PROJECT SUMMARY

FUSE

FUnctional Safety and Evolvable architectures for autonomy

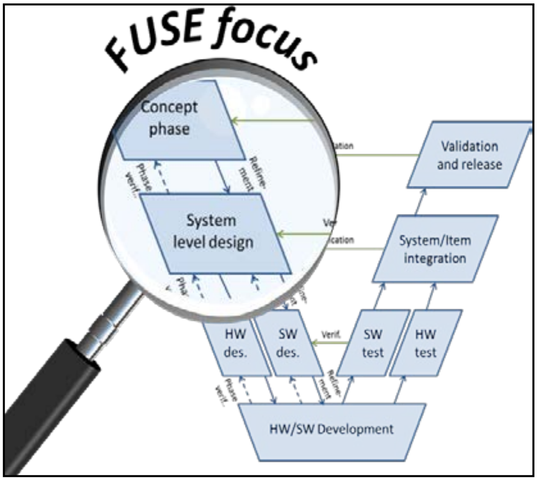
Project partners

Funding

The Project

In this report we summarise the results of the FUSE project, which has focused on system architectures and functional safety for autonomous road vehicles. Current automotive systems and functional safety standards are evolving, but have so far only considered autonomy to a limited extent. Functional safety considerations and the scalability and cost-efficiency of architectures represent potential blocking factors for introducing new autonomy functions. The project addressed the identified autonomy related problems by:

- Developing an understanding of limitations of current architectures and safety standards.
- Developing requirements for autonomy considering safety and architecture.
- Creating guidelines and update suggestions for ISO 26262.
- Creating a reference architecture



The FUSE project is a collaboration between Comentor, KTH, Qamcom, Semcon, SP and Volvo Cars and was partly financed by Vinnova through the FFI program Vehicle and Traffic Safety. It ran from Q4 2013 to Q3 2016 with a budget of 13 MSEK.

The Problem

In a near future it is likely to find more or less autonomous (e.g. self-driving) cars in series production. For example, Volvo Car Group's project 'Drive Me' aims (2017) to feature 100 self-driving cars on public roads in everyday driving conditions. When autonomous vehicles are put in series production, the electrical/electronic (E/E) system will need to provide all functionality to assure that the vehicle behaves safely on the road without any involvement of a driver. This includes taking care of all unplanned and unforeseen situations that may occur once the autonomous drive (AD) is activated.

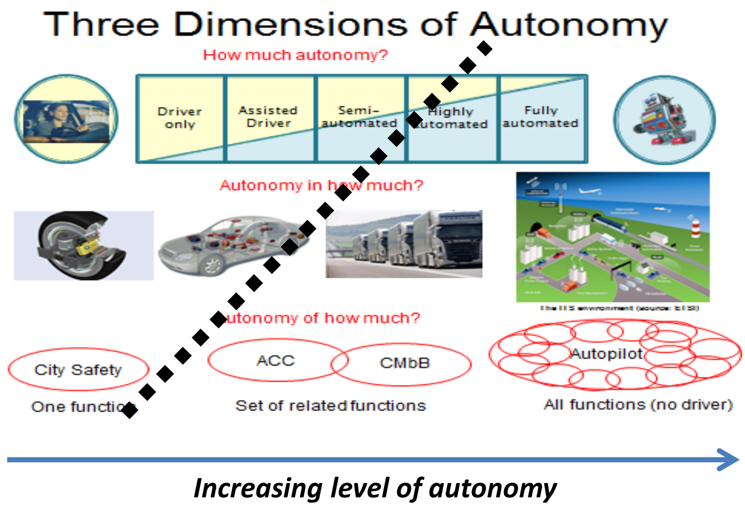


Figure 1: What about functional safety when crossing the dotted line?

Figure 1 shows levels of autonomy in different dimensions: from driver only via semi-automated to fully automated vehicles, from single parts to the entire transportation system, or from single independent functions to all functions needed for piloting a vehicle in cooperation. But what happens to functional safety when crossing the dotted line, i.e. where the E/E system fully takes over the driving task from the driver in at least some driving scenarios?

