Supporting the Management of Reusable Automotive Software

Xabier Larrucea, Tecnalia

Alastair Walker, Lorit Consultancy

Ricardo Colomo-Palacios, Østfold University College

The OpenCert toolkit helps engineers define safety cases, manage evidence, and comply with automotive standards. The development of a Safety Element out of Context based on a Hall-effect sensor illustrates OpenCert's use.

Automotive software is becoming increasingly relevant[1] and complex.[2] Such systems are based on a wide set of components that are usually provided by third parties. In this changing landscape, the automotive industry faces challenges related to management, software development, and regulatory compliance. As part of this transition, manufacturers now must deal with not only the traditional hardware components but also component-based software,[3] which involves applying the appropriate software engineering skills to safety-critical scenarios.

The automotive domain is also becoming a highly regulated environment in which final products strive to ensure characteristics such as reliability. Standards and certifications[5] introduce a structure that lets stakeholders increase the level of confidence in the finished product.

In the automotive domain, the ISO 26262 standard[6] introduced the *Safety Element out of Context* (SEooC). An SEooC is a safety-related component that isn't specific to a particular product and is therefore reusable. Despite the SEooC's importance, its application is still immature, and its definition and practical implications are still unclear to many users. One of the SEooC's aspects particularly relevant to the development of automotive software is that it highlights and emphasizes the role of assumptions and requirements.

In this article, we discuss OpenCert (www.polarsys.org/proposals/opencert), a safety case management toolkit for dealing with ISO 26262 compliance and SEooC development in complex, safety-critical situations. OpenCert's theoretical background has been described elsewhere.[7,8] Here, we discuss its practical implementation, using a scenario involving Hall-effect sensors (also called Hall sensors).

ISO 26262 and the SEooC

The current version of ISO 26262 was released in 2011 as a standard covering all activities related to road vehicles' functional safety. Sometime in 2017, a new version will be unveiled. ISO 26262 has 10 parts; the SEooC is defined in part 10.

ISO 26262 supports the entire development lifecycle; it presents system, hardware, and software development best practices. The automotive industry doesn't require ISO 26262 certification, but the standard is increasingly being used[9] and is becoming a reference model. Its use is foreseen as a challenge with the emergence of autonomous vehicles because it currently assumes the presence of a driver as the final fallback in case of failures.

From a certification viewpoint, organizations identify which evidence and process-based arguments demonstrate their compliance with ISO 26262.[10]

From a component perspective, organizations identify which assumptions, requirements, and design decisions apply to a component. Figure 1 illustrates these relationships for a Hall-sensor-based SEooC.[11] An SEooC design should take into account not only traditional requirements but also SEooC requirements such as reusability and the overall set of assumptions regarding an SEooC. These elements can be managed with traditional requirements-

capturing techniques or by using an open source tool such as OpenCert.

Figure 1. The relationships between the assumptions and a Safety Element out of Context (SEooC) design for a Hall-sensor-based component, and its adaptation to the ISO 26262 standard.

The compliance process regarding the assumptions, requirements, and design is based on the assessment of product characteristics. Our goal is to not only ensure the resulting products' safety but also instill confidence about a system.[12] ISO 26262 requires a blended method ensuring that processes are followed and that some product characteristics (hardware and software) are satisfied. These characteristics depend on the assumptions defined by stakeholders and are included as functional and nonfunctional requirements as well as standards recommendations. Concerning the software aspect, ISO 26262 part 6 states the following:

> *The specification of the software safety requirements shall be derived from the technical safety concept and the system design in accordance with ISO 26262-4:2011, 7.4.1 and 7.4.5, and shall consider the timing constraints among others.*[6]

All these requirements affect a range of activities covered by ISO 26262, such as the specification of software safety requirements, software unit design and implementation, and software unit testing, which are part of SEooC development. The end result is that an SEooC is

- reusable, considering both software and hardware components and the system as a whole;
- compatible with traditional development phases;
- linkable to other ISO 26262 parts and clauses; and
- checkable, so that the resulting product owns a set of quality characteristics.

An SEooC is considered compliant with ISO 26262 when its assumptions and requirements are met, and all these elements affect the product architecture (see Figure 1). The design in Figure 1 is based on the assumptions and requirements that stem from the analysis.

OpenCert

OpenCert, part of an Eclipse and Polarsys initiative, is a customizable safety assurance toolkit integrated with manufacturers' existing development and safety assurance processes and tooling. It supports the following activities.

Standards and Regulations Information Management

This functionality helps stakeholders identify and interpret standards (such as ISO 26262) and regulations. This information is stored in a common database, and stakeholders can perform traditional operations such as CRUD (create, read, update, and delete).

