

Automotive Engineering in the Age of AI and Software-Defined Vehicles

Milan Z. Bjelica¹ , Bogdan Pavković² , Ivan Kaštelan³ 

¹ University of Novi Sad, Faculty of Technical Sciences, Novi Sad, Serbia

¹milan.bjelica@uns.ac.rs, ²bogdan.pavkovic@uns.ac.rs, ³ivan.kastelan@uns.ac.rs

Abstract – Automotive engineering is undergoing a transition in which software, connectivity and artificial intelligence increasingly determine vehicle functionality throughout the complete product lifecycle. The software-defined vehicle extends development beyond the traditional point of production, as functions, operational knowledge and customer-facing features can continue to evolve after deployment through over-the-air updates. Artificial intelligence adds another dimension to this development. AI is becoming relevant not only for perception in advanced driver assistance and automated driving, but increasingly for decision making and more integrated end-to-end approaches. These developments create significant opportunities for vehicle functionality and customer experience, while at the same time increasing the importance of systematic verification, safety engineering and lifecycle management. This paper provides an editorial perspective on this changing automotive engineering landscape. Particular attention is given to AI safety, verification pipelines, continuous engineering, software-defined vehicle architectures, over-the-air updates and the emerging use of end-to-end AI. The central observation is that these topics are closely connected. Vehicles are becoming continuously evolving cyber-physical products, while safety arguments have traditionally been developed around controlled configurations and well-defined engineering baselines. Future automotive engineering therefore requires development, verification, monitoring and update processes that can preserve engineering evidence across repeated product changes. AI further extends this requirement because its behaviour depends on data, operational conditions and statistical performance in addition to conventional software correctness.

Keywords: automotive engineering, artificial intelligence, AI safety, software-defined vehicle, OTA, verification, continuous integration, autonomous driving, customer experience

1. INTRODUCTION

Automotive development has always combined several engineering disciplines. Mechanical systems, electronics, embedded software, communication networks and control functions have gradually become parts of one tightly integrated product. During the last decade, however, the relative importance of these disciplines has shifted considerably. A growing part of vehicle functionality is determined by software, while connectivity allows that software and the information supporting it to continue evolving after a vehicle has entered operation.

The concept of the software-defined vehicle (SDV) captures much of this transition. The vehicle remains a physical machine with demanding real-time, reliability and safety constraints, but its capabilities are increasingly defined through software architectures, computing platforms, connectivity and services. This changes both engineering and the commercial relationship with the customer. A vehicle delivered today does not necessarily have to provide the same feature set, user experience or operational capability several years later.

Artificial intelligence (AI) accelerates this transition. AI has already become an important technology for perception, including recognition of vehicles, pedestrians, traffic signs, lanes and other elements of the driving environment. Its role is progressively moving toward more complex interpretation, prediction and decision-making tasks. At the same time, research is exploring more integrated architectures in which several conventional processing stages are represented by one learned model.

These developments meet at an important engineering point. Software-defined vehicles encourage frequent evolution. AI itself can evolve through new models, new datasets and new operational evidence. Safety engineering, in contrast, requires confidence that a particular vehicle configuration satisfies its requirements and achieves an acceptable level of risk. Modern automotive engineering therefore needs ways to combine product evolution with controlled engineering evidence.

This paper discusses this situation from a broad automotive engineering perspective. The focus is on the relationship between AI, AI safety and verification, SDV and OTA paradigms, customer experience, continuous engineering and emerging end-to-end AI approaches. The objective is to identify how these developments fit into

one engineering lifecycle and where the main challenges for research and industry currently lie.

2. THE VEHICLE AS A CONTINUOUSLY EVOLVING PRODUCT

The SDV represents more than an increase in software content. Its larger consequence is that the vehicle becomes an evolving product after production.

Traditional vehicle development is naturally described through a V-model. Requirements are refined into system architecture, detailed design and implementation. Verification then proceeds from individual implementation units toward subsystem and complete-system verification, followed by validation of the final product. Configuration management, change management and traceability provide control over this process.

This model remains useful for modern vehicles, although the lifecycle increasingly contains repeated iterations of the V rather than one major development cycle followed by relatively limited maintenance. Software can be modified after release. New functionality can be introduced. Operational information can be refreshed. AI models may be retrained. Changes discovered through fleet operation can return to development and result in another vehicle release.

