Maximilian Eibl, Martin Gaedke. (Hrsg.): INFORMATIK 2017,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2017 1463

Research Challenges for a Future-Proof E/E Architecture

A Project Statement

Stefan Kugele[1], Vadim Cebotari[1], Mario Gleirscher[1], Morteza Hashemi Farzaneh[1], Christoph Segler[2], Sina Shafaei[1], Hans-Jörg Vögel[2], Fridolin Bauer[2], Alois Knoll[1], Diego Marmsoler[1], Hans-Ulrich Michel[2]

Abstract: During the last decades, the functional power and complexity of automotive E/E architectures grew radically and is going to grow further. We identified two key factors namely *autonomy* and *intelligence*. Both pose research challenges for the next generation E/E architecture. We aim to tackle the design challenges with methods and technologies. We propose in this project statement to use a *service-oriented architecture* on top of an in-vehicle communication network based on *time-sensitive networking*. Moreover, a rigor *risk analysis and mitigation* approach enables synthesis of a safety controller. A *learning architecture* facilitates a shift towards *user centralization* by proactively adapting functions according to user profiles. In addition, further functions might need to be learned at *run-time*.

Keywords: Automotive; safety assurance; service-oriented architectures; communication; time-sensitive network; deep learning; machine learning; artificial intelligence

1 Introduction

During the last decades, a large number of software-based functions has been introduced in form of embedded systems. Many of them are characterized as safety-critical such as those found in the avionic, automotive, railway, and industry automation domains. The number of those functions as well as the complexity of involved hardware topologies will grow. For decades, the autopilot in airplanes has been a well-established software-controlled mechatronic system. Shipping and aerospace are further sectors where autopilot features have been heavily used. Next, we describe the state-of-the-art and framework conditions from various perspectives.

Technology: E/E architectures (electric/electronic) can be best characterized as historically grown, mostly federated but sometimes integrated architectures with often pragmatic, cost-efficient, and ad-hoc solutions. The notion of E/E encompass (i) the electrical network (e. g. high-voltage network, power electronics, generator), (ii) the reliable distributed control

[1] Technical University of Munich, Boltzmannstr. 3, 85748 Garching bei München, {vadim.cebotari, mario.gleirscher, morteza.hashemi, stefan.kugele, diego.marmsoler, sina.shafaei}@tum.de, knoll@in.tum.de
[2] BMW Group, 80788 München, {Fridolin.Bauer, Hans.Michel, Christoph.Segler, Hans-Joerg.Voegel}@bmwgroup.com

doi:10.18420/in2017_146

system architecture (CSA) (e. g. sensors, actuators, electronic control units (ECU), bus systems), and (iii) less safety-critical infotainment systems (e. g. components from the consumer electronic industry). We focus on the last two mentioned. Automotive domain separation (e. g. body, chassis, etc.) is still present. However, new ADAS and infotainment functions will break domain boundaries. Thousands of software-controlled functions are realized by a complex interplay of signals with timing requirements sent via heterogeneous bus systems with complex gateway structures connected to purpose-built ECUs with often closed proprietary designs. Over time, there was a shift from a "one function per ECU" paradigm towards a situation where several functions are deployed onto a single ECU and a function is potentially divided into several sub-functions executed on several ECUs. However, this "ECU-" or "signal-based" development approach has reached its limits of mastering complexity **(GC1)**.
Criticality: In automotive software systems, only a few functions pose strongest safety requirements (i. e. ASIL D, cf. ISO 26262 [IS11]). However, along with highly automated or fully automated driving (SAE levels 4 and 5 of driving automation according to [So14]) this situation changes: we are facing a *paradigm change* in that the responsibility of driving a car switches from the driver to the machine. This coincides with a change of safety goals and safety measures because for level 5 there is no driver to fall back to anymore. Hence, *fail-operational* behavior instead of *fail-silent* is E/E-architectural challenge **(GC2)**.

1.1 Project Challenges from our Industrial Collaboration

In this section, we list the **project challenges (PC)** that we derived together with representatives from the German automotive industry: Architectures for vehicular electrics, electronics, communication, and software are facing several challenges driven by strategic trends. These trends—(i) *automated driving*, (ii) *artificial intelligence*, (iii) *drivetrain electrification*, and (iv) *connected systems and services*— require innovations for reducing system complexity, improving verifiability, and enhancing system integration, consequently driving the introduction of new technologies.

