause is F_Z, we know that F_Z can never have less than ASIL B. Otherwise, it wouldn't meet H2's safety requirement. That decreases the possible allocations for H1 as well.

Our tool handles all of these conflicting calculations automatically. HiP-HOPS performs multiple fault tree analyses together (one for each system hazard) and shares branches and failures wherever possible. So, according to its knowledge base, it's aware when a particular failure might contribute to multiple hazards. Once the decomposition is complete, the tool lists possible ASIL allocations for all the relevant system failures. Users can then employ different heuristics to evaluate and sort the results, letting them concentrate on the most promising allocations and removing unnecessarily expensive ones.

A Hybrid-Braking-System Example

To better demonstrate our tool, we applied it to a hybrid brake-by-wire system.[8] This system, which

Ricardo de Castro and his colleagues introduced,[9] is for electric vehicles propelled by four in-wheel motors (IWMs). “Hybrid” means that braking occurs through the combined action of electric braking (provided by the IWMs) and friction brakes (electromechanical actuators).[10] During braking, the IWMs transform the vehicle’s kinetic energy into electricity, which charges the power train battery, increasing the vehicle’s range.

Brake-by-wire eliminates the hydraulic connection between the brake pedal and individual wheel brakes. Instead, an electronic bus system handles communication between a central processing unit, which senses brake pedal movement and transforms it into braking force for each wheel, and local processing units, which each subsequently control a pair of braking actuators (see Figure 1).

Figure 1. A hybrid brake-by-wire system. Braking occurs through the combined action of electric braking (provided by the in-wheel motors—IWMs) and friction brakes (electromechanical actuators).

For this simplified case study, we modeled the system to consider the braking function of only one wheel. Because braking forces can be controlled individually for each wheel, this simplification doesn’t affect the failure analysis. The system operates as follows. As the driver presses the brake pedal, an electronic pedal unit senses and processes the driver’s actions. Braking demands are then sent via a duplex bus system to a wheel node controller (WNC). After processing the demands, the WNC generates commands to the power converters to activate the electromechanical brake (EMB) and IWM.

The system uses two batteries: an auxiliary battery feeds the car’s low-voltage loads, and a power train battery handles the high-voltage power transfers with the IWMs. During braking, power flows from the auxiliary battery to the EMB and from the IWM to the power train battery.

We considered two hazards:

- no braking after request from the driver (H1)
- braking more/less than required by the driver (H2).

We could have also examined other hazards, such as unintended braking, but for illustrative purposes, we focused on these two.

Furthermore, each hazard receives an ASIL based on its severity; for illustration, we assigned ASIL D to H1 and ASIL A to H2. (Normally, ISO 26262 risk assessment would determine the appropriate ASILs.) H1 occurs when braking is omitted from both the IWM and the EMB. H2 occurs when either braking device brakes with an incorrect value. This results in two failure expressions:

Omission of Braking = *Omission of EMB.out1* AND *Omission of IWM.out1*,

Value Deviation of Braking = *Value Deviation of EMB.out1* OR *Value Deviation of IWM.out1*.

Analysis Results

After fault tree analysis, HiP-HOPS found these results:

- H1 had 19 minimal cut sets—1 single-point failure and 18 dual-point failures.
- H2 had 11 minimal cut sets—10 single-point failures and 1 dual-point failure.

Using these cut sets, HiP-HOPS computed 125 possible ASIL solutions. Manually evaluating every solution would be difficult, and this is only a simplified test model—larger models could produce thousands of possible solutions. HiP-HOPS lets designers use different cost-based heuristics to automatically evaluate and rank solutions (and filter out redundant ones), letting them focus on the most promising options. There has been some discussion in the functional-safety community concerning the definition of the most realistic ASIL-dependent cost function, and people have proposed many theories. Anecdotal evidence suggests that the biggest cost jump is from ASIL B to C. Here, we used this heuristic:

- QM = 0 cost,
- A = 10 cost,
- B = 20 cost,

- C = 40 cost, and
- D = 50 cost.

This heuristic isn't ideal for real-world situations because two components with the same ASIL requirements will likely have different costs. However, it could serve as a useful guideline, and more accurate heuristics could be developed for each system being analyzed (for example, based on average ASIL costs across the system). Our algorithm is generic in that its parameters can easily be changed to achieve adaptability across industries and isn't tied to a particular heuristic.