• How to define functional safety in this context?	⇒	Lacking definitions in ISO 26262
• How to achieve it?	⇒	Demand for architectural patterns, and division of responsibility
• How to prove it?	⇒	Demand for new compositional safety arguing

While other efforts in the area of self-driving vehicles have looked at problems such as sensing, control strategies and algorithms, the FUSE project has focused on functional safety, scalable architectures, and new methods for development and safety analysis. The chapters below each introduce one of the identified problems within this scope, and the proposed solutions that have resulted from the project.

Contents

Methodology Framework for Hazard Analysis and Risk Assessment

In ISO 26262:2011, the scope and requirements of an E/E function are parts of the item definition, which is an input to the hazard analysis. This means only situations and hazards that affect the already defined function are the focus of the analysis. The risk in the context of AD is that the function specification is too narrow, or that there are gaps between functions, so that relevant hazardous events are not handled by any of the functions contributing to the AD. We therefore argue that an objective of the hazard analysis should be to make sure the proposed functions contributing to AD together covers all hazardous events relevant to the goal of autonomous operation. The result of this analysis may include necessary changes to the scope and requirements of the function(s).

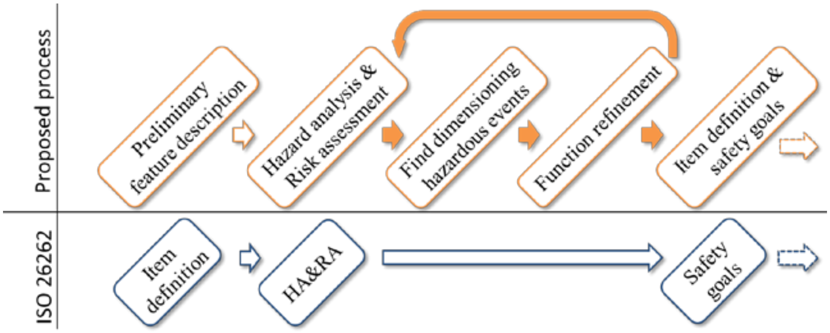

Proposed process
ISO 26262
Preliminary feature description
Hazard analysis & Risk assessment
Find dimensioning hazardous events
Function refinement
Item definition & safety goals
Item definition
HA&RA
Safety goals

Figure 2: Iterative hazard analysis and function refinement process.

In FUSE we propose an iterative process, shown in Figure 2, that combines hazard analysis and function requirements refinement. The process starts with a light-weight preliminary feature description. Based on this input, hazardous events and functional requirements are iteratively refined until the scope of the function is clear and useful safety goals can be created.

A key challenge is to achieve confidence that the hazard analysis covers all relevant situations and hazards, as the AD needs to cope with all situations that might occur once it is activated. Our process therefore includes techniques aimed at reducing the risk of missing relevant hazardous events. One of these tools is the use of generic situation and hazard trees. The process is described in more detail in [14].

Guidelines for Detailed Hazardous Events

In ISO 26262 it is prescribed that the HA&RA shall be done, and that there shall be performed a verification activity showing the "completeness with regard to situations and Hazards". However, there is no information how to solve the problem of showing completeness. When have all Hazardous Events been analysed? There are a number of cases when adding a new Hazardous Event would not extend the list of already identified Safety Goals. Such Hazardous Events are of no interest, as the objective of the list of Hazardous Events, is to identify the list of Safety Goals. It would be beneficiary to have a set of rules that automatically can check whether a given candidate Hazardous Event would generate a new Safety Goal, or if it can be considered as redundant.

In FUSE we have formulated of a number of rules enabling to reduce the potentially infinite set of candidates of Hazardous Events to a limited number, still sufficient to cover all safety goals. By this we can reach two things. Firstly it enables to produce a list of detailed hazardous event (not producing too conservative safety goals) that still have a limited length. Secondly, it can be used as a tool for reviewing the completeness of a HA&RA (hazard analysis and risk assessment). Details about the rules can be found in [6].