Automation introduces a plethora of additional sensors, computationally complex, data-intensive algorithms, generating in-vehicle network load. Furthermore, automation requires safety engineering to a much greater extent than for previous vehicle generations (GC2.**PC1**). Moreover, intelligent algorithms not only help automated vehicles to drive safely, but *artificial intelligence* (AI) will be making its way into many aspects of a functional vehicle design with the advent of intelligent personal assistants (IPA), multi-modal natural user interaction, situational awareness and augmented reality (GC1.**PC2**). Vehicle's self-awareness requirements are driving architectural design, not to the least regarding enhanced service descriptions and flexible organization of information streams (GC1.**PC3**).

Two software *process*-related challenges have to be tackled: "In-housing" of software development or at least provision of verified and OEM-approved services is essential

(GC2.**PC4**). This goes along with a shift from an ECU-centric development towards a *service*-centric development of automotive systems (GC1.**PC5**).

Derived from these challenges, we formulate the overall research question: **How does a future-proof automotive E/E architecture look like with emphasis on safety, automation, and intelligence?**

1.2 Contributions

We contribute a *catalog of research questions* to be tackled by a CSA for autonomous, intelligent vehicles. This catalog spans four *focus areas* (see Sect. 3): **control safety**, **communication (TSN)**, **service-oriented architecture (SOA)**, and **artificial intelligence**.

(1) In the **safety area**, we want to find *safety invariants* of automatic vehicle controllers and *reliability design constraints* to be implemented in the CSA, (2) in the **communication area**, we aim at a *synthesized, efficiently scheduled, and reliable* communication network required for the in-vehicle distribution of control functions, (3) in the **SOA area**, we strive for an *abstract model* of core elements of our CSA in terms of *services, their discovery and execution*, and (4) in the **AI area**, we seek to *optimize adaptive vehicle comfort functions* to be aligned with the safety invariants imposed on the CSA.

We sketch our research plan in each focus area: risk analysis and mitigation, modeling, verification, synthesis (i. e., safety controller, TSN network, "hardened" SOA/TSN-based architecture), and integration. In Sect. 3.5, we indicate how results from the focus areas form a coherent picture. Based on this picture, in Sect. 4, we provide a *question catalog* to be tackled by any research on the challenges posed in Sect. 1.1. We think, this catalog contributes to *validation obligations* of any autonomous road vehicle test and demonstration platform.

1.3 Related Work

Back in 2006, Broy [Br06] postulated a couple of research challenges in automotive software engineering. With SOA and TSN, we think we can address the architecture-related challenges and even refine them as depicted in Tab. 1. Traub et al. [TMB17] also favor the use of SOA-concepts for automotive functions. The RACE [So13] project demonstrated a centralized automotive architecture equipped with redundant Ethernet-based communication channels. With regard to full driving automation, Koopman and Wagner [KW16] outline research challenges. Among them, the technical are: (i) *driver out of loop* in level 5, (ii) *complex requirements* e. g. when using machine learning, (iii) *non-deterministic and statistical algorithms*, (iv) *machine learning systems*, and (v) *mission critical operational requirements*. Within the project, we focus on (i), (ii), (iv), and (v).

Outline The remainder of the paper is structured as follows. Sect. 2 introduces the focus areas. Next, in Sect. 3, we explain the challenges and briefly sketch the envisioned solutions for the selected topics and point to the interrelations between the focus areas. Sect. 4 lists our research questions and finally we conclude in Sect. 5.

2 Project Statement

Derived from the challenges in Sect. 1.1, we identified focus areas for a future-proof automotive E/E architecture, depicted in Fig. 1: (1) integrated *safety* concept, (2) infrastructure for real-time, safety-critical communication (e. g. TSN), (3) service-oriented architecture (SOA) for safety and non-safety-critical service provision, and (4) artificial intelligence (AI) providing e. g. machine learning capabilities.

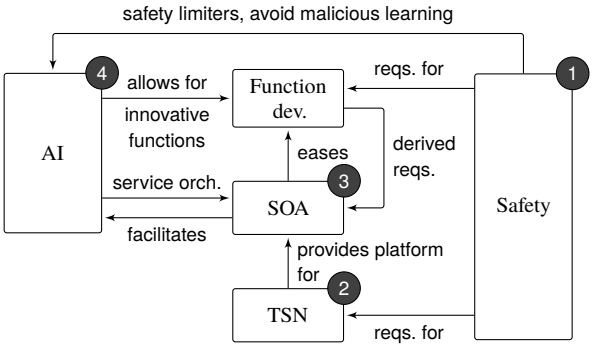

Fig. 1: Focus areas and interrelations.

