MBSE and Safety Lifecycle of AI-enabled systems in transportation

Anton Korolev, Oleg Kirovskii

Abstract — Recently, many autonomous driving companies announced reducing or even closing their business. Even though billions of dollars were invested into autonomous driving in the last decade, there is still no commercially viable autonomous vehicle capable of using public roads.

One of the biggest hurdles that prevent AI-enabled systems from achieving their target functionality is the lack of provable safety. This lack is rooted in the methods that are used in developing autonomous systems. Those methods are agile, in the most cases there are no clear process descriptions. As a result, no complete system description is created. This in turn makes safety analysis and creation of a structured safety argument impossible.

In this article we analyze models and methods for ensuring safety of AI-enabled systems in transportation. We define the task for ensuring safety applicable to any level of driving automation according to the SAE J3016 standard. One of the goals of this publication is to characterize the changes in safety lifecycle depending on the autonomy level.

***Keywords* — System Safety, Functional Safety, Safety Engineering, Autonomy Levels, Safety Lifecycle**

I. Introduction

Recently, many autonomous driving companies (such as Argo AI, Motional, Embark Trucks) announced reducing or even closing their business. According to the various analytical reports, in USA and China the investments into development of autonomous intellectual transport vehicles were at the peak in 2017-2019 [1]. At the same time, there was an increase in scientific publications on this topic [2]. However, since 2018 people worldwide began expressing lesser trust in self-driving cars [3,4]. There is also decline in investments in autonomous driving industry, up to the level that indicates decrease compared to the previous years, as seen in the case with China [1].

Initially, the creators of the self-driving cars considered its greater safety (compared to cars with human driver) to be a noticeable advantage. However, by this time, this goal has not yet been achieved, and autonomous vehicles are not commonly used. Lesser trust in driverless vehicles is directly linked to its insufficient safety.

Automated driving systems are commonly divided into driving automation levels. The globally recognized classification is set in the standard SAE J3016 [5]. It defines 6 levels of driving automation depending on the distribution of the driving processes between the automatic control system (ACS) and the human operator (the word “driver” is not applicable for the self-driving vehicle). Levels L0-L3 are considered non-autonomous, so reacting to fallback situations is the responsibility of the human operator. For autonomous levels (L4-L5) ACS can fully operate the vehicle in any road conditions, including recognizing objects and events on the road, reacting to those, and ensuring safety in case of emergency.

Thus, on higher automation levels, the safety aspect becomes essential. Without the proof of safety, those systems cannot be trusted, and cannot be utilized on public roads.

Standard ISO 26262 [6] gives the following definition of the safety case: “argument that *functional safety* (3.67) is achieved for *items* (3.84), or *elements* (3.41), and satisfied by evidence compiled from *work products* (3.185) of activities during development”.

Currently, there are several international standards that describe the safety process for various aspects of road vehicles safety. These include the following documents:

1. ISO 26262 “Road vehicles — Functional safety”.
2. ISO 21448 “Road vehicles — Safety of the intended functionality”.
3. ISO 21434 “Cybersecurity for Road Vehicles”.

These standards suggest that system safety can be achieved by performing certain work packages throughout the whole system lifecycle. Despite the similar approach, work packages themselves and requirements on them differ from one standard to the other [7].

One of the main problems for applying those standards comes from the impossibility of utilizing the same safety case process for systems that belong to different driving automation levels. Thereby, the complexity of developing vehicle ACS in-creases by 3-5 times compared to developing similarly complex systems of general purpose (i.e. not having specific safety requirements). Besides, a fixed set of work packages and products matches poorly with current agile development approaches. In this article, we suggest solving this problem by, first, applying the principles of system engineering and life cycle (LC) approach to creating autonomous systems, particularly w.r.t the safety case. Second, we suggest modifications to the system’s LC model in order to create safety case depending on the system’s autonomy level.

II. Safety and Modern Engineered Systems

A. Safety as a System Characteristic

Safety is defined as absence of unreasonable risk [8]. This definition contains parts that need to be defined separately.