Assurance Project Management

For an assurance projects in safety-critical scenarios, OpenCert provides functionalities regarding assurance case development, evidence management, and assurance process management. In addition, OpenCert implements the monitoring of compliance with standards and regulations. It supports guidance on, and reuse of, assurance artifacts such as safety cases and evidence. Such functionality helps stakeholders during certification and compliance because it offers transparent product and process assurance and certification with the ability to automate the most labor-intensive activities (such as traceability, compliance checking, assurance process planning, and metrics management). OpenCert also provides facilities to integrate engineering activities with certification activities starting at early stages.

Compliance Management

OpenCert helps engineers assess whether they're adhering to the specified safety practices and standards. In addition, it motivates them to see their work's progress and level of compliance.

Modular and Incremental Certification

OpenCert supports modular safety assurance and certification to enable cost-effective reuse of prequalified building blocks in different contexts (for example, systems, configurations, and upgrades).

Defining an SEooC for Hall Sensors

Hall sensors vary their output voltage on the basis of a magnetic field. They're used in different applications in the automotive domain,[13] including control systems and the control of position and velocity. Our previous research paper provided a set of safety cases for automotive Hall sensors.[11] This article emphasizes the industrial insight on how to model an SEooC using OpenCert and how it satisfies ISO 26262. ISO 26262 includes the development of not only the SEooC but also the item in which the SEooC will be integrated.

Basically, we identified a common set of activities for SEooC development:

- The manufacturer defines the safety assumptions for the SEooC.
- The manufacturer lists the assumptions that will impact safety when the SEooC is integrated in the target item.
- The integrator validates the assumed requirements.
- If a mismatch occurs between the SEooC and target item, the integrator changes either the SEooC or item on the basis of the type of mismatch.

To comply with ISO 26262, our Hall sensor must include these functional-safety requirements:[11]

- The consideration of any component-related risk must take into account its highest level of assurance. This is called Automotive Safety Integrity Level D (ASIL D).
- The maximum magnetic-field strength must be ±250 milliteslas (mT).
- The minimum sensitivity must be 10 LSB/mT.
- The maximum magnetic-field strength must be ±0.1 percent (nonlinearity).
- The maximum magnetic drift must be ±5 μT.
- The onboard diagnostics must detect single faults and latent faults within the allocated time.
- The sensor must be calibrated to ensure that it remains accurate over time, and the related data must be stored in nonvolatile memory.
- The design must be diverse to minimize common causes of failures.
- The nonvolatile memory must be single-fault-tolerant.
- The design must be redundant to prevent a single point of failure from rendering the Hall sensor inoperable.
- The product must remain safe and operational throughout its specified lifetime.

The SEooC boundaries are based on the functional-safety requirements and assumptions made and agreed upon by the manufacturer and integrator. This agreement follows the activities we listed earlier; it's a common process that we explain next.

Checking ISO 26262 Compliance with OpenCert

Figure 2 shows the phases that OpenCert supports; each row in the figure is supported by an OpenCert functionality.

Figure 2. The phases that the OpenCert toolkit supports. Each row is supported by an OpenCert functionality.

The first step is to model the ISO 26262 framework and extract requirements and needs from the standard (see the top left of Figure 2). For this, we use OpenCert's standards editor (see Figure 3). We extracted and adapted the activities from ISO 26262 part 10 for the development of our Hall-sensor-based SEooC. Basically, this process doesn't differ much from traditional component development. We identified six main activities (see Figure 3): system-level assumptions, specification of software safety requirements, software architectural design, software unit design and implementation, software unit testing, and the remaining development phases.

Figure 3. Using OpenCert's standards editor to define activities and artifacts compliant with ISO 26262. This process doesn't differ much from traditional component development.

The second step is to define a basic assurance project, using the OpenCert editor. This project is an instance of the activities identified and compliant with ISO 26262. This step involves identifying assumptions and requirements.

The third step is closely related to defining the assurance project because it takes into account all assumptions and requirements. In it, we use Goal Structuring Notation (GSN) diagrams (see Figure 4) to represent the required assumptions and arguments for building an assurance case. We drill down into all these assumptions and arguments and identify evidence supporting our arguments.

Figure 4. The OpenCert tool and assumptions for hazard assessment by risk analysis (HARA) based on ISO 26262. This tool uses Goal Structuring Notation diagrams.