We are convinced that *function development* effort can be radically reduced by *service-oriented architecture*. It facilitates the use of *machine learning* and other AI approaches, which support service orchestration of innovative functions. Time-sensitive networking provides a communication infrastructure for safety- and real-time-critical service provision. An integrated *safety* concept allows to monitor "learned" behavior, and poses requirements for functions as well as the execution platform.

By taking a two-fold approach we look at (i) the functional breakdown of an automotive system and deriving requirements for our focus areas on the one hand, and at (ii) the common design questions raised across the interfaces between the areas when integrating them to form an automotive system architecture, on the other hand. The focus areas have been chosen to contribute to the challenges in Sect. 1.1: computationally demanding applications (e. g. the learning, proactive intelligent personal assistant); engineering excellence to deliver functionally safe autonomous systems while organizing them for a maximum level of modularity, re-usability, and extendibility; and providing a scalable, yet well-performing and reliable data communication infrastructure. While there are advanced aspects of automotive E/E architectures, these focus areas contribute most to data-driven approaches that rely on the efficient definition, organization, and handling of data streams in the car. Hence, we will look at these focus areas and study mutual dependencies and requirements.

3 Approach

In the following, we briefly discuss our focus areas which we think are of particular importance to leverage *intelligent* and *automated* vehicles.

3.1 Focus Area 1: Vehicle Control System Safety

In this focus area, we approach control safety or *hazard reduction* by two complementary views: (1) We deal with *safety by designing safe system dynamics* (aka "safety of the *intended or nominal function*"): The state-of-the-art, customer expectations, and competition mandates the implementation of function enhancements (e. g. collision avoidance, safety braking) responsible for mitigating hazardous situations typical of an application. (2) We aim for *safety by improving resilience, reliability, and security*: When performing, system functions can exhibit various anomalies (e. g. due to faults, intrusions, attacks, misuse) violating the *intended function*. The state-of-the-art and product liability practices mandate the taking care of avoiding, detecting, and containing such anomalies by improving various properties of a system design. The goals underlying these two views can be expressed in terms of *safety invariants* imposed on the vehicle control loop, for (1) and (2), and *reliability design constraints* to be respected by the CSA, for (2).

Our **objective** from the perspective of (1) and (2) is to *enable highly automated and autonomous vehicles to reach acceptably safe states with respect to the hazards identified in the most relevant operational situations*. This objective includes the design of *fail-operational architectures*, i. e., safe driving behavior in presence of covered faults.

Approach and First Results To address vehicle safety as outlined in (1) and (2), we envisage the following three tasks: *(a) Assess the architecture:* We perform an assessment of the *as-is and to-be* CSAs for highly automated and autonomous driving. *(b) Develop a safety kit:* We collect best practices from a systematic literature review and prepare recommendations for further or improved use of selected practices for the assurance of the envisaged CSAs. In [GK16] we discuss an overview of best practices known in the design of high-assurance embedded control software and systems. *(c) Improve the assurance procedures for the envisaged CSAs:* We develop a method for hazard analysis and risk assessment in the co-design of *vehicle control functions* and of safety measures such as *monitor-actuators*. In [GK17, GK] we study a modeling framework as well as first steps towards such a method.

3.2 Focus Area 2: Communication

An increasing number of functions in automotive E/E architectures introduces further requirements for in-vehicle communication systems such as the capability to support

mixed-criticality and hard real-time requirements. Such a communication system has to offer higher bandwidth (required for e. g. demanding video data transmission $> 1GB/s$) and timing guarantees to fulfill safety-related requirements (e. g. deterministic minimum latency and jitter, reliability, and fail-operational).

Our **objective** is to reduce heterogeneity and configuration complexity of in-vehicle communication sub-systems while supporting mixed-criticality and hard real-time. This yields advantages for the function developer, who is spared the task of network management and can focus more on function development itself.