Methodology for HMI Safety Argumentation

When introducing an autopilot which in some driving situations takes full responsibility to drive the vehicle, it becomes crucial to ensure safe transitions between the manual and the automated driver. The existence of dual driving modes brings two new sources of risk. The first is mode confusion which is when the manual driver and the autopilot have different opinion of whether a transition of responsibility hs occurred or not. Especially it is critical to avoid that a situation occurs when none of the drivers considers itself as responsible. The second is unfair transition. This is when the previous non-responsible driver finds itself as responsible without agreeing to this previously.

We propose to define a safe transition as a transition where either any (!) manual single mistake or an E/E failure, or combination of these, leads to an unfair transition or mode confusion. Furthermore, we demonstrate on a system example how to allocate safety requirements on system elements to ensure safe transitions.

Results from this example show that it is sufficient to allocate safety requirements on the sensor and lock of a single lever to ensure safe transitions. No safety requirements are needed on visual feedback to the driver, e.g., displays. We remark that the example implementation by no means is a unique solution to the safe transitions problem. . A complete description of how to argue for safe transitions is found in [10].

Disarming the Trolley Problem Paradox

The ‘trolley problem’ is a classical ethical dilemma saying that something surprising happens and then there are two choices on how to act, none of them good. If no active choice is made then some persons are killed, and if an active choice is made the consequence is that other but fewer persons die. The dilemma is named from the scenario of a trolley running down a track unable to brake, approaching a fork point. You are beside the track having the time to reach a lever which can enable you to make the trolley change track. If you do not act, five people will be killed. If you pull the lever and make the trolley turn, another but single person will die.

The problem is to illustrate the conflict between what is called utilitarianism and deontology, respectively. In the former case you try minimize the total harm (here minimize the number of deaths), and in the latter case you avoid to do things that always are wrong (here you avoid to actively kill a certain person).

In FUSE we argue that the introduction of self-driving cars can solve (sic!), and not cause, this ethical dilemma. In short the solution to the trolley problem is that a self-driving car must be able to estimate its own operational capability for handling surprising situations, and adjust its own tactical behaviour accordingly. By limiting the risk for the case of not being able to handle all surprising events in a similar way as for other safety goals today, the remaining risk for the trolley problem can be argued as low as any other acceptable risk of vehicle E/E implemented functionality. A complete description of the solution is found in [11].

Architectural Pattern for Functional Safety Concept

Architecting autonomous vehicles involves a multitude of challenges. New and/or extended functionalities for perception, navigation, decision making and control will pose new requirements for computation and communication. System architectures also have to satisfy stringent safety, availability and other requirements of autonomous systems while also enabling cost-efficient realization, maintenance and evolvability over the life-cycle of platforms and vehicles! Meeting those requirements becomes even more challenging considering a variety of business cases, different levels of automation and changing regulations.

A further key challenge is the integration of the technological fields of AI with traditional safety related embedded control systems. These two fields have very different traditions and emphasis, yet need to be reconciled to provide higher levels of automation. Current automotive architectures have been developed more or less bottom up while state of the art AI and control systems architectures provide clean hierarchical structures. Current automotive functionalities are also designed to be fail-silent – a characteristics that is generally considered insufficient for higher levels of automation.

Architectural work within the FUSE project has investigated such challenges, [1], and provided results in terms of functional reference architectures, patterns and architecting methodological guidance. Experiences from the architecting of autonomous systems have been synthesized into a so-called functional reference architecture, also referred to as a functional architecture view (FAV), see Figure 3 (see [5,15] for further details). The FAV constitutes a generic solution pattern for a given set of system behaviours, which may then be implemented in a variety of ways, considering either novel or incremental designs. The FAV identifies key functional components and their interactions, and also divides these components into a vehicle platform, a "cognitive driving intelligence", and an off-board management system.

To use and instantiate the reference architecture there is a need for further guidance. In particular, strategies for legacy (how much to reuse, trade-offs involved), decentralization of the functionalities, and architecture patterns that provide cost-efficient safety and reliability/availability have been investigated, [2,5,7,13].