Over-The-Air (OTA) distribution provides the technical mechanism enabling much of this lifecycle. For autonomous and highly automated systems, OTA can also maintain part of the vehicle's operational knowledge. Environmental models, maps and other reference information can be refreshed without physically accessing each vehicle. This can be understood as continuous maintenance of the vehicle's expected representation of its operating environment.

Stored operational knowledge nevertheless remains only one input to automated driving. The actual environment changes continuously. Construction zones appear; temporary objects enter the road, weather changes and road conditions evolve. Live sensing and perception therefore remain essential even when detailed environmental information is maintained through connectivity.

The same infrastructure has a strong customer-experience dimension. The customer increasingly interacts with a product whose functionality can evolve during ownership. Vehicle features, digital services and user-facing behaviour become part of a continuing relationship between the product, the manufacturer and the supporting cloud ecosystem. This is one reason why SDV development connects engineering with broader consumer experience (CX) strategies.

There is, however, an important difference between updating a customer-facing convenience function and changing behaviour that contributes to vehicle safety. The technical capability for frequent OTA deployment does not imply that every software component can follow the deployment practices familiar from consumer web services. The more strongly an update influences vehicle behaviour, the more engineering evidence must accompany the release.

3. AI AS PART OF THE AUTOMOTIVE SYSTEM

AI safety is easiest to understand when AI is treated as another technology within the complete vehicle rather than as an isolated object.

Classical automotive safety engineering addresses failures in hardware and software through established processes such as those defined by ISO 26262 [1]. Safety of the Intended Functionality, addressed by ISO 21448 [2], extends the discussion toward situations in which no component has failed but the intended functionality becomes insufficient under particular operating conditions. AI introduces further causes of unsuitable behaviour associated with aspects such as model specification, training data, generalization, performance metrics and operating-domain coverage. ISO/PAS 8800 [3] places these concerns into an automotive safety-engineering context.

Consequently, several safety perspectives coexist in an AI-enabled vehicle. Hardware can fail. Conventional software can contain faults. Sensors and algorithms can be functionally insufficient under difficult conditions. An AI model can perform poorly within a particular data slice even while its execution platform operates correctly. Connected systems also introduce cybersecurity concerns that interact directly with software-update and safety mechanisms. [4]

The practical engineering task is therefore to determine how an automotive system containing AI can achieve acceptable safety. The established system-level concepts remain relevant: hazards, risk assessment, safety goals, safety requirements, architectural safety measures, verification, validation, traceability and a supporting safety case.

AI adds new evidence to this process. Dataset representativeness must be considered. Performance needs to be evaluated within meaningful portions of the operational design domain (ODD) rather than exclusively through overall averages. Robustness against plausible changes in the input becomes important. Runtime conditions such as computing load, model quantisation, stale data or invalid intermediate values can influence behaviour. Operational monitoring provides another source of evidence once the system is deployed.

Methods for discovering hazards must also evolve. Hazard and Operability (HAZOP)-style analysis can be applied early to functions and requirements, including deviations related to inputs, outputs, data and operating conditions. Once the architecture is sufficiently defined, System-Theoretic Process Analysis (STPA) can examine unsafe control actions and interactions between AI and conventional components. Exploratory AI-specific probes can complement these structured methods by searching for issues such as dataset blind spots, shortcut learning, inconsistency under moderate input transformations, uncertainty problems and unexpected behaviour close to operational boundaries.

None of these activities alone provides sufficient confidence. Their value comes from their combination within the engineering process. [5]

4. VERIFICATION IS BECOMING A PIPELINE

The increasing complexity and update frequency of vehicle software also changes the practical interpretation of verification.

At implementation level, automated checks can already be integrated into continuous integration environments. Software units can be verified whenever relevant implementation changes occur. Integration testing then evaluates interactions between complete subsystems. At vehicle level, simulation and physical testing provide evidence for complete system behaviour. Fault injection is particularly useful for checking whether safety mechanisms correctly recognize abnormal conditions and initiate the expected fallback or safe response.

AI introduces further verification dimensions. The trained model, its training and validation basis, relevant datasets, operating-domain assumptions and performance targets become part of the configuration being verified. A software change may leave an AI model untouched while changing its runtime behaviour. Similarly, a new model may leave the conventional software architecture unchanged while materially changing the behaviour of the vehicle.