Статья получена 31 мая 2023.
О.М. Кировский, ассистент РТУ МИРЭА, oleg.kirovskii@gmail.com
А.С. Королев, доцент НИЯУ МИФИ, заведующий кафедрой РТУ МИРЭА, askorolev@mephi.ru

Risk is defined as combination of the probability of occurrence of harm and the severity of that harm [6]. Unreasonable risk is a risk judged to be unacceptable in a certain context according to valid societal moral concepts [6].

The list of definitions presented above can be interpreted as follows. Any action (and inaction) of a human being is linked to some risk. We are used to that and tend to accept risks of our daily lives. The task of safety engineering, however is to help engineer systems that do not bring unacceptable risk, i.e. the risk related to use of those systems does not exceed the acceptable risk level.

Now we can formulate the task for safety engineering research. Our goal is to create a system for which we can prove that the risk related to this system does not exceed acceptable level for any relevant use case (incl. proper use as well as intentional and unintentional misuse, see chapter 2.3 below). In this work, we develop a system engineering method to achieve the goal defined above.

B. Dynamic Aspect of Modern Systems

Modern engineered systems are commonly perceived not as a technical artifact, but as complex sociotechnical systems that influence processes in human societies. Moreover, this influence can come on all stages of LC of engineered systems, from the early stages of creating the concept to the phase of decommissioning. The complexity of such systems comes not only from technical difficulties (which include structural and behavioral complexity), but also from specifics of human factors and being a part of complex-organized processes that such system is involved in. One of essential features of engineered systems is the fact that conditions and configurations of these systems, such as their elements, features and interconnections, are not al-ways clearly specified and volatile. So, this aspect of a system is dynamic. As an example, consider changes in the way vehicles move on the roads adjusting to traffic, in usage of electricity and communication networks depending on the number of users and their needs at the time, and so on.

Because modern systems are complex and integrated in society activities, their development needs to pursue not only functional parameters that intend to increase effectiveness and efficiency for the system itself, but also achieving non-functional features, so called “ilities”, such as portability, reliability, availability, maintainability, etc. In the beginning of 20 century a shift has occurred in the understanding of systems: before those times, technology was understood as a set of separate artifacts, while later the understanding changed to underline complexity and interconnected-ness of technical systems. Before these “ilities” were just a few, and now they total to a few tens. Different types of systems have different essential ilities. Same as any high-risk system, safety is the essential feature for automated driving systems, it is often combined with three other itilies: reliability, availability, maintainability (abbreviates together as RAMS).

C. Safety Aspects

In engineering, it is convenient to group safety concerns in accordance with the potential source of danger. For example, safety engineering includes branches that focus on fire safety, electrical safety, nuclear safety. To secure system safety, each of those branches has its own standards. In the case of complex systems that may pose risks not due to a specific source, but caused by pure malfunctioning, modern standards define three types of safety: functional safety, cyber-security and safety of the intended functionality (see Fig. 1).

The increase in autonomy system level leads to increased difficulty of operating and maintaining the system. The process of ensuring safety becomes more complicated as well. Moreover, some types of systems and components do not have a generally accepted safety case method. For example, this includes machine vision systems that are based on statistical analysis such as neural networks, Bayesian networks, etc. Thus, such systems cannot be easily implemented in widespread use projects due to safety, reliability and stability issues.