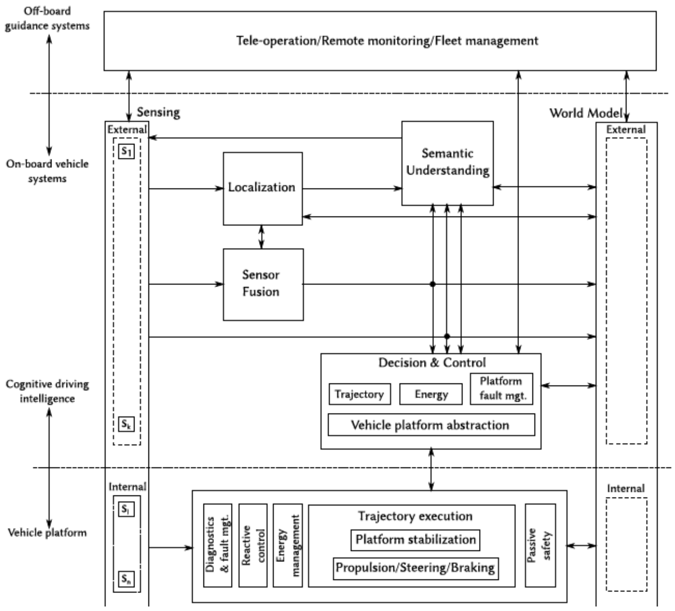


Figure 3: Functional reference architecture for autonomous driving.

One of the most challenging aspects of self-driving vehicles is the design and safety argumentation for sensor fusion, i.e. environment perception. In [9], we identify generic functional safety requirements to be allocated on an environment perception block.

In this functional safety concept, we have a full separation of responsibility between the two blocks Environment Perception (EP) and Decision and Control (DC), respectively. The EP block has no responsibility to decide the speed of the vehicle or distance marginal to other objects, but is just required to tell the absence and the presence of each pre-defined object type.

A good strategy that enables safety assessment according to ISO26262 implies that the environment perception block should address its safety requirements for all the ASIL attribute values

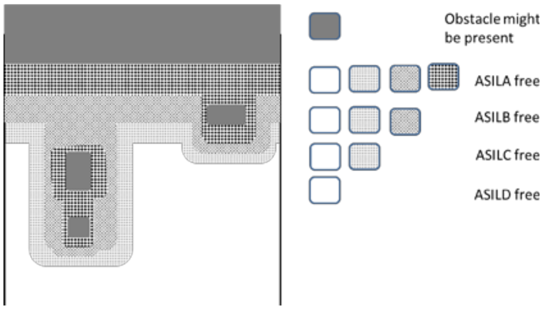


Figure 4: Example map of anti-object of type Vehicle. The ego vehicle is in the bottom of this picture

simultaneously, see Figure 4. The DC block, then has four sets of maps, each with a separate ASIL value, to which the driving is adapted. Depending on the distances to the different known anti-objects, it can match the driving style tactics to fulfil the safety requirements.

Methodology Framework for Refinement of Safety Requirements

In ISO 26262, every safety requirement refinement needs to be proven complete and correct by means of a verification activity. This implies that for the item it shall be verified that the set of SGs (Safety Goal) are complete, for each SG it shall be verified that a certain set of FSRs are complete, for each FSR it shall be verified that a certain set of TSR are complete, etc.
The "distance" between two requirement levels e.g. SG and FSRs (or any other adjacent requirements levels) is denoted the *Semantic Gap*, see Figure 5. The concept phase of ISO 26262 (part 3) describes how SGs are determined from the results of the HA&RA. SGs are refined into FSRs, which implies that the SGs can be interpreted as top-level safety requirements in a layered requirement hierarchy. A SG is a high-level description of an objective on vehicle level, and the refinement of the SG to reach FSC may need a substantial amount of assumptions, domain knowledge or other input. If no or only weak arguments for the refinement of SG to FSC

exist, then verification to argue correctness and completeness is at best difficult.

