Towards Flexible and Dependable E/E-Architectures for Future Vehicles

Gereon Weiss, Philipp Schleiss, Christian Drabek

Towards Flexible and Dependable E/E-Architectures for Future Vehicles

Gereon Weiss, Philipp Schleiss, Christian Drabek
Fraunhofer ESK
Hansastr. 32, Munich, Germany
{*firstname.lastname*}@esk.fraunhofer.de

***Abstract*— Future vehicles are expected to evolve towards enabling fully electric and autonomous driving. However, technically this evolution requires fundamental changes of traditional automotive engineering principles. Specifically, challenges arise for the Electric/Electronic (E/E) vehicle architectures as underlying basis for almost all car functionalities. Higher demands on vehicle system's flexibility and dependability have to be incorporated. We present a novel approach for such future E/E-architectures which considers these requirements as first principles by exploiting runtime adaptation capabilities. Based on use cases, a generic hardware and software architecture is presented which enables technology-independent realization of the provided concepts. Additionally, the incorporated generic failure management and design support are introduced. The approach has been evaluated in different prototype demonstrators, including an e-vehicle prototype compromising enhanced driving functionality. Thereby, the advantages of the concepts for future vehicle E/E-architectural development could be highlighted.**

I. Future Smart Vehicles

The automotive domain is facing tremendous changes with the transition to novel mobility paradigms. One particular market driver is the promotion of *Fully Electric Vehicles (FEVs) [1]* targeting zero-emission transportation. Its impact on the vehicle system is substantial, since many previous purely mechanical parts are vanishing from the car. For example, mechanical braking can be substituted by sole engine braking with recuperation, saving energy as well. Another prominent change is the advent of *Automated Driving* [2] towards autonomous smart cars, which e.g., decreases the number of accidents and enables higher traffic density. With increasing levels of automation, the driver is more and more taken out of the loop of control. In turn, this has great influence on the required dependability of vehicle systems. Even in the case of a system failure, it has to be operational until a safe state has been reached. For example, a failing functionality could also be substituted by a simpler but working functionality, leading to *graceful degradation* of the car, instead of a total failure. This shift from so-called *fail-safe* to *fail-operational* behavior is essential for realizing any highly automated driving. The innovation cycles and time-to-market for introduction of new features in cars are becoming shorter, analogous to other electronics consumer industries. Moreover, car owners demand flexible updates of their vehicles, e.g., with newest driver assistance features. In addition, since with increasing performant Electronic Control Units (ECUs) in today's cars, many features like advanced services can mainly be implemented through *software* [3].

As promising as the benefits of future mobility are, diverse challenges have to be tackled before this visions can be realized. One specific aspect is the E/E-architecture of modern vehicles. Already today, present systems have reached their limits with respect to integrating more and more novel software-driven features, like manifold advanced driver assistance systems [4]. Novel driving functionality in contrast is often not constraint to a single domain, requiring cross-domain dependencies and communication. Therefore, so-called multi-domain controllers are introduced to reflect this conceptual change. However, they do not inherently provide the required dependability and pose only as a temporary patch for the antiquated domain architecture concept. Thus, novel E/E-architecture concepts are required that incorporate the requirements of future smart vehicles, like flexibility and higher dependability. In the following, we present our approach for developing such a new vehicle E/E-architecture exploiting adaptability concepts addressing such challenges. It comprises a novel system E/E-architecture with redundant communication paths, a generic safety mechanism, a development methodology, and the respective tooling. The approach has been implemented and evaluated in different prototype demonstrators, including an e-vehicle prototype car with dependable software-based functionality.

The paper is structured as follows. In Section II the overall approach and use cases are introduced. The derived system architecture is presented in Section III, along with its generic safety mechanism. Developed prototype demonstrators and evaluation are outlined in Section IV. Section V discusses related work and projects, before this paper is concluded in Section VI.

II. Motivation & Use Cases

The target of the presented SafeAdapt approach [5] is to enhance robustness, availability, and efficiency of in-vehicle E/E-architectures while preserving the highly-demanding functional safety requirements of anticipated future smart cars. Enhanced redundancy concepts and mechanisms are required for automotive systems to achieve sufficient safety levels. Duplication of complete systems like in aerospace domain [6] is alone for cost reasons no viable option. SafeAdapt focuses on FEVs with respect to safety and re-configurability of their basic control system's E/E-architecture. Several specific automotive use cases and requirements have been investigated:

A. Enhanced Failure Management with Fail-Operational Behavior

This use case includes ensuring the availability of vehicle functionality despite failures, which can also be safety-critical. Failures have to be managed in an adequate and safe way, so the control of the car is given at any time. An exemplary application is the *Steer-by-Wire (SBW)* functionality of a modern car, which steers purely electronically, without any mechanical backup.

B. Plug'n'Play & Extensibility

For taking higher flexibility and shorter time-to-market into account, a use case enabling the changing of functionality of a car is considered. An example is a driver wanting to upgrade the vehicle by installing new software. Another example is the adding and removing of components of the vehicle in a plug'n'play fashion.

C. Energy Efficiency

The E/E-architecture is also contributing to the overall vehicle energy consumption. The addressed scenarios here extend existing approaches, like partial networking, by enabling a flexible deactivation of not needed functionality in specific contexts. An example is the deactivation of comfort functionality in the case of a low battery state-of-charge.

III. System Architecture

For fully unleashing the potential of future adaptive E/E-architectures a hardware architecture capable of meeting enhanced dependability requirements is essential, e.g., consideration of redundant communication paths. Based on this, a software architecture enabling flexible configuration and adaptation of the in-vehicle system based on the assumptions and capabilities of the underlying hardware architecture can be utilized. For achieving interoperability and backward compatibility, our concepts are introduced as extension to the wide-spread automotive standard *AUTOSAR* [7] and follow the relevant *ISO 26262* automotive functional safety standard [8].