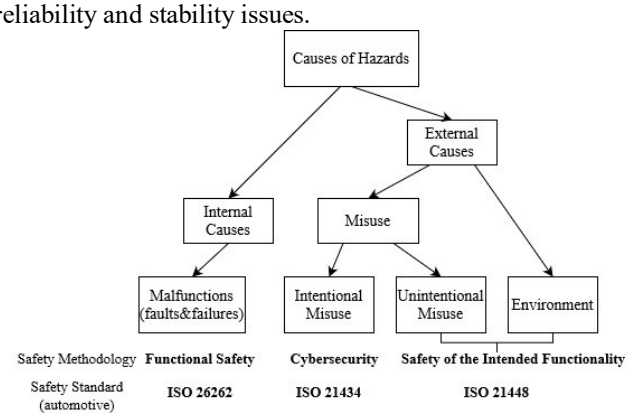

Fig. 1. Safety methodology: use cases [7]

D. Autonomous Vehicles Safety Capability Measure

Takeover. Besides the autonomy levels description, SAE J3016 tries to describe terms such as “autonomous”, “self-driving”, “driverless”, “unmanned”, “robotic”, that are often perceived as jargon in professional community. Meanwhile, it is important to define a common understanding of the term for “autonomous”, because lack of exactness in definition greatly affects our attempts to describe functional capabilities of the vehicle, and therefore, safety requirements for the system.

SAE guidelines highlight the fact that the term “autonomous” is both popular and ambiguous and suggest not using it to describe driving automation. Some of current variants of using this term in describing the technical systems are given in the overview [9].

In systems that belong to levels L0-L2 the human driver is fully responsible for controlling the car trajectory, detecting and reacting on the important objects on the road. Therefore, the system’s role in preventing safety issues reduces to mitigation of the risks in case of malfunctioning. In this case, the safety analysis focuses only on selected adverse scenarios. In systems of the level L3 the human driver is responsible only for acting in case of emergency (“fallback operation”). In systems of levels L4-L5 the human driver is not supposed to intervene at all, as the system should be able to guarantee safety autonomously. This leads to an increased number of possible road situation scenarios and requirements needed for precise description of the system, which inevitably leads to using statistical analysis methods. The task of ensuring safety gradually passes from a human driver to systems.

However, even for systems within the same autonomy level, the human driver’s situational awareness (meaning the driver’s perception of the road situation must be equal to the actual road situation) still plays a significant role in his ability to ensure the safe driving process. For example, we

compare two level 1 ADAS functions: automated emergency braking (AEB) and adaptive cruise control (ACC). When using AEB, the human driver maintains situational awareness about the vehicle's longitudinal and lateral movement at any time, and the system intervenes only in case of emergency that the driver failed to handle. The driver controls road conditions including static and dynamic objects on the road, possible trajectory of other vehicles in the traffic and the vehicle's own intended trajectory. In this case, we the most important safety characteristic is the number of interventions by the autonomous systems, because it happens only when the systems detect parameters that exceed the estimated safety requirements (for AEB this is estimated time to collision with the object in front of the car). While using ACC, the driver loses situational awareness, because the vehicle automatically maintains steady speed and distance to vehicles in the front. When ACC is turned off, the driver needs some time to evaluate current speed and his vehicle's position with other vehicles on the road, before he is able to operate safely. In this case, we can use the number of takeover requests (i.e., how many times the driver requests to take control over steering from ACC) as a safety characteristic.

Proving safety for autonomous vehicles on levels up to L3 also requires assessing takeover scenarios, because these impact the safety of automated vehicles. The system must support effective takeover capability to a reasonable extent during transition to support controllability for humans after takeover situations. In addition, longterm effects of prolonged use of an automated driving system may also desensitize the driver's situational awareness.

Behavioral complexity. Another concern for safety is that the increased amount of possible road scenarios that we need to analyze leads to more complexity in the analysis. So, the number of different combinations of input parameters for systems on levels L3-L4 significantly exceeds the number for systems on levels L0-L2. Since different system operating modes (for example, autonomous driving mode and ADAS) require different data, we can conclude that different system configurations correspond to different sets of scenarios. Safety assessments must be conducted for each system mode.

In addition, we must consider entirely new scenarios that we have not encountered and examined before. These can appear as a result of changes in the surrounding world, such as new road signs or new traffic patterns, or because of new scenarios of interactions between road users and autonomous vehicles, for example, trailing (the case when self-driving car equipped with ADAs system follows another self-driving car, so the safety of both vehicles depends on its ACS systems). All these scenarios also must be included in the safety case (V&V).