The ultimate goal is to generate enough confidence in the resulting product, using the assumptions, arguments, and evidence and managing the links among them. For the Hall-sensor-based SEooC, we defined these assumptions:[11]

- The external source will ensure adequate diversity and freedom from interference[14] for the two power supplies to the Hall sensor.
- If the Hall sensor detects an internal error and communicates this to the external source, the external source will take the necessary actions to switch the affected item to the safe state within the defined fault reaction time.
- The external source will ensure that the magnet's position is such that the magnet's mechanical limits aren't exceeded.
- The external source will maintain the recommended operating conditions.
- The external source will meet the Hall sensor's latency requirements such that the ISO 26262 fault-tolerant time interval (FTTI) requirements are met.

The fourth step is to store and manage the evidence. For this, we use Apache Subversion. There are many types of evidence, and they all should be managed. For example, ISO 26262 prescribes *hazard assessment by risk analysis* (HARA) to ensure that the resulting product has taken into account requirements such as situation analysis and hazard identification. This HARA is part of the assumptions and requirements, and its results should be stored in an accessible database. Figure 4 illustrates this situation. All potential hazards are taken into account (and are represented graphically) and are related to requirements and design parts.

One relevant step is to identify which artifacts suggested by ISO 26262 are in the assurance project. These artifacts are evidence supporting our arguments and help us automatically check ISO 26262 compliance. The left side of Figure 5 shows the repository explorer, which stores arguments, assurance projects, evidence, and processes. The right side shows a tree view of artifacts for our SEooC.

Figure 5. A chunk of our SEooC project evidence. The left side shows the repository explorer, which stores arguments, assurance projects, evidence, and processes. The right side shows a tree view of artifacts for our SEooC.

A compliance management panel (see Figure 6) summarizes which ISO 26262 requirements our project has fulfilled and to what extent we covered all the requirements. This panel (which is connected to a webserver) manages the list of baseline elements that our assurance project should satisfy (see the left side of Figure 6). For each ISO 26262 requirement, the compliance status is highlighted in green, orange, or red. The panel also indicates the impact-analysis (IA) status, which is used when the evidence has changed.

Figure 6. The compliance management panel summarizes which ISO 26262 requirements our project has fulfilled and to what extent we've covered all the requirements.

Using OpenCert for the Hall-sensor-based SEooC taught us two main things. First, you should combine the analysis of assumptions and the analysis of functional-safety requirements by using tools that support not just safety case diagrams but also evidence and compliance. Second, engineers must be aware of the evidence supporting each decision, even at the architectural level.

This approach is being improved under the European AMASS (Architecture-Driven, Multi-concern and Seamless Assurance and Certification of Cyber-physical Systems; www.amass-ecsel.eu) project, which will develop cross-domain functionalities.

For a comparison of OpenCert to similar tools, see the sidebar.

Acknowledgments

*We thank our Eclipse and Polarsys supporters, developers, and partners for their great effort on the OpenCOSS (*Open P*latform for* E*volutionary* C*ertification of* S*afety-Critical* S*ystems) project, especially Janusz Studziżba, Dariusz Oszczędłowski from Parasoft, and Angel Lopez from Tecnalia.*

References

1. J. Mössinger, "Software in Automotive Systems," *IEEE Software*, vol. 27, no. 2, 2010, pp. 92–94.
2. C. Ebert, "Looking into the Future," *IEEE Software*, vol. 32, no. 6, 2015, pp. 92–97.
3. I. Crnkovic, J. Stafford, and C. Szyperski, "Software Components beyond Programming: From Routines to Services," *IEEE Software*, vol. 28, no. 3, 2011, pp. 22–26.
4. M. Polo et al., "Test Automation," *IEEE Software*, vol. 30, no. 1, 2013, pp. 84–89.
5. X. Larrucea, A. Combelles, and J. Favaro, "Safety-Critical Software," *IEEE Software*, vol. 30, no. 3, 2013, pp. 25–27.
6. *ISO 26262—Road Vehicles—Functional Safety—Part 1: Vocabulary*, Int'l Standard Org., 2011.
7. X. Larrucea et al., "Standards-Based Metamodel for the Management of Goals, Risks and Evidences in Critical Systems Development," *Computer Standards & Interfaces*, Nov. 2016, pp. 71–79.
8. J.L. de la Vara et al., "Model-Based Specification of Safety Compliance Needs for Critical Systems: A Holistic Generic Metamodel," *Information and Software Technology*, Apr. 2016, pp. 16–30.
9. C. Hernandez and J. Abella, "Timely Error Detection for Effective Recovery in Light-Lockstep Automotive Systems," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 11, 2015, pp. 1718–1729.
10. B. Gallina, "A Model-Driven Safety Certification Method for Process Compliance," *Proc. 2014 IEEE Int'l Symp. Software Reliability Eng. Workshops* (ISSREW 14), 2014, pp. 204–209.
11. X. Larrucea, S. Mergen, and A. Walker, "A GSN Approach to SEooC for an Automotive Hall Sensor," *Systems, Software and Services Process Improvement*, C. Kreiner et al., eds., Springer, 2016, pp. 269–280.
12. L. Strigini, "Assessment Techniques, Certification and [What Else We Need for] Confidence in Software," *Proc. 2014 IEEE Int'l Symp. Software Reliability Eng. Workshops* (ISSREW 14), 2014, pp. 484–484.