For this reason, verification increasingly resembles a pipeline connecting several forms of evidence: *implementation verification → integration verification → scenario and simulation evaluation → vehicle-level testing → safety evaluation → release → operational monitoring*.

Operational evidence can then feed another engineering cycle: *operation → event or insufficiency detection → data acquisition → analysis or retraining → verification → controlled release → operation*.

This resembles continuous software engineering, but the safety implication is important. Continuous integration can run continuously; safety-relevant deployment remains conditional on sufficient evidence.

Traceability becomes particularly valuable in this environment. When a component, model, dataset, requirement or ODD assumption changes, engineers need to identify which safety analyses and verification results may have become invalid. A mature verification pipeline therefore needs more than automated test execution. It needs configuration awareness and links between requirements, hazards, safety measures, implementations and the evidence supporting them.

This can also make verification more efficient. A change with a limited and well-understood impact should not necessarily require repeating every activity ever performed on the vehicle. At the same time, confidence in such selective re-verification depends on strong traceability and reliable impact analysis.

5. OTA AND THE SAFETY CASE

OTA updates make the relationship between software evolution and safety particularly visible.

A safety argument always refers, explicitly or implicitly, to a particular product configuration and a defined operating environment. If an update materially changes the behaviour relevant to that argument, previously collected evidence cannot automatically be assumed to remain valid.

This becomes especially important for AI. Consider a perception model that is retrained after new operational data reveal an insufficiency. The updated model may resolve the original problem and simultaneously behave differently in other parts of the ODD. A successful update should therefore be treated as a new behavioural configuration whose relevant properties must again be demonstrated.

The same principle applies more broadly to software-defined vehicles. OTA infrastructure enables deployment, while engineering governance determines whether deployment is justified. An effective continuous engineering process therefore connects update management with verification and the safety case.

Field monitoring becomes part of this loop. Events such as interventions, fallback activations, repeated low-confidence situations or unexpected task violations can reveal weaknesses that were insufficiently represented during development. Once such a condition has been understood as a repeatable safety-relevant insufficiency, it should become an explicit engineering issue with mitigation and verification rather than remain hidden inside aggregate field statistics.

This creates an interesting convergence between SDV engineering and safety engineering. SDV platforms provide the technical ability to collect evidence and improve deployed systems. Safety processes provide the

discipline needed to decide what may be changed and what evidence is required before that change reaches the fleet.

6. END-TO-END AI AND THE VERIFICATION CHALLENGE

End-to-end AI represents a further architectural direction with important consequences for automotive verification.

A conventional automated-driving architecture can expose relatively clear functional boundaries between perception, localisation, prediction, planning and control. These boundaries provide interfaces that can be specified and tested. They also make it possible to analyse how uncertainty or an upstream error propagates through the system.

A more integrated learned architecture may combine several of these transformations. Such an approach can reduce manually designed interfaces and allow the model to optimize behaviour across a larger portion of the driving task. The corresponding verification problem becomes more strongly associated with the behaviour of the complete function.

This shifts attention toward task-level requirements and observables. The engineering process needs a precise definition of what constitutes correct or unacceptable behaviour under defined operating conditions. Task-level invariants, runtime supervisors, independent monitors and safety-related subsystems become valuable because they can observe relevant behaviour without depending entirely on the internal decomposition of the AI model.

The ODD also becomes increasingly important. A claim that an end-to-end system performs adequately has little meaning without specifying the operational conditions represented by the claim. Performance observed in one environment cannot simply be generalized to another environment with different traffic, illumination, road geometry, temporary obstacles or interaction patterns.

End-to-end models also make exploratory verification important. Scenario-based testing, physically plausible perturbations, meaningful data slices, metamorphic consistency checks and operational monitoring provide different views of system behaviour. The objective is to accumulate complementary evidence rather than depend on one global performance metric.

Architectural safety mechanisms can additionally constrain the AI system. Independent monitoring, safety envelopes, fallback behaviour and restrictions of the allowed ODD can limit the consequences of inappropriate AI outputs. These mechanisms allow safety arguments to remain partly architectural even when the internal AI implementation becomes increasingly integrated.

7. FROM VERIFICATION TO CONTINUOUS SAFETY ENGINEERING

The broader change in automotive engineering can therefore be described as a transition toward continuous evidence management.