Some autonomous vehicles elements can be implemented in machine learning algorithms. This will require new safety case methods, because of the non-deterministic behavior of machine learning algorithms. Besides, machine learning based components and systems cannot be decomposed and must be tested as a black box, which requires a statistical approach to scenarios analysis.

III. Safety Process for AI-enabled systems

A. Systems Engineering for Safety

Systems Engineering (SE) and the Life Cycle Approach (LCA) belong to the SE methodology. It offers methodological basis for successful artificial systems, defining the main criterion for success as achieving a balance between interests of all stakeholders [10,11].

There are more than twenty SE processes, or methods, each containing several repetitive actions needed to build a system. The iterative way of application of SE process at each stage of the system Life Cycle (LC) allows to reduce the cost of the complete life cycle, reduce the time of system creation, as well as achieve other competitive advantages [12].

When creating systems using principles and methods of system engineering, its functions and non-functional parameters, as well as its numerical indicators are developed according to requirements and system architecture. In requirements, they appear in operational capabilities, and pass to entire system level, logical architecture level and physical architecture level. During this process, requirements get decomposed, and new requirements are defined and distributed to system's architectural components. The most admissible architectural components are then selected for trade-off analysis. Currently, the described algorithm can be implemented using tools of model-oriented system engineering, such as ARCADIA methodology and Capella software.

Speaking of creating systems in consideration to achieving safety, the first thing to develop would be strategy for managing the system's LC in terms of safety. This includes establishing safety parameters that indicate that the safety goal is achieved and defining the tasks for managing the system LC to ensure that. A common example is the LC model based on V-model that allows to give convenient description of system analysis and assembling process.

The basic approach on safety case for goal function is the approach that conducts highly iterated functional analysis and system developing, including verification and validation processes that prove that goal function achieves the required safety (pic.2). This approach implies that we can define an area of confirmed scenarios in which the system conducts safely, and an area of unknown scenarios in which the system may cause harm.

The approach that is generally recognized today demands both testing of the developed components and the quality audit of the development process.

B. Safety Process according to ISO 26262

The safety case process is often conducted using the system LC managing approach. This includes standard ISO 26262.

According to this standard, the whole LC consists of five phases: concept phase, product development at the system level, product development at hardware level, product development at software level, production and operation. For each phase, the standard formalizes the operational process that ensures the developing system's safety during its operation (see Fig. 2). Also, each process provides safety arguments (information that proves that the system can be considered safe). Combined, these safety arguments summarize into safety case.

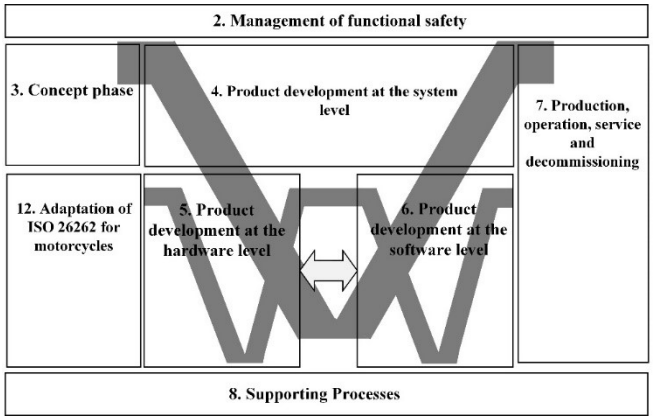

Fig. 2. V-model Safety Lifecycle according to ISO 26262 [6]

Standard ISO 26262 is suitable for developing safe systems for levels L0-L2, but it is not enough for higher autonomy levels. ISO 26262 covers only hazards resulting from failures. It does not cover hazards resulting from insufficiencies in system design, in other words, cases when the system specification does not fully cover ODD.

