Analysis of safety requirements evolution in the transition of land transportation systems toward autonomy

Youssef Damak, Marija Jankovic, Yann Leroy, Bernard Yannou

▶ **To cite this version:**

Youssef Damak, Marija Jankovic, Yann Leroy, Bernard Yannou. Analysis of safety requirements evolution in the transition of land transportation systems toward autonomy. 15th International Design Conference - DESIGN 2018, May 2018, Dubrovnik, Croatia. pp.2845 - 2854, 10.21278/idc.2018.0448 . hal-01785779

HAL Id: hal-01785779
https://hal.science/hal-01785779

Submitted on 12 Jun 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Analysis of Safety Requirements Evolution in the Transition of Land Transportation Systems toward Autonomy

Y. Damak, M. Jankovic, Y. Leroy and B. Yannou

In: *DS92: Proceedings of the DESIGN 2018 15th International Design Conference*. 2018. p. 2845-2854.

Abstract

Autonomous Vehicles (AV) are the future evolution of Land Transportation Systems (LTS). They promise an improvement in road safety. However, safety requirements stay a big challenge for their development. The literature presents a lack of insight on the way LTS safety requirements will evolve. This paper proposes an analysis method of LTS safety requirements evolution toward AV. The ASIL metric is used to evaluate the safety criticality. The application on two case studies, the steering by wire and platooning systems, results in a better understanding and characterization of this evolution

Keywords: model based engineering, design process, safety analysis, autonomous vehicles

1. Introduction

Autonomous Vehicles (AV) design and development are gaining more and more focus in the last years. Many carmakers got in the heated race to be the first to industrialize a safe autonomous car. AVs are the evolution of current Land Transportation Systems (LTS) and will introduce new forms of urban mobility and usages (Dokic et al., 2015). They will likely introduce considerable changes in society. Often mobility experts consider road safety improvement as one of their major positive impacts. On the other hand, the safety requirement is one of the major challenges of AV design and development (Hulse et al., 2018).

Our field observations and interviews showed that carmakers want to reuse existing Advanced Driver Assistant Systems (ADAS) for the AV development, with minimal redesign and reworking. However, the Automotive Safety Integrity Level (ASIL) defined in the ISO 26262 constrained the development of ADAS development and their integration in today's car (ISO, 2011). Due to the systemic changes brought by AV, the evaluation of ASIL on the system level will impact their system architecture. In the scope of this paper, the authors propose to address how to analyse the safety requirements evolution in the transition of LTS toward AV. The analysis is applied on two evolution types with the steering by wire and the platooning use cases.

The remainder of this paper is organized as follows. The reviewing of the literature in presented in section 2, followed by the analysis method in section 3. The section 4 presents the results of the analysis on two case studies. The paper ends with a discussion on the validity of the method and its results in section 5.

2. Literature review

2.1. Systems evolution

Autonomous Vehicles (AV) are considered the cells for the future mobility. They are actually the evolution of current LTS. As such, their design is constrained and limited by previous vehicles architectures, in particular current ones. Systems evolution was studied in the literature from many aspects. The TRIZ theory implies that all technical systems are subject to 8 laws of evolution. With each evolution stage, contradictions are generated. The TRIZ methodology proposes different methods, tools and algorithms to help design during the evolution of technical systems. The matrix of contradictions is an example of tools useful to solves the contradictions (Savransky, 2000). Even though powerful, the TRIZ approach doesn't propose a framework to adapt the current design process to the evolving system. In addition to TRIZ and thanks to new methods and computational capabilities for data analysis and decision making, knowledge-based approaches emerged to orient designer during systems evolution. Gerdes et al. propose a tool to support constraint-based decision making in the context of systems evolution using previous knowledge about the system for new architectures evaluation (Gerdes et al., 2015). However, the knowledge feed-back for AV is still too small to know architecturally significant factors to efficiently generate and evaluate architectures.