A traditional release establishes a baseline. The SDV makes that baseline temporary. AI makes some of its behavioural properties dependent on data and operational experience. OTA enables a new baseline to be distributed rapidly. Field connectivity then provides information about how that baseline behaves in actual operation.

This lifecycle creates a circular relationship between development and operation. Verification is no longer located only near the end of product development. Evidence is generated during requirements engineering, safety analysis, model development, software implementation, simulation, vehicle testing and field operation.

For AI-enabled vehicles, four questions become particularly important for every significant release:

1. What changed?

This includes software, models, datasets, parameters, operational assumptions and supporting environmental information.

2. What behaviour can the change influence?

Traceability and impact analysis need to connect the change to functions, hazards, safety measures and ODD assumptions.

3. What evidence must be regenerated?

Verification should address the affected implementation and the system-level consequences of the change.

4. What should be monitored after deployment?

Residual uncertainty remains, particularly for complex AI behaviour. Runtime and fleet evidence therefore form part of the continuing safety argument.

A useful consequence of this viewpoint is that continuous engineering does not require abandoning the V-model. Each meaningful release can be regarded as another controlled iteration through the relevant portions of

that process. Automation makes such iterations practical, while safety engineering determines their required rigor.

8. DISCUSSION AND RESEARCH OUTLOOK

Several research directions emerge from this combined view of AI, SDVs and continuous automotive engineering.

The first concerns verification scalability. Scenario spaces for automated vehicles are extremely large, and AI adds further variation through data and model behaviour. Verification therefore needs intelligent selection and combination of evidence. Simulation, structured scenarios, AI-specific probes, statistical evaluation, system testing and field monitoring each address different parts of the problem.

The second concerns quantitative interpretation of AI behaviour. Classical vehicle reliability engineering can reason in terms of component failure rates and system reliability models. AI evidence more commonly appears as test results, failed scenarios, interventions or observed behavioural insufficiencies. Creating useful interfaces between these forms of evidence could support architecture-level decisions and risk acceptance without reducing AI safety to a single performance number. Behavioural measures based on clearly defined automation-task violations are one possible direction. [6]-[7]

The third concerns the safety case itself. A continuously evolving vehicle requires a safety argument that can evolve in a controlled manner. Research and industrial tooling therefore need to support incremental evidence, impact analysis and configuration-dependent safety claims. The long-term objective should be a safety case that remains connected to the actual software and AI configuration deployed in the vehicle.

A fourth question concerns architectural evolution. Modular AI architectures provide useful engineering boundaries, while end-to-end models may offer performance and development advantages for increasingly complex driving tasks. Automotive engineering will need practical methods for obtaining sufficient observability, verification and safety control at whichever abstraction level the architecture exposes.

Finally, the relationship between customer experience and engineering assurance deserves attention. OTA makes continuous product improvement visible to customers and can extend the useful functionality of a vehicle throughout its lifetime. The same update infrastructure also becomes part of safety maintenance. Industry therefore must reconcile the commercial desire for rapid evolution with the evidence requirements of a safety-related cyber-physical system.

Risk acceptance remains the final engineering question. Comparative principles such as Globally at Least Equivalent (GAME) can ask whether an automated solution performs at least as safely as an accepted reference, while absolute approaches such as Minimum Endogenous Mortality (MEM) consider the residual contribution of the technical system to risk. Whatever criterion is selected, confidence ultimately depends on a chain of evidence connecting the deployed system to its actual operational behaviour.

9. CONCLUSION

Automotive engineering is moving toward a lifecycle in which the vehicle continues to change after it enters operation. Software-defined architectures, connectivity and OTA updates enable this transition, while AI expands both the capabilities of the vehicle and the complexity of demonstrating its behaviour.

These developments make verification increasingly continuous. Software tests within continuous integration environments form only the first layer. Integration testing, simulation, scenario-based evaluation, vehicle testing, fault injection, AI-specific verification, safety analysis and operational monitoring need to contribute to one connected evidence pipeline.

AI safety fits naturally into established automotive safety engineering when the complete system remains the object of the safety argument. ISO 26262, SOTIF and AI-oriented safety activities address different causes of hazardous behaviour, while traceability and the safety case connect them throughout the development lifecycle.