To demonstrate further how HiP-HOPS determines possible ASIL solutions, consider the electronic pedal's internal failures. The pedal generates two independent outputs, both of which can deviate from their normal behavior by value (value failures VF_1 and VF_2) and omission (omission failures OF_1 and OF_2). This gives these failure expressions:

- OF_1 and OF_2 occurring simultaneously causes H1.
- VF_1 independently causes H2.
- OF_1 and VF_2 occurring simultaneously causes H2.

In accordance with ASIL algebra, this produces these inequalities:

- ASIL of OF_1 + ASIL of OF_2 ≥ ASIL of H1 (D).
- ASIL of VF_1 ≥ ASIL of H2 (A).
- ASIL of OF_1 + ASIL of VF_2 ≥ ASIL of H2 (A).

According to the second inequality, we can infer that ASIL A is the most economical allocation for VF_1. Regarding the first inequality, we can derive the set of solutions:

- $OF_1 = 0$, $OF_2= 4$;
- $OF_1 = 1$, $OF_2 = 3$;
- $OF_1 = 1$, $OF_2 = 4$ (dominated);
- $OF_1 = 2$, $OF_2 = 2$;
- $OF_1 = 2$, $OF_2 = 3$ (dominated);
- $OF_1 = 2$, $OF_2 = 4$ (dominated);
- $OF_1 = 3$, $OF_2 = 1$;
- $OF_1 = 3$, $OF_2 = 2$ (dominated);
- $OF_1 = 3$, $OF_2 = 3$ (dominated);
- $OF_1 = 3$, $OF_2 = 4$ (dominated);
- $OF_1 = 4$, $OF_2 = 0$;
- $OF_1 = 4$, $OF_2 = 1$ (dominated);
- $OF_1 = 4$, $OF_2 = 2$ (dominated);
- $OF_1 = 4$, $OF_2 = 3$ (dominated); and
- $OF_1 = 4$, $OF_2 = 4$ (dominated).

A dominated solution contains an ASIL that's at least one higher than another solution but contains no lower ASILs. For example, ($OF_1 = 1$, $OF_2 = 4$) is dominated by ($OF_1 = 1$, $OF_2 = 3$). We filter out dominated solutions because they offer no benefits: other solutions are cheaper and just as capable of fulfilling the top-level safety requirements. Furthermore, as the third inequality shows, OF_1 is involved in another cut set; its ASIL and the ones allocated to OF_2 and VF_2 must take into account the constraints of the first and third inequalities. We consider only ASIL QM and ASIL A for VF_2 because it's not involved in another cut set and higher ASILs would always produce dominated solutions.

Table 1 shows the nondominated ASIL allocations for the four failures.

Table 1. The nondominated Automotive Safety Integrity Level allocations for four failures.*

OF_1	OF_2	VF_1	VF_2	Total cost
0	4	1	1	70
1	3	1	0	60
2	**2**	**1**	**0**	**50**
3	1	1	0	60
4	0	1	0	60

* Bold indicates the cheapest solution according to the cost heuristic.

Discussion

We evaluated the previous results only by a cost heuristic. You could choose another set of ASILs based on components' real cost information, if available, or as a result of a designer's system expertise. The possible ASIL allocations that our tool produces are intended only to inform decision making. When allocating ASILs, designers still must take into account other considerations such as the use of legacy components (which are typically treated as QM), the difficulty of proving components' independence, and hardware metrics' impact. Designers are free to draw conclusions from the information provided by the list of possible ASIL allocations and evolve the architecture accordingly, which could result in a new hazard analysis and thus potentially new ASIL constraints. As such, you can view ASIL decomposition as part of an overall iterative design process, and our proposed tool facilitates this.

Selecting the final ASIL allocation requires an informed decision by the designers, taking into account all the factors. This decision might also involve the supplier. In our case study, the cheapest solution assigned ASIL B to H1 and ASIL A to H2. In this case, the supplier could indicate that it has a component that complies with ASIL B for both kinds of failures; using that component would be faster and likely cheaper than developing a new one. This would require revising all the cut sets involving the EMB value failure. However, such design and analysis iteration is much faster with an automated tool such as HiP-HOPS.

ISO 26262 is still relatively new; practitioners still need time to fully understand its principles and develop compatible methodologies. For now, ASIL decomposition is often interpreted and applied in different ways.[11] Our tool isn't intended to prescribe a particular method or interpretation but to help automate decomposition to support whichever method designers are using.

For our tool's application in other domains, the key element to address is a thorough examination of how other standards define SILs. For example, in aerospace, SILs are called DALs (development assurance levels), and they follow slightly different rules than ISO 26262 ASILs. However, once these rules have been identified, our general method can be adapted for other domains, and the relevant constraints can be imposed on possible decompositions according to each standard.