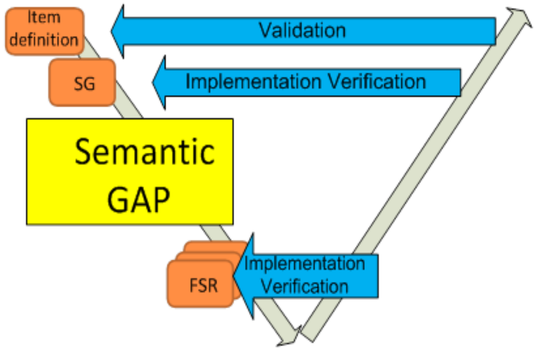


Figure 5 The semantic gap

A requirement (the upper level of two adjacent requirement levels) is refined into a composition of lower level requirements and rationale, known as satisfaction arguments. The satisfaction arguments shall be collected for the composition, see Figure 6. This bridge of information should "fill" the semantic gap. Satisfaction arguments may be e.g. assumptions, domain knowledge, design patterns. This is essential in almost every non-trivial refinement. The rationale justifies the "refinement path taken" through the semantic gap and improves to traceability.

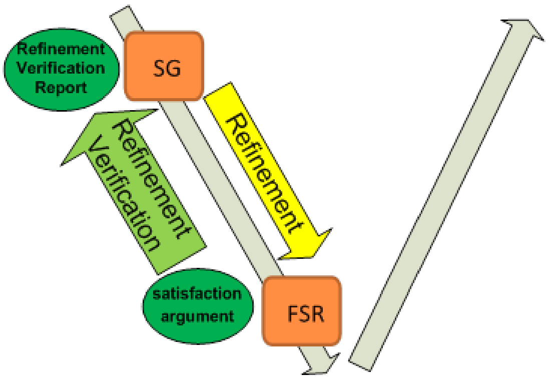


Figure 6 Activities between adjacent requirement levels.

Satisfaction arguments are used in *refinement verification* to prove completeness and correctness. The result is a refinement verification report that give proof that the composition of refined requirements fulfils the requirement (upper level). We distinguish refinement verification from implementation verification. The latter concerns ensuring that requirements are correctly implemented, see. Obviously the two activities have different goals and may utilise different tools.

Summary and Conclusions

The main result of FUSE has been to structure the area of functional safety for autonomous road vehicles, especially to identify what is specific for autonomous vehicles in comparison with ordinary manually driven road vehicles as of today [12].

One identified difference is that the task to perform a complete hazard analysis and risk assessment is based on an implicit 'item definition' (the scope of the vehicle functionality responsibility). The FUSE project has two preliminary results how to address this to reach a complete and efficient set of safety goals. The first is a methodology framework for hazard analysis and risk assessment [3,14], and the second are guidelines how to formulate a detailed but still manageable set of hazardous events [4,6].

Another difference is the important area of safely assessing the transition of responsibility between a manual driver and the autopilot of the vehicle. The FUSE project has formulated a preliminary methodology for HMI safety argumentation [10].

A major identified difference regarding autonomous vehicles, is related to the decision hierarchy of driving. Today's ADAS functionality are focused on the lower level (operational), leaving the higher decision levels (strategic and tactic) to a manual driver. The transition of tactical decision to an autonomous road vehicle has been identified as a key area. The FUSE project has identified architectural patterns applicable for this and for the complex functionality implied by autonomous driving [2,5,7,13]. An interesting conclusion of the value of dividing safety the responsibility between the hierarchy levels, is that a general argument can be constructed how to disarm the so called trolley problem [11]. Another conclusion is that safety requirements need to be formulated in one version for each ASIL level, and this is true when they are refined down in the implementation structure. A high ASIL level will not necessarily dominate a low level, and what version of a certain safety requirement that is the dimensioning one, may change during run time [9].