External causes of systems evolution are not considered in the TRIZ theory. System's environmental changes were explored by Bürger et al. from the security knowledge point of view. They proposed a model-based framework using ontologies to adapt the co-evolution of systems with the organization's security knowledge. To ensure the co-evolution, they use events modelling, graph algorithms for and graphs synchronization and transformation. They could realize meta-models synchronization and identify property violation (Bürger et al., 2018). This approach is specific to security knowledge evolution. Adequate meta-models and context analysis have to be performed to reuse it on other contexts. Other studies consider the system's interdependencies with (design) processes and stakeholders requirements as external factors of systems evolution. Hoang et al. proposed a model to support mechatronic systems evolution based on the interdependencies of products, processes and resources (Hoang et al., 2017). However, the modelling concept needs to be accompanied with an interdependencies analysis specific to each studied system.

This literature review of systems evolution showed a lack of understanding for significant internal and external factors in the AV's context. These factors are numerous. So, for a better contribution to the understanding of LTS evolution, this study is focused on only one factor. The following literature review explains the importance of safety requirements factor for this topic.

2.2. Safety requirements evolution

Public acceptance of AV is often positively associated with the improvement of road safety. AV are perceived as a great opportunity to increase the safety of pedestrian and passengers (Hulse et al., 2018). Ironically, safety is one of the major challenges of AV's design and development. If AV couldn't ensure a minimal safety level (unknown yet), it will have a negative impact on their public acceptation (Dokic et al., 2015; ERTRAC, 2015). For this reason, the future evolution of safety requirements of LTS represents one of the major factors in the evolution of their design processes.

The literature contains a few studies in eliciting safety requirements evolution and analysing its impact. Luiz and Mikulski analysed the safety requirements changes of spacecraft's software. These changed where deduced from post-launch anomalies thanks to reporting mechanism whenever the system's behaviour differs from the requirement. The criticality level was evaluated according to the standards of the domain. A fourth of the highly safety-critical anomalies resulted in software requirements changes (Lutz and Mikulski, 2003). Transposed on AV's case, the safety criticality of functions would be evaluated only post-tests.

Anderson and Felici proposed metrics to measure software requirements evolution. They proposed the Requirement Maturity Index (RMI) and its 2 extensions the Requirement Stability Index (RSI) and the Historical Requirement Maturity Index (HRMI) (Anderson and Felici, 2002). These metrics are useful to understand and monitor the requirements evolution. However, they work on past and present software releases, and are not yet used for requirements evolution forecasting. On safety requirements forecasting, Szarata and Dźwigoń predicted the impact planned new roads in Poland on road safety. They analysed data such as average traffic volume to quantitatively predict the density of accidents, injuries and fatalities (Szarata and Dźwigoń, 2015). Their approach is inspiring; however, to the best knowledge of the authors, there is no model to quantitatively predict safety criticality of new AV designs. For this particular reason, safety analysis of AVs and Cyber-Physical Systems (CPS) in general became an active research domain.

2.3. Safety analysis of Cyber-Physical Systems

Izosimov and Levholt synthesised the safety-based design flow of CPS. Their proposition complied with the different standards such as ISO 26262 and Do-178B (Izosimov and Levholt, 2015). Safety oriented activities were added to a standard design process as follows: Hazard Analysis and Risk Assessment (HARA) follow to the system definition. A system's safety analysis is applied on system architecture for a safety-based architecting. The process is iterative until the safety requirements are satisfied (Izosimov and Levholt, 2015). Most of the research works join both activities with different synergy. Matsubasa and Aoyama proposed a combination of safety requirements meta-modelling with scenarios modelling and hazard analysis. It was applied for the safety analysis of Toyota's Crown Majesta. It resulted in safety requirements models used for the design of mitigation actions. They finally evaluated the safety model after the mitigation plan with Bayesian network analysis (Matsubara and Aoyama, 2017). Bayesian network analysis was also used by Duran et al. to evaluate the probability of safety failure of vehicle's optical systems. However, they applied it on Fault Tree Analysis (FTA) modelling chain of events. The tree is based on a preliminary HARA (Duran et al., 2013).