Approach and First Results Time-Sensitive Networking (TSN) is a new set of standards in the development by the Institute of Electrical and Electronics Engineers (IEEE) which aim to support hard real-time communication based on Ethernet technology. We investigate the capabilities of TSN standards in the automotive domain and develop methods and concepts simplifying its integration, configuration, and verification in a future-proof E/E architecture. To achieve the mentioned objectives and following our previous work in [FSK16] and [FK16], our approach contains the following steps: (i) Investigation of TSN features useful for the automotive domain, (ii) development of a network modeling approach composed of Object-Oriented and Logic Programming paradigms, (iii) development of network facts and inference rules for network configuration and verification, (iv) Time-Aware Shaper schedule synthesis based on e. g. Satisfiability Modulo Theories (SMT), (v) formal timing verification for safety-critical systems, (vi) validation of different network topologies because of different equipped options in the vehicle, (vii) development of a fail-operational communication verifier (expert system) based on IEEE 802.1CB, and (viii) development of a prototypical TSN demonstrator to evaluate the developed concepts.

3.3 Focus Area 3: Service-Oriented Architectures

We think that a shift from traditional ECU-based development towards a service-oriented approach helps to reduce complexity of software development. The use of a service-oriented architecture helps to obtain a clear understanding of the complex functional interplay, which is necessary when aiming at a high or even full level of driving automation. It facilitates reuse of functionality e. g. by defining them as a *Safety Element out of Context* (SEooC). Therefore, services are described as hardware- and technology-agnostic as possible. Moreover, SOA is a means for "learning" by orchestrating services at run-time to provide intelligent functions. A *unification* in how to access any and all data aims at establishing an intuitive way to think about interdependence. A *service registry* facilitates to access (respecting access control mechanisms) data using services.

Our **objective** is to provide a framework for (i) formal service specification, (ii) design, and (iii) implementation of automotive approved service-oriented architectures comprising both in-vehicle as well as back-end components.

Approach and First Results To achieve this objective we envisage the following steps:
Identification and Classification of Services: We are going to work out a *service classification scheme*. One possible classification can be according to their level of hardware independence (in [Ku17], we assigned services to layers). Hardware-dependent services (e. g. sensor and actuator interaction) are fused to and coordinated by higher-layer services.
Service Interface Specification and Service Level Agreements (SLA): There are different standards and tools that can be used to address this topic. Most tools offer a very good support for the specification of syntactic interface. On the other hand, these tools only insufficiently support behavior specification. Due to the challenges in autonomous driving and machine learning, we consider a behavioral interface description as necessary. Moreover, to learn *new* functions by dynamic orchestration of existing services, a detailed *SLA* (also referred to as service contract) is necessary.
Service Bus and its Responsibilities: To design a service-oriented architecture with built-in safety support and in-car intelligence, we have to determine the responsibilities of the service bus as the core of the technical infrastructure for a service-oriented architecture. The service bus provides middleware features on top of TSN (cf. Sect. 3.2) meeting all requirements from safety and AI perspective.

3.4 Focus Area 4: Artificial Intelligence

The next generation of Intelligent Personal Assistant systems are capable of covering a variety of tasks from being a car advisor on different applications, and well-being coach of the occupants, to being a chauffeur for the passengers in absence of the driver for autonomous driving scenarios. The increasing amount of data in car beside the difficulties in hyperparameter tuning for training (learning rate, loss function, number of training iterations, gradient update smoothing, optimizer selection, etc.) and therefore, growing complexity in defining a good reward function in most of the cases, shows the need for scalable machine learning approaches.

Our **objective** is to realize the idea of an intelligent and self-developing car, first of all the context needs to be acquired from different data source in the car and outside the car. The next step is to select the suitable machine learning methods according to the context and the expected outcome, considering the related safety aspects as well, with an acceptable computational load. A challenging part in automotive E/E architectures is to find the suitable place for the computation phase of learning in AI-based functions (e. g. local vs. cloud).