C. Safety Process according to ISO 21448

Standard ISO 21448 [13] describes the process of ensuring safety of goal function. In other words, it answers the question "how does the incongruity between specification and operational design domain (ODD) affect the safety of utilizing the certain system function (often intellectual)?". ISO 21448 describes three stages of the process:

- identification and evaluation of triggering events, i.e., specific conditions outside the system that possibly lead to a hazardous event;
- evaluation of risks for known hazardous scenarios;
- evaluation of risks for unknown hazardous scenarios.

The final edition of standard ISO 21448 was published in 2022. It can be applied to a system of any autonomy level and defines subjects such as validation, specification of the road situation scenarios, ensuring safety for systems with non-determined algorithms. Despite the additional chapters in the latest edition (in comparison to earlier ISO PAS 21448, published in 2019), the description of safety ensuring process for autonomy levels L3 and L4 is still incomplete.

To summarize, we can conclude that safety requirements for systems on different automation levels are very diverse. So, at this moment, there is no complete answer to the question "Which measures should we take and which methods to use to ensure vehicle safety on every autonomy level"?

IV. Safety Lifecycle Model for Different Levels of Autonomy

An important step in system safety process is to verify and validate that system solutions meet the requirements defined by system safety capabilities. Safety case requires validation that is based on statistical estimate that ensures safety for all known and unknown scenarios with sufficient confidence. Scenarios for this validation can either be directly created in physical environment and allow controlled testing procedures or happen spontaneously while utilizing the system.

Another prospective option is to analyze these scenarios in a virtual environment (which is generated for a previously created and analyzed system) using model-oriented system engineering approach (see Fig. 3).

According to LC V-model, the first step is decomposing of system solutions that develops functions and defines non-functional system features, including safety. The right branch in the model demonstrates integration of developed system solutions with verification of its functioning and integrity of its non-functional characteristics. Model-oriented systems engineering ARCADIA method enables embedding requirements into any system solution on every LC stage. This allows us to conduct work on safety case using models, which is a more efficient approach.

With ARCADIA method, we develop system solutions on five architecture levels: Operational Analysis, System Needs Analysis, Logical Architecture, Physical Architecture, EPBS. For each level, we specify system functioning parameters and other relevant characteristics depending on our chosen Viewpoint. For instance, safety parameters are examined from Safety Viewpoint.

As an example, let's examine the following mission (M): "Avoid any collisions between the vehicle and other road users".

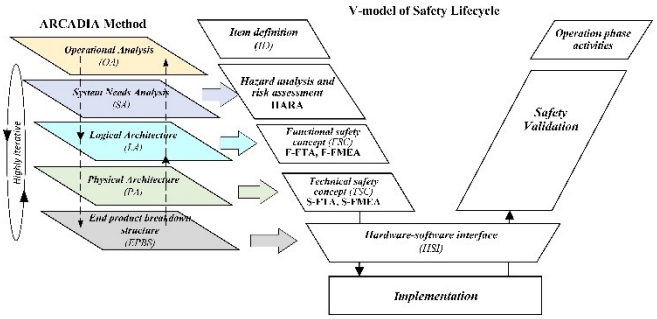

Fig. 3. ARCADIA Method and Safety Lifecycle V-model

Operational capabilities (OC) are:

- Identify approaching objects;
- Ensure acceptable deceleration depending on distance to the object.

Safety requirements for this level may be:

- Ensure accuracy in classifying the detected objects;
- Ensure accuracy of defining the objects depending on distance, weather conditions, time of day;
- Ensure efficient deceleration.

The same level is used to model operational processes and scenarios. For example, operational process of moving in the lane consists of several Operational Activities (OA):

OA1. Maintain the vehicle's motion;
OA2. Perform surveillance over nearby road users;
OA3. Analyze potential collision risks;
OA4. Stop the vehicle's motion.

During the modeling process for following architectural levels, we start with analysis of the system black box model, and then proceed with its logical and physical architecture, and EPBS. By applying this functional analysis, acting gradually and iteratively, we process through actions of flow-down, decomposition, derivement, allocation of functions and non-functional parameters.