Other propositions for CPS's safety analysis use the Model-Based System Engineering (MBSE) approach. The deployment of MBSE in the industry is increasing and is considered by the INCOSE as the future of systems engineering (Friedenthal et al., 2007). Mhenni et al. used SysML for an enhanced safety analysis of mechatronic systems. They semi-automatically generated a Failure Modes and Effects Analysis (FMEA) from the SysML Internal Block Diagram (IBD) of the systems. They extended the classical SysML modelling with multi-physical flows which enhanced the initial FMEA and improved the safety analysis of mechatronic systems (Mhenni et al., 2014). The research team extended their proposal in an integrated MBSE-MBSA framework for complex systems design. The framework is based on 4 main tasks conducted synergistically between MBSE and MBSA. With the MBSE approach, the designers use scenarios modelling for (safety) requirements definition. They also model the system's functional topology. FMEA and FTA are then conducted based on this topology. In synergy and iteration with these

tasks, simulations are used for performance and safety assessment (Choley et al., 2016). Yakymets et al. also propose a MBSE-MBSA framework with SysML models converted to the AltaRica language for the automatic generation of FTA and the computation of model checking simulations. The main advantage of their approach is to loop on the SysML models by propagating the FTA results (Yakymets et al., 2013).

3. Safety requirements evolution analysis

In the previous section, the authors argued the interest of studying safety requirements evolution of LTS. On this line, they propose a method based on the MBSE and MBSA approaches to build a descriptive study for the analysis of LTS safety requirements evolution. This research study was conducted in collaboration with the autonomous systems team of the engineering company AKKA Technologies. The method was designed and tested on two R&D projects for the design of a steering by wire system and a platooning system. Its steps are applied in parallel on today's systems and the systems under design for future autonomous LTS.

The MBSE approach was applied with Polarsys open source MBSE software *Capella* and the *Arcadia* method (Roques, 2016). The *Arcadia* method is divided into 4 main engineering levels: Operational Analysis, System Analysis, Logical Architecture and Physical Architecture. The method presented below takes place in the two first levels. The reader can find more detail on *Arcadia*'s overall process in (Roques, 2016). Following is the detailed steps of the safety requirements evolution analysis:

1. Nominal operational scenarios definition with sequence diagrams on an MBSE software

In the Operational Analysis phase of the projects, the designers and the stakeholders define the operational capabilities. Then, sequence diagrams are used to define nominal scenarios describing each capability.

2. Hazard Analysis and Risk Assessment (HARA) with ASIL evaluation of nominal scenarios

A HARA is conducted on the nominal scenarios defined in the previous steps. The operational activities, elements of the scenario's sequence diagram, are analysed individually. The hazards that result from their independent failures are analysed by the engineers and evaluated with the ASIL metric from ISO 26262.

The ASIL of a hazard is calculated from its Severity (S), Exposition time (E) and Controllability (C). ASIL is a scale from A to D and is used to designate the necessary level of safety measures engineers have to take in the design, development and manufacturing of a component or a system. ASIL A represents the lowest level and ASIL D the highest level of required safety measures. Quality Managements (QM) is a level bellow ASIL A and indicates that the hazard will impact the quality of the final product without real safety risks (ISO, 2011).

The Severity, Exposition time and Controllability are defined and scaled as follows:

- Severity (S): the severity of a hazard represents the **gravity** of a failure's consequence within a scenario of interest. It is scaled as follows
  - S0=0: No injuries
  - S1=1: Light and moderate injuries
  - S2=2: Severe and life-threatening injuries (probable survival)
  - S3=3: Life-threatening injuries (uncertain survival), fatal injuries

- Exposition time (E): the exposition time of a hazard represents the **relative time** when the hazard could happen. It doesn't take into account the probability of functional failures
  - E0=0: Improbable
  - E1=1: Very low probability
  - E2=2: Low probability
  - E3=3: Medium probability
  - E4=4: High probability
- Controllability (C): the controllability of a hazard represents the capacity of external systems (example: driver) to control the situation in case it happens:
  - C0=0: Controllable in general
  - C1=1: Simply controllable
  - C2=2: Normally controllable
  - C3=3: Difficult to control or uncontrollable

To observe the possible hazards for the system of interest, the consequences of operational activities failures are analysed: loss of activity during the scenario, intermittent activity, unintended activity activation and wrong activity output or result. The resulting hazards are evaluated on the Severity (S), Exposition time (E) and Controllability (C) scales. Table 1 synthesises the computation of ASIL in ISO 26262.