End-to-end AI may shift some verification from internal component boundaries toward task-level and system-level behaviour, making clear ODDs, runtime monitoring, independent safety mechanisms and systematic behavioural testing increasingly important.

The resulting automotive engineering paradigm is therefore one of continuous product evolution combined with controlled evidence evolution. OTA provides the means to change the vehicle; AI and software provide much of the reason to change it; verification and safety engineering determine when those changes are ready for deployment. Establishing efficient ways to maintain this relationship is likely to remain one of the central engineering challenges of the software-defined and AI-enabled vehicle.

FUNDING

This publication was co-funded by the European Union (Interreg IPA CBC Croatia-Serbia programme, project

CySafe SmartMob, grant number HR-RS00269). Its contents are the sole responsibility of the authors and do not necessarily reflect the views of the European Union.

DECLARATIONS

The authors declare no conflict of interest. The data availability is not applicable.

REFERENCES

- [1] ISO 26262, *Road Vehicles — Functional Safety*.
- [2] ISO 21448, *Road Vehicles — Safety of the Intended Functionality*.
- [3] ISO/PAS 8800, *Road Vehicles — Safety and Artificial Intelligence*.
- [4] ISO/SAE 21434, *Road Vehicles — Cybersecurity Engineering*.
- [5] Automotive SPICE, *Process Assessment Model*.
- [6] M. Z. Bjelica and B. Mrazovac, “Reliability of Self-Driving Cars: When Can We Remove the Safety Driver?,” *IEEE Intelligent Transportation Systems Magazine*, vol. 15, no. 4, pp. 46–54, 2023.
- [7] M. Z. Bjelica, *Reliability Estimation for a Central Vehicle Computer with Upgradeable Software Components*, Master’s Thesis, University of Applied Sciences – FH Campus Wien, 2021.

AUTHOR BIOGRAPHIES



Dr. Bjelica is the Full Professor at the FTN Uni Novi Sad, a Functional Safety Instructor at the University of California San Diego, and the CEO of NIT Institute. He actively participates in research and innovation activities for various computer-engineering sectors, focusing on consultancy and training in the fields of system safety, functional safety, automotive engineering, and consumer electronics. During his career, he consulted companies in the automotive industry (ZF Germany, TTTech Austria, Qualcomm Automotive USA, Daimler Germany) and also other companies in consumer electronics, industrial machinery, and computing domains. He is a frequent participant and a speaker at major industry events worldwide. Dr. Bjelica holds a Ph.D. degree in computer engineering from the University of Novi Sad, Serbia, as well as an Academic Safety Engineer degree from the FH CampusWien - Vienna Institute for Safety and Systems Engineering (Austria). His professional and research focus is on complex system and software architectures with specific interactions and virtualization. He authored over 100 publications across major journals and scientific conferences and holds 30 patents.



Dr. Pavkovic is the Functional Safety Consultant and Instructor at the University of California San Diego; Associate Professor at the Faculty of Technical Sciences - University of Novi Sad; CTO of NIT Institute; and ex Technical Group Lead and Software Architect at the TTTech Automotive. He focuses on application of Functional Safety and SOTIF for software-defined vehicles in automotive production projects for the industry leading OEMs across several design centres in Europe. He was in charge of around 30 people working on cutting edge computer vision and machine learning optimizations for embedded automotive platforms for top-tier OEMs from both camera and lidar domain, with regard to their appropriate design and utilization with regard to safety. He has extensive professional experience in planning, executing, leading, supporting and analysing research projects and technologies in computer engineering, communication technologies and automotive software.



Dr. Kastelan is the Full Professor at the University of Novi Sad, Faculty of Technical Sciences. His teaching and research interests are the design of digital and computer systems, algorithm design, and engineering education. During his research work, he worked on the problem of verification of consumer electronics devices with touch screens, and the development of platforms and teaching materials for the needs of engineering education in the field of computer engineering. He participated in various European, national and provincial projects, and is currently a team coordinator in the project, funded by the Interreg IPA Cross-border Cooperation Programme Croatia-Serbia, which aims to establish living laboratories and develop study materials for courses in the topics of artificial intelligence, functional safety and cybersecurity. He is the author or co-author of more than 10 papers in international and national journals, over 100 papers in conference proceedings, 13 technical solutions, 13 patents, a textbook and a problems book.