As we move to the next system level, from the safety viewpoint, the system may have the following capabilities (SC) and functions (SF):

SC1. Identify approaching objects;

SC1. Determine the distance between the vehicle and the objects;

SC1. Ensure automated braking depending on the distance to the objects.

SF1. Determine the location of an object on the road;

SF2. Perceive relevant objects on the road;

SF3. Predict the future behavior of relevant objects;

SF4. Estimate the risk of collision;

SF5. Create a collision-free driving plan that corresponds with traffic laws and regulations;

SF6. Correctly execute the driving plan;

SF7. Communicate and interact with other (possibly more vulnerable) road users;

SF8. Determine if specified nominal performance is not achieved;

SF9. Perform the deceleration/braking;

SF10. Stop the vehicle's motion;

SF11. Resume the vehicle's motion.

Further on, we can use the same approach to the following steps, such as: developing general requirements for system functions; developing logical components and functions and its requirements; ending process breakdown structure (which is essential for concluding contracts on manufacturing and delivery). We are not giving a full example of this model, because such excessive description does not comply with the intent of this publication.

According to the described model, safety case for operational analysis and system needs analysis requires conduction of hazard analysis and risk assessment (HARA). During the development of logical architecture and physical architecture we need to provide functional safety concept (FSC) and technical safety concept (TSC). Finally, modelling EPBS requires developing hardware-software interface (HIS).

Considering all this, we can perform system functional analysis using model-oriented system engineering tools from the safety viewpoint.

V. Conclusion

In this article, we reviewed the issue of safety case for highly automated vehicles in terms of autonomy level. We examined features of current HAV and defined its limitations, as well as formulated a problem for ensuring safety on system level. We described the mechanism of applying the principles of systems engineering and life cycle process to creating efficient systems regarding safety. We offer a method of adjusting the LC model applicable for any HAV autonomy level.

References

[1] https://www.cbinsights.com/research/autonomous-driving-tech-smart-money-vc-funding/ (accessed 21/06/2023).

[2] Parekh, D.; Poddar, N.; Rajpurkar, A.; Chahal, M.; Kumar, N.; Joshi, G.P.; Cho, W. A Review on Autonomous Vehicles: Progress, Methods and Challenges. Electronics 2022, 11, 2162.

[3] https://www.statista.com/chart/16654/self-driving-cars/ (accessed 21/06/2023).

[4] https://www.carscoops.com/2023/03/trust-in-self-driving-cars-is-falling-while-fear-of-them-rises-study-finds/ (accessed 21/06/2023).

[5] SAE J3016. Surface Vehicle Recommended Practice. Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles. Sep 2016.

[6] ISO 26262:2018. Road vehicles – Functional safety.

[7] Kirovskii, O., Gorelov, V.: Driver assistance systems: analysis, tests and the safety case. ISO 26262 and ISO PAS 21448. In: IOP Conference Series: Materials Science and Engineering, Volume 534, International Automobile Scientific Forum (IASF-2018), Intelligent Transport System Technologies and Components 18–19 October 2018, Moscow, Russian Federation. IOP Publishing Ltd (2018).

[8] IEC Guide 51:2014 Safety Aspects.

[9] Korolev, A., Ryazanov D.: Modern approaches to understanding the autonomy of technical systems. International Journal of Open Information Technologies ISSN: 2307-8162 vol. 10, no. 12, 2022.

[10] INCOSE Systems Engineering Handbook v. 3.2.1/INCOSE–TP–2003–002 – 03.2.1/January 2011.

[11] Benjamin S. Blanchard. System Engineering Management. – Wiley, 2008. – 560 p.

[12] A. Kosjakov, W. Sweet, S. Seymour, S. Biemer. Systems Engineering. Principles and Practice. Second Edition. – Wiley, 2011. – 528 p.

[13] ISO 21448:2022 - Safety of the intended functionality.