Table 1. ASIL evaluation from the (S), (E) and (C) scales

		C1	C2	C3
S1	**E1**	QM	QM	QM
	E2	QM	QM	QM
	E3	QM	QM	ASIL A
	E4	QM	ASIL A	ASIL B
S2	**E1**	QM	QM	QM
	E2	QM	QM	ASIL A
	E3	QM	ASIL A	ASIL B
	E4	ASIL A	ASIL B	ASIL C
S3	**E1**	QM	QM	ASIL A
	E2	QM	ASIL A	ASIL B
	E3	ASIL A	ASIL B	ASIL C
	E4	ASIL B	ASIL C	ASIL D

3. System design and functional modelling

After several iterations with the different stakeholders, a set of hazard events is agreed on. The transitions and traceability from the Operational Analysis to the System Analysis is managed by the MBSE software *Capella*. In the System Analysis phase, the functional exchanges and functional architecture of the system are modelled

4. Functional chains modelling

Once the system analysis phase of the project is completed, every system's nominal scenario is decomposed into one or more functional chains. The functional chains describe the needed functional

flows for the execution of the scenario. Functional chains and scenarios are important to build a common base for the comparison between two different systems: today's system and their evolution. They also limit the variation in the functions granularity levels.

5. ***Failure Modes and Effects Analysis (FMEA) of the functions and mapping with the previous ASIL evaluation***

An FMEA is conducted on the functions forming the functional chains. Their effects are mapped with the hazard events of the nominal scenarios. Figure 1 illustrates the first five steps of the method. It shows from the mapping between the system functions (SF) and the hazards that the failure of a single function can cause two different hazards. These two hazards can even be on two different scenarios. In addition, they potentially have two different ASIL. In this case, the function's ASIL would be the highest ASIL of the caused hazards (in the figure's example, it would be ASIL D).

As explained earlier, the exposition time (E) is evaluated with the **relative time** when the hazard could happen, or the function could fail. When a function can cause two hazards, one could think that its **relative time of possible failure will increase**. However, the ASIL is evaluated on hazards considered **independent. As such,** we consider no increase of its ASIL if a function causes multiple hazards.

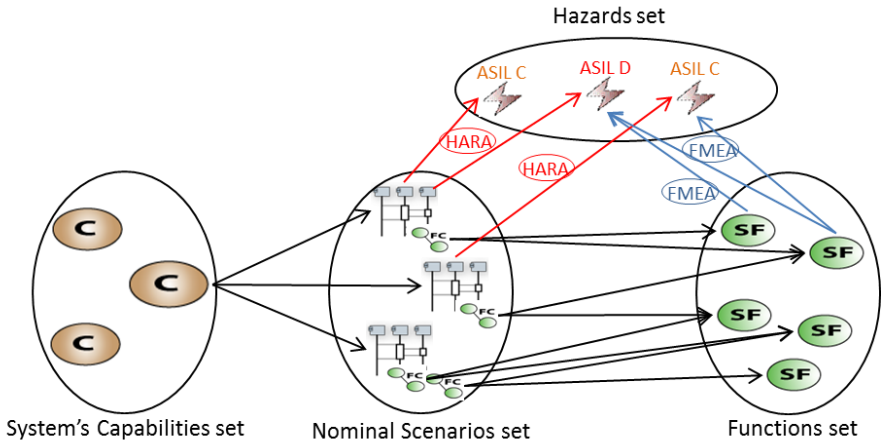

Figure 1. Mapping of the system functions, the nominal scenarios and their hazards

6. ***Comparison of overall ASIL evaluation of both systems***

The final step is to compare the overall ASIL of driving assisted vehicles from the present and the conceptualized future AV. The comparison is done for equivalent operational scenarios, on the base of the functional chains. Three types of differences naturally emerged from the comparison: new functions; disappearance of functions; and a change in the ASIL evaluation of functions. Finally, an analysis of the ASIL evolution is conducted to characterize the safety requirements evolution.

4. Case studies

With the autonomous systems team of AKKA Technologies, the authors observed mainly two types of systems development for autonomous LTS. The first type corresponds to an evolution of existing system

toward higher level or complete automation. The second one is the design of new systems to answer new usages of AVs. In this part, two case studies of both types are presented:

- An evolution of the electrical power-assisted steering system toward a steering by wire then to an absence of steering (vehicle wheels controlled directly with from on board calculator)
- A platooning system composed of a lead vehicle driven by a human operator and 4 AVs

4.1. Steering by wire & Automated steering

Currently, AKKA Technologies is working on its second AV prototype, the Link&Go 2. For this vehicle, they intend it to be equipped with an advanced steering by wire system. Compared to the first version, the advanced steering by wire system provides road feedbacks. Today's vehicles steering system is ensured with a mechanical system mainly accompanied with an electrical power-assistance. It is composed of an intermediate shaft that mechanically links between both the steering column and the rack assemblies. Actually, The former is composed of a steering wheel, a steering column and potentially the power-assistance motor. In addition to the rack, the rack assembly could also contain a power-assistance motor instead of the steering column assembly. Moreover, the rack controls the direction of the vehicle's wheels.