A second issue is scale and complexity. Our tool is inherently hierarchical, which enables its application as part of a supply chain. So, designers can determine an allocation of ASILs for an overall system and pass on those ASIL requirements to component suppliers. The suppliers can then employ the same techniques independently in smaller parts of the design.

Although this helps manage complexity, many combinations of component integrity levels achieve the same result, and the number of combinations grows as these options multiply during refinement. Evaluating every feasible decomposition of ASILs is possible for smaller systems but impractical for large-scale architectures. To solve this, we're working with genetic algorithms and other metaheuristic optimization techniques because they don't need to search an entire design space. We hope that through them, our tool will achieve similar results much more quickly and efficiently.

References

1. S.-H. Jeon et al., "Automotive Hardware Development according to ISO 26262," *Proc. 13th Int'l Conf. Advanced*

Communications Technology (ICACT 11), 2011, pp. 588–592.

2. D. Chen et al., “Modelling Support for Design of Safety-Critical Automotive Embedded Systems,” *Computer Safety, Reliability, and Security*, LNCS 5219, M. Harrison and M. Sujan, eds., Springer, 2008, pp. 72–85.
3. M. Hillenbrand et al., “Failure Mode and Effect Analysis Based on Electric and Electronic Architectures of Vehicles to Support the Safety Lifecycle ISO/DIS 26262,” *Proc. 21st IEEE Int’l Symp. Rapid System Prototyping* (RSP 10), 2010, pp. 1–7.
4. R. Mader et al., “Automatic and Optimal Allocation of Safety Integrity Levels,” *Proc. Ann. Reliability and Maintainability Symp.* (RAMS 12), 2012, pp. 1–6.
5. Y. Papadopoulos et al., “Automatic Allocation of Safety Integrity Levels,” *Proc. 1st Workshop Critical Automotive Applications: Robustness and Safety* (CARS 10), 2010, pp. 7–10.
6. Y. Papadopoulos et al., “Engineering Failure Analysis and Design Optimisation with HiP-HOPS,” *J. Eng. Failure Analysis*, vol. 18, no. 2, 2011, pp. 590–608.
7. M. Bozzano and A. Villafiorita, *Design and Safety Assessment of Critical Systems*, Auerbach Publications, 2010.
8. L.P. Azevedo, “Hybrid Braking System for Electrical Vehicles: Functional Safety,” master’s thesis, Dept. of Electrical Eng., Univ. of Porto, 2012.
9. R. de Castro, R.E. Araújo, and D. Freitas, “Hybrid ABS with Electric Motor and Friction Brakes,” *Proc. 22nd Int’l Symp. Dynamics of Vehicles on Roads and Tracks* (IAVSD 11), 2011, pp. 1–7.
10. S. Savaresi and M. Tanelli, *Active Braking Control Systems Design for Vehicles*, Springer, 2010.
11. *J. D’Ambrosio and R. Debouk, ASIL Decomposition: The Good, the Bad, and the Ugly, tech. paper 2013-01-0195, SAE Int’l, 2013.****Luís da Silva Azevedo*** *is a PhD student in the University of Hull’s Department of Computer Science. His research interests include automation, automotive power electronics, and optimization of requirements distribution in safety-critical systems. Azevedo received an MSc in electrical and computer engineering from the University of Porto. Contact him at l.p.azevedo@2012.hull.ac.uk.*

David Parker *is a lecturer in computer science at the University of Hull. His research interests include the automatic optimization of safety-critical systems. Parker received a* ***PhD*** *in* ***computer science*** *from* ***the University of Hull****. Contact him at d.j.parker@hull.ac.uk.*

Martin Walker *is a computer science lecturer at the University of Hull. His research interests include the development of model-based safety analysis techniques and tools for dynamic systems. Walker received a* ***PhD*** *in* ***computer science*** *from* ***the University of Hull****. Contact him at martin.walker@hull.ac.uk.*

Yiannis Papadopoulos *leads the University of Hull’s Dependable Systems research group. He pioneered HiP-HOPS and contributed to EAST-ADL. Papadopoulos received a DPhil in* ***computer science*** *from York University. Contact him at y.i.papadopoulos@hull.ac.uk.*

Rui Esteves Araújo *is a senior researcher at INESC TEC and an assistant professor in engineering at the University of Porto. His research interests include control theory and its industrial applications to motion control, electric vehicles, and renewable energies. Araújo received a PhD in electrical and computer engineering from* ***the University of Porto****. Contact him at raraujo@fe.up.pt.*