Approach We target two categories of *Context Acquisition and Modeling* besides the *Context Reasoning and Prediction*. In other domains, different architectural concepts for context acquisition have already been developed (see [Pe14]). To acquire context data in an automotive E/E architecture, several issues have to be addressed. One of these is the highly-divided communication infrastructure. There is no central point in the network where all raw data can be collected and processed. Furthermore, the transferred messages are highly optimized leading to uninterpretable data without further information. This issue can

be addressed with a middleware, a central context server, context servers divided in domains (like comfort, motion control, etc.), or even other architectural concepts. Afterwards, the context has to be derived to represent a model of the current situation and linked to create higher-value context information.
After collecting and providing context data the next step is to reason about and learn from the collected data. Techniques of making conclusions from context data consider two factors of performance and human readability. One of the problems is to choose the right data to reason about in combination with learning algorithms. Context reasoning is followed by context prediction and it will bring up architectural challenges which we are going to investigate and address in this project. An evaluation of prediction architectures and algorithms must be performed. After clarifying the limitations and problems, modifications to improve the as-is architecture or proposing a new one may be required. *Neural networks* (NN) along with deep learning methods have shown noticeably better performance in comparison with their ancestors. Use cases such as driver behavior recognition based on deep convolutional NNs [Ya16] and real-time vehicle detection in autonomous driving [Hu15], lend credence to promising future of deep learning in autonomous driving but it increases the complexity.

3.5 Relationships between the Focus Areas

Impacts from Focus Area "Safety" For design practices included in the safety kit, we have to rely on assured properties of the employed in-vehicle communication network (cf. Sect. 3.2). There is a bidirectional relation between TSN and safety. TSN offers standards as services to support safety-related requirements of functions such as timing and reliability. Following the proposed model-driven approach, TSN users have to annotate such critical requirements at design time that fulfill all safety requirements (in particular but not limited to timing) at run-time. For example, TSN guarantees that a synthesized time-triggered schedule at design time is safe against network traffic changes caused by less critical data flow at run-time. Similarly, the availability of fail-operational disjoint paths (defined as safety requirement at design time) and required network bandwidth (for dynamic traffic changes) are guaranteed by TSN. On the other hand, TSN itself has to be safe. Network faults have to be classified and to be considered at network design time to compensate network failures. We aim to use SOA (cf. Sect. 3.3) as a model (i) to represent and analyze functional interactions within a distributed vehicle architecture and (ii) to understand safety-related interaction of vehicle control functions. Finally, we aim to assure safety properties in presence of learning algorithms (cf. Sect. 3.4) interacting with vehicle control functions.

Impacts from Focus Area "Communication" Non-functional requirements of functions (e. g. safety and timing constraints) are defined using a service-oriented paradigm. SOA is the interface between TSN QoS services and higher level functions. We focus on a publisher/subscriber communication paradigm.

In the context of safety, it has to be guaranteed that *learned* functions do not disturb other critical functions at run-time (e. g. because of extra required bandwidth). For dynamically

or static *learned* functions the support of mixed-criticality is significant. TSN guarantees that properties (e. g. bandwidth) of such dynamic functions at run-time do not disturb the safety-critical communication (usually scheduled at design time). On the same physical communication medium (Ethernet), the critical traffic (e. g. time-triggered) is safely separated and the remaining bandwidth is reserved for other functions and their requirements for dynamic network reconfiguration.

Impacts from Focus Area "SOA" The service-oriented architecture has to support *guarantee fulfillment* for safety requirements (cf. Sect. 3.1). Thus, we distinguish between static and dynamic binding of services. Of course, dynamic binding is generally a desired property of SOA. However, to meet safety requirements, we have to support static binding of safety-critical services. Furthermore, the SOA concept of a service bus has to be mapped to TSN by providing a necessary middleware. Finally, the developed SOA concept has to provide means for efficient and controlled data collection for intelligence and learning.

Impacts from Focus Area "Intelligence" The services which will be used by an intelligent system like IPA have to be determined in advance. A clear definition for "malicious" and "wrong" learning must be given and according to that, the safety approaches must be considered by implementing the predictive or preventive mechanisms. Service-oriented architecture defines the services for data access. One of the features considered for IPA is the ability to learn functions, so the research fields involved in this process may identify important factors and form mechanisms to assure the safety of a changed service orchestration. Changed service orchestrations may define further requirements for the communication network to be addressed by a suitable network architecture (e. g. TSN). Moreover, in presence of probabilistic hazards caused by learning components, mechanisms for identifying and mitigating them must be considered.

4 Research Questions and Evaluation Plan

Based on the focus areas and their relationships, we derived *research questions* (cf. Tab. 1) which have to be answered to develop a *future-proof automotive E/E architecture*. Ideas how to evaluate the approach follow.