With a steering by wire system, the link between the steering wheel and the vehicle's wheels rotation is ensured with an electrical connection. As the rotational torque in not transmitted with a mechanical link, motors on both sides are needed to replace this function. Two popular architectures for the wheel rotation are: A unique motor on the rack to rotate the vehicle's front wheels at the same time, or to remove the rack and rotate them independently. The current Link&Go prototype has one motor on each of the front vehicle wheel for a rotation without a rack.

The team developed the HARA of the systems based on the operational analysis through iterations. After the transition to the system analysis the team ended up with 4 functional chains: Initialize and align direction system; Block vehicle's wheel rotation when idle; Turning the steering wheel; and providing road feedback. With these functional chains analysed, the first difference observed is an increase of the functions number from present system to the steering by wire one. The number of functions drops back for a fully automated steering system as shows the Table 2:

Table 2. Number of functions in the different steering systems

	the electrical power-assisted steering system	Steering by wire system	Automated steering
Number of functions	10	21	11

This observation indicates that it is more complex to design a hybrid system where two concepts -the assisted steering and the automated steering- co-exist in the form of the steering by wire system. In fact, both systems share a few function such as "Apply the direction command on the road wheels" or "Acquire road wheel information". However, for the co-existence of both systems, specific functions need to be added such as "Acquire steering wheel angle" and "Transmit and display alignment notification".

The FMEA and mapping of the functions with hazard events emphasised a second result corresponding to the safety requirements evolution. It is possible to observe from Table 3 an increase of the ASIL level in functions shared between the systems. In addition, new functions appear with high ASIL.

Table 3. Extract of the evolution of functions ASIL in the steering system

Functions	Assisted-steering				Steering by wire				Automated Steering			
Scales	**S**	**E**	**C**	**ASIL**	**S**	**E**	**C**	**ASIL**	**S**	**E**	**C**	**ASIL**
Monitor vehicle's wheels component	2	4	2	B	2	4	3	C	2	4	3	C
Apply the direction command	2	4	2	B	3	4	3	D	3	4	3	D
Acquire vehicle's wheel information	2	4	2	B	3	4	3	D	3	4	3	D
Extract road information from filtered wheel angle	-	-	-	-	3	4	3	D	3	4	3	D
Generate road feedback	3	4	3	D	3	4	3	D	-	-	-	-

The number of identified ASIL D functions increases with the automation of the steering system, with 4 ASIL D functions against 1 ASIL D function for the assisted steering system: "Generate Road Feedback". Unsurprisingly, it is even higher with the hybrid steering by wire system with 6 ASIL D functions. These conclusions are discussed more in the fifth section of the paper.

4.2. Platooning system

The platooning system, illustrated in the Figure 2, is one of the new usages of the AV creating a considerable amount of interest. It can take numerous possible forms such as a pre-fixed platoon or dynamically cooperating vehicles with the same common interests. It also presents several economic advantages such as the lowering of fuel consumption and of goods transportation cost. A considerable amount of work was undergone for the development and introduction of the platooning system on response to this increasing interest (Jia et al., 2016).

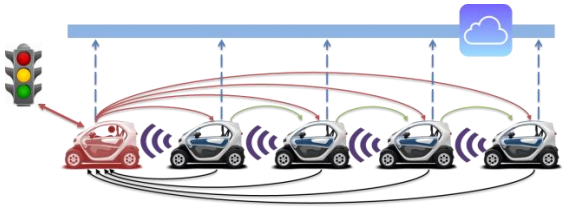

Figure 2. Illustration of the platooning system

AKKA's AV team is also developing a platooning system in collaboration with other companies for the integration on a test platform and the development of the communication network. The system is composed of a lead vehicle, driven by a trained operator, and followed by 4 AVs. To limit the risk and severity of injuries, the maximum speed of the platoon is 30 km/h. There is a platooning component in each of the 5 vehicles managing the integrity of the platoon. The system turned out to be quite complex because of a great number of interactions. Using the *Arcadia* method and a number of iteration, the operational team could manage this kind of complexity and apply the HARA.