Further Reading

The project has resulted in an number of scientific publications. Abstracts and links can be found on the project website: http://www.fuse-project.se/. The following publications are related to the projct:

1. Sagar Behere, Fredrik Asplund, Andreas Söderberg and Martin Törngren. *Architecture challenges for intelligent autonomous machines: An industrial perspective.* 13th International Conference on Intelligent Autonomous Systems (IAS-13), July 2014.
2. Sagar Behere and Martin Törngren: *A Functional Architecture for Autonomous Driving*. 1st Workshop on Automotive Software Architectures (WASA), May 2015.
3. Carl Bergenhem, Jörgen Tryggvesson, Rolf Johansson, Andreas Söderberg, Martin Törngren, Jonas Nilsson and Stig Ursing*. How to Reach Complete Safety Requirement Refinement for Autonomous Vehicles.* CARS 2015 - Critical Automotive applications: Robustness & Safety, September 2015.
4. Rolf Johansson. *The Importance of Active Choices in Hazard Analysis and Risk Assessment.* CARS 2015 - Critical Automotive applications: Robustness & Safety, September 2015.
5. Sagar Behere. *Reference Architectures for Highly Automated Driving.* Doctoral Thesis, KTH, January 2016.
6. Rolf Johansson. *Efficient Identification of Safety Goals in the Automotive E/E Domain.* ERTS2 2016 - 8th European Embedded Real Time Software and Systems Congress, January 2016.
7. Sagar Behere, Xinhai Zhang, Viacheslav Izosimov and Martin Törngren. *A Functional Brake Architecture for Autonomous Heavy Commercial Vehicles.* SAE World Congress, April 2016.
8. Naveen Mohan, Martin Törngren, Viacheslav Izosimov, Viktor Kaznov, Per Roos , Johan Svahn, Joakim Gustavsson and Damir

Nesic. *Challenges in architecting fully automated driving; with an emphasis on heavy commercial vehicles.* 2nd Workshop on Automotive Software Architectures (WASA), May 2016.

9. Rolf Johansson and Jonas Nilsson. *The Need for an Environment Perception Block to Address all ASIL Levels Simultaneously.* 2016 IEEE Intelligent Vehicles Symposium (IV), June 2016.
10. Rolf Johansson, Jonas Nilsson and Martin Kaalhus. *Safe Transitions of Responsibility in Highly Automated Driving.* The Ninth International Conference on Dependability (DEPEND 2016), July 2016.
11. Rolf Johansson and Jonas Nilsson. *Disarming the Trolley Problem – Why Self-driving Cars do not Need to Choose Whom to Kill.* 4th International Workshop on Critical Automotive Applications: Robustness & Safety (CARS 2016), September2016.
12. Rolf Johansson, Jonas Nilsson, Carl Bergenhem, Sagar Behere, Jörgen Tryggvesson, Stig Ursing, Andreas Söderberg, Martin Törngren, and Fredrik Warg. *Functional Safety and Evolvable Architectures for Autonomy.* Chapter in Book: Automated Driving Safer and More Efficient Future Driving, Editors: Daniel Watzenig and Martin Horn, ISBN 978-3-319-31893-6 ,Springer, September 2016.
13. Sagar Behere and Martin Törngren. *Systems engineering and architecting for autonomous driving.* Chapter in Book: Automated Driving Safer and More Efficient Future Driving, Editors: Daniel Watzenig and Martin Horn, ISBN 978-3-319-31893-6 ,Springer, September 2016.
14. Fredrik Warg, Martin Gassilewski, Jörgen Tryggvesson, Viacheslav Izosimov, Anders Werneman, and Rolf Johansson. *Defining Autonomous Functions Using Iterative Hazard Analysis and Requirements Refinement.* 5th International Workshop on Next Generation of System Assurance Approaches for Safety-Critical Systems (SASSUR), September 2016.
15. Sagar Behere and Martin Törngren. A functional reference architecture for autonomous driving. Journal of Information and Software Technology, 2016. Elsevier.

Contact Information

- Rolf Johansson, rolf.johansson@sp.se, +46 105 165546
  (Project coordinator)
- Carl Bergenhem, carl.bergenhem@qamcom.se, +46 31 3819021
- Jonas Nilsson, jonas.nilsson@volvocars.com, +46 31 597627
- Martin Törngren, martint@kth.se, +46 8 7906307
- Ola Örsmark, ola.orsmark@comentor.se, +46 733 626909
- Stig Ursing, stig.ursing@semcon.com, +46 31 7615016

FUSE

***FU**nctional **S**afety and **E**volvable architectures for autonomy*