In the focus area "safety", we will provide a comparison of pattern variants relevant for vehicle CSAs (RQ1.1-1.2), an approach to formulate suitable safety invariants and recommend means required to perform invariant checks (RQ1.3), and deliver a list of arguments how our risk assessment model improves the vehicle assurance case (RQ1.4-1.5).
In the focus area "communication", we will build an experimental setup composed of two to three TSN switches which support Time-Aware Shaper and a set of adequate embedded boards as end-stations (supporting GPIO, SPI, and UART interfaces) (RQ2.1-2.5).
In the focus area "SOA", we will build up an experimental demonstrator (execution platform as well as tooling for service design) (RQ3.5) by implementing selected, safety-critical and intelligent functions as services (RQ3.2-3.4). For a reasonable selection, we first come up

Tab. 1: Research questions assigned to the challenges summarized in Sect. 1.1

RQs & Challenges		Safety
RQ1.1	PC1	Which variants of redundancy patterns will mostly improve our envisaged CSA?
RQ1.2	PC1	Which variants of monitor-actuator patterns are feasible and which perform best in achieving safety?
RQ1.3	PC2, PC4	When do we have to limit or shutdown AI-based ADAS functions, i. e., which ways are feasible to constrain learning for maintaining safety invariants?
RQ1.4	PC1	Does the employed risk assessment model lead to a more complete or more efficient analysis and assurance of CSA safety?
RQ1.5	PC1	How much does the safety of a CSA for an L4/5-vehicle improve over the safety of state-of-the-art CSAs of ADAS-enabled vehicles?
TSN Communication		
RQ2.1	PC1, PC3	Which advantages do TSN features have for the automotive E/E architectures, i.e. CSAs?
RQ2.2	PC2, PC3	Which steps are required to automate TSN configuration?
RQ2.3	PC1	How can different network topologies affect the network performance and configuration overhead?
RQ2.4	PC1, PC4	How can network timing and fail-operational requirements be formally verified?
RQ2.5	PC1, PC2, PC3	How can network failures be classified and handled?
Service-Oriented Architecture		
RQ3.1	PC5	Which criteria can be used for an efficient and structured classification of services in an automotive SOA?
RQ3.2	PC3, PC4	How can safety requirements be satisfied when using dynamic service binding?
RQ3.3	PC2, PC3, PC5	How does a SLA/contract look like in order to facilitate learning?
RQ3.4	PC5, PC4	How can the (i) functional behavior of a service and (ii) its semantics (cf. Semantic Web) be formulated?
RQ3.5	PC3, PC5	How does an automotive compliant implementation of a service bus look like?
Intelligence		
RQ4.1	PC2, PC5, PC3	How does a future E/E architecture have to look like with the integration of AI-based functions?
RQ4.2	PC3	How does the vehicle architecture provide data for “optimization” and “learning” in AI-based functions?
RQ4.3	PC2, PC3	How to use the new dynamics in a future architecture (e. g. SOA, TSN) for AI-based functions?
RQ4.4	PC1, PC4	How to identify and avoid correlations between AI-based functions and safety-critical functions? (e. g. satisfying safety requirements in AI-based functions?)
RQ4.5	PC1, PC4	Within the context of “malicious learning”, how is the relation of AI-based functions and safety? (e. g. identify, predict, and avoid “malicious learning”)

with a classification scheme (RQ3.1).
In the focus area "intelligence", we will concentrate on the occupants' comfort by providing interesting scenarios and demonstrating them on ECUs or lab equipment (RQ4.1). Next, required services (cf. SOA) for the possible safety-critical functions, and the role of the new dynamics with these functions will be defined (RQ4.3-4.4). Approaches for predicting and avoiding malicious learning on the demonstrated functions will be examined (RQ4.5) and after identifying their performance besides their limitations, we will extend the comfort scenarios to cover the related important roles of an IPA.

Use Case We want to show a use case describing an architecture to ease the development and verification of added services. In this architecture, the vehicle control system will be equipped with fail-operational strategies containing fault classes with high risk priority. We want to design proactive behavior into the intelligent components to improve the passenger experience. The operation of intelligent component will be accompanied with high-bandwidth communication requirements, which we want to address by using a TSN infrastructure. Concrete we plan to combine a simulated drive from A to B to study intelligent adaptions and fail-operational behavior in injected, critical situations. Intelligent functions are modeled using services and a coupling between the simulator and a TSN-equipped test bed allows to study TSN capabilities.