As the platooning system is novel and answers to a new usage, one can identify many new interactions and functions involved. At the system analysis level, 4 functional chains for the nominal scenarios of the platooning were identified. The scenario "*Drive in normal platooning mode*" is composed of 3 functional chains: Transmit driver commands; Collect and calculate data for platooning; Transmit own data to other vehicles. The second scenario is the emergency stop procedure and is composed of 1 function chain. The block diagram in Figure 3 displays the exchange flow of the 4 functional chains, between the platooning system, the vehicle's platform, the connexion network and the other entities.

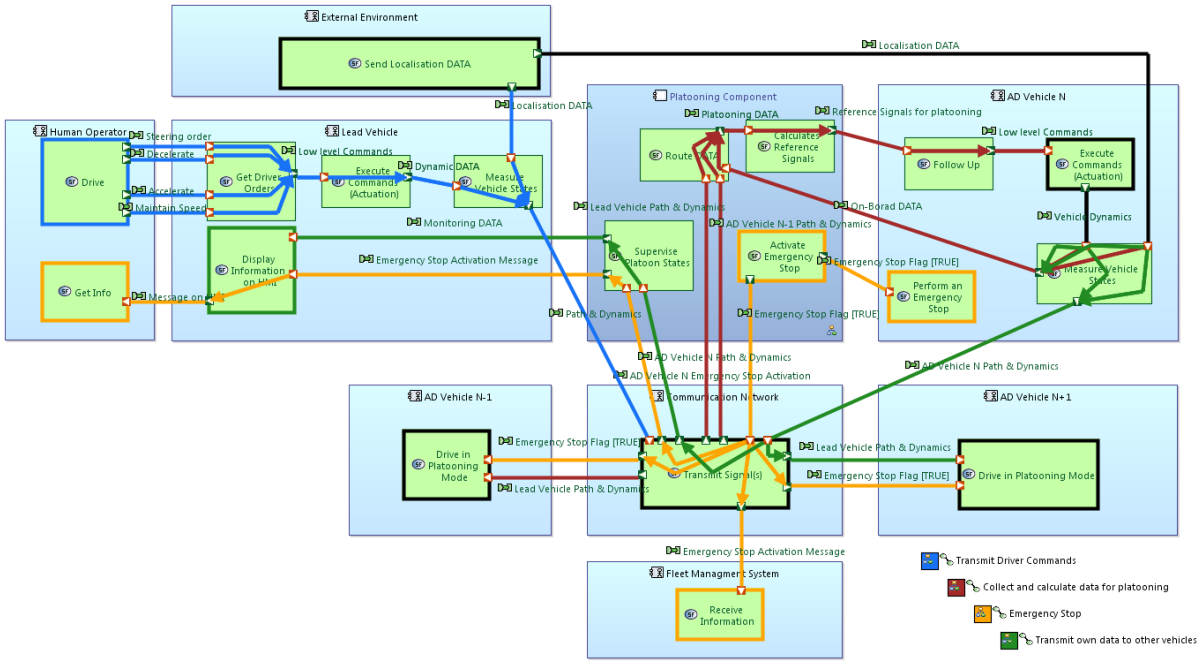

Figure 3. Functional chains block diagram of the platooning nominal scenarios

It can be noticed from the diagram that some of the functions are bottlenecks in most of the functional chains such as "Transmit signal" and "measure vehicle states". "Transmit signal" is an example of a function involved in more than one hazard event. The loss of the function could lead to the 2 scenarios of Table 4.