13. C. Chen et al., “Magnetic Induction Model of the Hall Sensor: Analysis and Simulation of an Automotive Shift,” *IEEE Vehicular Technology Magazine*, vol. 7, no. 1, 2012, pp. 38–43.
14. F. Leitner-Fischer, S. Leue, and S. Liu, “Automated Freedom from Interference Analysis for Automotive Software,” *Proc. 4th Int’l Workshop Critical Automotive Applications: Robustness & Safety* (CARS 16), 2016; homepages.laas.fr/mroy/CARS2016-papers/CARS2016_paper_14.pdf.

Xabier Larrucea *is a research scientist at Tecnalia. His research interests include software engineering, metamodeling, software process improvement, and safety-critical applications. Larrucea received a PhD in software engineering from Universidad del Pais Vasco. He’s an IEEE Senior Member and is the* IEEE Software *Constituency Ambassador for Spain and Latin America. Contact him at xabier.larrucea@tecnalia.com.*

Alastair Walker *is the owner of, and a consultant at, Lorit Consultancy. He has extensive knowledge of developing embedded systems in safety-related industries. He’s a TÜV Rheinland Functional Safety Engineer. Walker received an MSc in electrical and electronic engineering from Edinburgh Napier University. Contact him at alastair.walker@lorit-consultancy.com.*

Ricardo Colomo-Palacios *is a professor of computing at Østfold University College. His research interests include software engineering improvement, software project management, software tools, and DevOps. Colomo-Palacios received a PhD in computer science from Universidad Politécnica de Madrid. Contact him at ricardo.colomo-palacios@hiof.no.*

Popular Safety Case Tools

Table A lists the most popular safety case tools used by industry. (As the table indicates, some of these tools are no longer supported. Also, owing to space limitations, we’ll provide a deeper tool analysis in a later article.) Most of these tools can represent safety cases in GSN. However, only OpenCert combines safety case definition with evidence and compliance management based on ISO 26262. In addition, it provides modules for safety case management that are appropriate for reusing components such as a Safety Element out of Context (SEooC). For more on OpenCert and SEooCs, see the main article.

Table A. Safety case tools and their functionalities.

Tool	Company	Website	Is available	GSN *	Evidence	Compliance management	SEooC* or modular approach
Access (Assistance Case Construction and Evaluation Support	Univ. of Virginia	www.cs.virginia.edu/~pvs5x/research.html	No	Yes	No	No	No

System)							
ACEdit	Univ. of York	code.google.com/archive/p/acedit	Yes	Yes	No	No	No\
AdvoCATE[1]	NASA	N/A	Yes	Yes	No	No	Yes
ASCE (Adelard Safety Case Editor)	Adelard	www.adelard.co.uk	Yes	Yes	No	No	Yes
Astah GSN	Astah	astah.net/editions/gsn	Yes	Yes	No	No	Yes
CertWare	NASA	nasa.github.io/CertWare	Yes	Yes	No	No	Yes
D-Case	Univ. of Electro-Communications	www.dcase.jp github.com/d-case/d-case_editor	Yes	Yes	No	No	No
e-Safety Case	Praxis Critical Software	www.rmri.co.uk/what-we-do/software/e-safety-case	No	No	No	No	No
Freeware Visio add-on	Univ. of York	www.goalstructuringnotation.info/archives/41	No	Yes	No	No	Yes
GSN CaseMaker ERA	Edif Group	N/A	No	Yes	No	No	No
ISCaDE (Integrated Safety Case Development Environment)	RCM2	www.iscade.co.uk	Yes	No	No	No	Yes
ISIS	High Integrity Solutions	N/A	No	Yes	No	No	No
NoR-STA	Argevide	www.argevide.com/en/products/assurance_case	Yes	No	Yes	Yes	No
OpenCert	Eclipse and Polarsys	www.polarsys.org/projects/polarsys.opencert	Yes	Yes	Yes	Yes	Yes

* GSN = Goal Structuring Notation; SEooC = Safety Element out of Context.

Reference

1. E. Denney, G. Pai, and J. Pohl, “Composition of Safety Argument Patterns,” NASA; ti.arc.nasa.gov/publications/30619/download.