5 Conclusion

In this paper, we outlined challenges and derived research questions for a future-proof automotive E/E architecture which is capable of dealing with challenges: (i) vehicles safely operating at levels 4 and 5 of driving automation and (ii) intelligent vehicles that proactively react on and interact with passengers. Moreover, as part of our research plan we sketched engineering and evaluation steps for each focus area. We plan to conduct the evaluation of the concepts developed for the research questions in Sect. 4 based on the described use case.

References

[Br06] Broy, Manfred: Challenges in automotive software engineering. In (Osterweil, Leon J.; Rombach, H. Dieter; Soffa, Mary Lou, eds): 28th International Conference on Software Engineering (ICSE 2006), Shanghai, China, May 20-28, 2006. ACM, pp. 33–42, 2006.

[FK16] Farzaneh, M. H.; Knoll, A.: An ontology-based Plug-and-Play approach for in-vehicle Time-Sensitive Networking (TSN). In: 2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON). pp. 1–8, Oct 2016.

[FSK16] Farzaneh, M. H.; Shafaei, S.; Knoll, A.: Formally verifiable modeling of in-vehicle time-sensitive networks (TSN) based on logic programming. In: 2016 IEEE Vehicular Networking Conference (VNC). pp. 1–4, Dec 2016.

[GK] Gleirscher, Mario; Kugele, Stefan: From Hazard Analysis to Hazard Mitigation Planning: The Automated Driving Case. In: NASA Formal Methods (NFM) – 9th Int. Symp. volume 10227 of LNCS. Springer, Berlin/New York.

[GK16] Gleirscher, Mario; Kugele, Stefan: A Study of Safety Patterns: First Results. Technical Report TUM-I1640, Technische Universität München, July 2016.

[GK17] Gleirscher, Mario; Kugele, Stefan: Defining Risk States in Autonomous Road Vehicles. In: High Assurance Systems Engineering (HASE), 18th Int. Symp. January 2017.

[Hu15] Huval, Brody; Wang, Tao; Tandon, Sameep; Kiske, Jeff; Song, Will; Pazhayampallil, Joel; Andriluka, Mykhaylo; Rajpurkar, Pranav; Migimatsu, Toki; Cheng-Yue, Royce et al.: An empirical evaluation of deep learning on highway driving. arXiv:1504.01716, 2015.

[IS11] ISO: , Road vehicles–Functional safety (ISO 26262), 2011.

[Ku17] Kugele, Stefan; Obergfell, Philipp; Broy, Manfred; Creighton, Oliver; Traub, Matthias; Hopfensitz, Wolfgang: On Service-Orientation for Automotive Software. In: IEEE International Conference on Software Architecture, ICSA 2017, Gothenburg, Sweden, April 3-7, 2017. IEEE Computer Society, pp. 193–202, April 2017.

[KW16] Koopman, Phil; Wagner, Michael: Challenges in Autonomous Vehicle Testing and Validation. In: SAE World Congress. 2016.

[Pe14] Perera, Charith; Zaslavsky, Arkady; Christen, Peter; Georgakopoulos, Dimitrios: Context Aware Computing for The Internet of Things: A Survey. IEEE Communications Surveys & Tutorials, 16(1):414–454, 2014.

[So13] Sommer, S.; Camek, A.; Becker, K.; Buckl, C.; Zirkler, A.; Fiege, L.; Armbruster, M.; Spiegelberg, G.; Knoll, A.: RACE: A Centralized Platform Computer Based Architecture for Automotive Applications. In: Electric Vehicle Conference (IEVC), 2013 IEEE International. pp. 1–6, Oct 2013.

[So14] Society of Automotive Engineers: Taxonomy and Definitions for Terms Related to On-road Motor Vehicle Automated Driving Systems, SAE Std. J3016. 2014.

[TMB17] Traub, Matthias; Maier, Alexander; Barbehön, Kai L.: Future Automotive Architecture and the Impact of IT Trends. IEEE Software, 34(3):27–32, 2017.

[Ya16] Yan, S.; Teng, Y.; Smith, J. S.; Zhang, B.: Driver behavior recognition based on deep convolutional neural networks. In: 2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD). pp. 636–641, Aug 2016.