Table 4. implication of the "Transmit signal" function with 2 hazard events

Function	Hazard Event	Undesirable Effect	S	E	C	ASIL
Transmit signal	Vehicles On-Board Measurement and/or orders are not available for follower vehicles	Crazy behaviours of vehicles followers lead to collisions with Vulnerable Road Users (VRU)	2	4	3	C
	No transmission of an emergency stop order to the following vehicle	Impossibility of avoiding collisions with front vehicle	1	4	2	A

In this case, ASIL C is attributed to the function. As explained in the fifth point of the analysis method, ASIL levels don't stack. The events and their effects are considered independent. The authors considered the platooning system as an evolution of the present vehicle. They compared the functional chains of the platooning system, both in the lead vehicle and follower ones, with existing ADAS functions. It resulted in new function with high ASIL level. For a relevant comparison, the safety evaluation of the present vehicle is done in the same operational condition: with a maximum speed of 30 km/h. It resulted in 2 new ASIL D functions. The risk of a collision with VRU in a sandwich effect is the cause of the high ASIL. In addition, 6 ASIL C functions were obtained. 3 of these functions were new and 3 had their ASIL increased. Table 5 provides an example.

Table 5. Example of functions ASIL level evolution with the platooning system

Functions	Present vehicle				Platoon Lead vehicle				Platoon Follower vehicles			
Scales	**S**	**E**	**C**	**ASIL**	**S**	**E**	**C**	**ASIL**	**S**	**E**	**C**	**ASIL**
Measure vehicle state	2	4	1	A	2	4	3	C	2	4	3	C
Display information on HMI	2	3	3	A	2	4	3	C	-	-	-	-
Transmit signal	-	-	-	-	2	4	3	C	2	4	3	C
Activate emergency stop	-	-	-	-	-	-	-	-	3	4	3	D

5. Discussion and future perspectives

The method proposed in this paper serves to build a descriptive study for the analysis of Land Transportation Systems (LTS) safety requirements evolution. It is based on the MBSE and MBSA approaches inspired from previous framework proposition (Choley et al., 2016; Mhenni et al., 2014). Additionally to FMEA, the method extends their work by using classical tools of CPS safety analysis: scenario modelling and HARA. However, it doesn't compute failure events' chains with FTA or probabilistic tools such as Bayesian network analysis.

The contribution within this method is the quantitative analysis of safety-criticality evolution of LTS using the standard ASIL metric. To the best knowledge of the authors, there is no such analysis in the literature. The authors use previous works in the CPS safety analysis domain to imitate Szarata and Dźwigoń's quantitative prediction without data on AVs safety criticality (Szarata and Dźwigoń, 2015). They use the functional chain modelling to ensure a common base of comparison for the different systems and limit the variation in the functions granularity levels. Furthermore, with the comparison of the function's ASIL and the analysis of its evolution, they provide a first insight to the evolution of the safety requirements of LTS. However, this evaluation is based on the final architecture choice of the designers. As such, the ASIL evolution of the two compared systems functions during the design phase cannot be expressed with this method. It could be improved using metrics for requirements evolution such as the RMI and its extensions, the RSI and HSMI (Anderson and Felici, 2002).

The application of the method on two evolution types of the LTS results in quantifying the way safety will impact their design process evolution. In fact, the evolution of ASIL from today's to the future systems shows two main points. First, if hybrid systems were to be developed for the transition from current LTS to the AV, many functions will emerge for the co-existence of both systems. These functions will bring a layer of complexity to the system as well as high levels of safety criticality. Second, in the new operational context of AVs, the old functions will have their ASIL increase. The reuse of such functions in AVs, with minimum modifications, will be difficult. It will at least require a change propagation analysis and bring integration issues.

On future perspective, the method should be tested in other domains with the appropriate metrics for safety evaluation. The authors think that it could be extended to R&D projects in which there is a focus on requirements engineering for the design of safety-critical systems with low technical maturity. Autonomous Vehicles would be an adequate case study for this research problematic.

Acknowledgement

The authors thank the AKKA Technologies Autonomous Vehicle team for providing the data, insight and expertise that greatly assisted the research project.

References

Anderson, S., Felici, M., (2002). Quantitative aspects of requirements evolution, in: Proceedings 26th Annual International Computer Software and Applications. Presented at the Proceedings 26th Annual International Computer Software and Applications, pp. 27–32. https://doi.org/10.1109/CMPSAC.2002.1044529

Bürger, J., Strüber, D., Gärtner, S., Ruhroth, T., Jürjens, J., Schneider, K., (2018). A framework for semi-automated co-evolution of security knowledge and system models. Journal of Systems and Software 139, 142–160. https://doi.org/10.1016/j.jss.2018.02.003

Choley, J.-Y., Mhenni, F., Nguyen, N., Baklouti, A., (2016). Topology-based Safety Analysis for Safety Critical CPS. Procedia Computer Science 95, 32–39. https://doi.org/10.1016/j.procs.2016.09.290

Dokic, J., Müller, B., Meyer, G., 2015. European roadmap smart systems for automated driving. European Technology Platform on Smart Systems Integration (EPoSS).

Duran, D.R., Robinson, E., Kornecki, A.J., Zalewski, J., (2013). Safety analysis of Autonomous Ground Vehicle optical systems: Bayesian belief networks approach, in: 2013 Federated Conference on Computer Science and Information Systems. Presented at the 2013 Federated Conference on Computer Science and Information Systems, pp. 1419–1425.

Friedenthal, S., Griego, R., Sampson, M., (2007). INCOSE model based systems engineering (MBSE) initiative, in: INCOSE 2007 Symposium.

Gerdes, S., Soliman, M., Riebisch, M., (2015). Decision buddy: tool support for constraint-based design decisions during system evolution, in: 2015 1st International Workshop on Future of Software Architecture Design Assistants (FoSADA). Presented at the 2015 1st International Workshop on Future of Software Architecture Design Assistants (FoSADA), pp. 1–6. https://doi.org/10.1145/2751491.2751495

Hoang, X.-L., Marks, P., Weyrich, M., Fay, A., (2017). Modeling of interdependencies between products, processes and resources to support the evolution of mechatronic systems. IFAC-PapersOnLine 50, 4348–4353. https://doi.org/10.1016/j.ifacol.2017.08.873

ISO, (International Organization for Standardization), (2011). ISO 26262-1:2011 Road vehicles – functional safety – Part 1: Vocabulary.

Izosimov, V., Levholt, E., (2015). Mixed Criticality Metric for Safety-Critical Cyber-Physical Systems on Multi-Core Architectures. methods (physical separation, different power supplies) 2, 8.

Jia, D., Lu, K., Wang, J., Zhang, X., Shen, X., (2016). A Survey on Platoon-Based Vehicular Cyber-Physical Systems. IEEE Communications Surveys & Tutorials 18, 263–284. https://doi.org/10.1109/COMST.2015.2410831

Lutz, R.R., Mikulski, I.C., (2003). Operational anomalies as a cause of safety-critical requirements evolution. Journal of Systems and Software 65, 155–161. https://doi.org/10.1016/S0164-1212(02)00057-2

Matsubara, M., Aoyama, M., (2017). An Analysis Method of Safety Requirements for Automotive Software Systems, in: 2017 24th Asia-Pacific Software Engineering Conference (APSEC). Presented at the 2017 24th Asia-Pacific Software Engineering Conference (APSEC), pp. 408–416. https://doi.org/10.1109/APSEC.2017.47

Mhenni, F., Choley, J.Y., Nguyen, N., (2014). Extended mechatronic systems architecture modeling with SysML for enhanced safety analysis, in: 2014 IEEE International Systems Conference Proceedings. Presented at the 2014 IEEE International Systems Conference Proceedings, pp. 378–382. https://doi.org/10.1109/SysCon.2014.6819284

Roques, P., (2016). MBSE with the ARCADIA Method and the Capella Tool, in: 8th European Congress on Embedded Real Time Software and Systems (ERTS 2016). Toulouse, France.

Savransky, S.D., (2000). Engineering of creativity: Introduction to TRIZ methodology of inventive problem solving. CRC Press. https://doi.org/10.1201/9781420038958

Szarata, A., Dźwigoń, W., (2015). Modelling of road safety attributes in the case of road network development on regional level in Poland, in: 2015 International Conference on Transportation Information and Safety (ICTIS). Presented at the 2015 International Conference on Transportation Information and Safety (ICTIS), pp. 177–183. https://doi.org/10.1109/ICTIS.2015.7232131

Yakymets, N., Dhouib, S., Jaber, H., Lanusse, A., (2013). Model-driven safety assessment of robotic systems, in: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems. Presented at the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1137–1142. https://doi.org/10.1109/IROS.2013.6696493

Youssef Damak, PhD Student
CentraleSupélec, Laboratoire Génie Industriel
642 Avenue Roger Salengro, 92370 Chaville, France
Email: youssef.damak@centralesupelec.fr