A. Generic Hardware Architecture

The developed hardware architecture is based on general concepts on which specific architectures can be derived from (cf. e-vehicle demonstrator architecture in Section IV). One fundamental principle is the detachment of sensors and actuators from the computational units, following a separation of the traditional input-process-control pattern (s. Figure 1). This results in *central computational core (CCC)* control units and via communication links connected sensors and actuators. Moreover, the communication is synchronized between the networked nodes. For enabling fault-tolerance, which is mandatory for the envisaged fail-operational behavior, redundant paths between every control unit and the detached sensors and actuators are essential. The latter are also likely to be installed redundantly to cope with failures. In detail, the proposed hardware architecture follows a *1-out-of-2 safety design* with additional diagnostic capabilities.

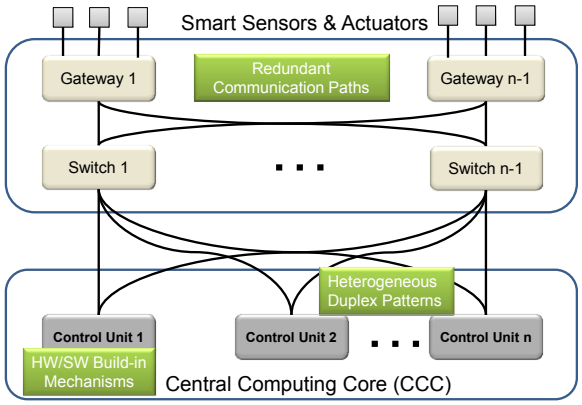

Figure 1. Generic hardware architecture

In addition to redundant communication and sensor/actuator infrastructure, the control units must also be capable of containing and handling failures. Examples are lockstepping mechanisms for detecting local errors by observing divergent computational outputs. In the scope of this project, two different hardware platforms (cf. Section IV) are being utilized. One is capable of lockstepping on-chip level, the other one on network level. Only failures which cannot be managed completely by hardware platforms are handled by the *Safe Adaptation Platform Core (SAPC)* (s. Section III.C).

B. Software Architecture

Based on the hardware architecture principles, the software architecture has been defined, reflecting a layering aligned with the AUTOSAR standard (s. Figure 2).

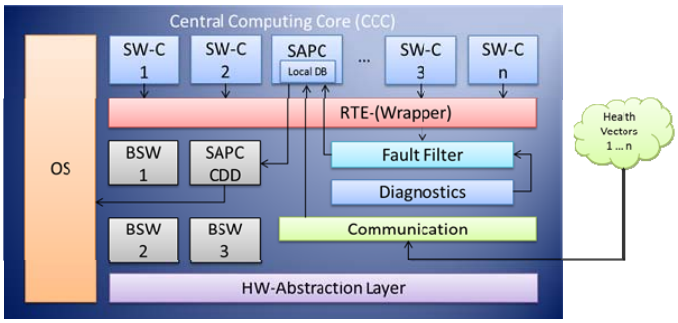

Figure 2. Overview of software architecture on an ECU

Through this alignment, generally required mechanisms can be realized based on specific implementations, e.g., enhanced real-time operating systems (OSs) or common AUTOSAR OS. However, below the Runtime Environment (RTE), platform-specific modules have to be implemented. These include for example diagnostics services and a *Fault Filter* (cf. Section III.C). Specific emphasis has been laid on defining adequate fault assumptions and fault containment regions, enabling to adopt safe states for distinct faults [9]. In addition, the SAPC *Complex Device Driver (CDD)* implements platform-specific functionality to change configurations, e.g., de- and activation of specific tasks. OS-specific mechanisms are triggered by the SAPC, which itself is developed as platform-independent AUTOSAR Software Component (SW-C). This enables a single implementation of the SAPC, thereby facilitating its safety qualification.

C. Safe Adaptation Platform Core

Allowing for a generic safety mechanism, the SAPC is introduced following the *Safety-Element-out-of-Context (SEooC)* concept from ISO 26262 [8]. In our approach, software applications are classified into *hot-* and *cold-standby tasks*. Their individual classification mainly depends on their need to access historic data, in order to function correctly. More specifically, cold-standby tasks are not scheduled during normal operation and only activated in case of a failure. In turn, hot-standby tasks actively replicate the actions of a primary instance, with suppressed outbound communication.

The actions needed to handle faults can be selected by factoring in the local and global status of the system. The local system status is diagnosed and forwarded to the SAPC by the Fault Filter mentioned above. For this, each control unit establishes specific platform-specific fault detection mechanisms, e.g., multicore processor lock-stepping or memory protection. To enable general and technology-independent fault handling, these mechanisms are represented as abstract fault classes, e.g., memory protection fault. These may be defined for specific systems individually during the system design in coherence with the fault assumptions. For deriving a global status of the entire system, the control unit exchange expanded heartbeat information during regular cycles using the so-called *Health-Vectors*. Through this, each control unit broadcasts the status of its software components to all other control units. Due to the extremely low failure rates of redundant communication paths, it can be assumed that a platform is no longer available, for instance by shutting itself down, whenever a Health-Vector does not arrive at a control unit within a specific time period. Such a shutdown situation occurs when a control unit detects that it is no longer trustworthy and therefore has transited into fail-silent mode.

By exploiting the abstract fault classes, the SAPC determines a predefined fault handling mechanism for the respective fault. For example, this may be the switching to a different configuration. These fault handling plans must obviously be consistent across the system. This is ensured through pre-validated data generated from the design. The actual fault handling itself is carried out with platform-specific mechanisms through the CDD. In order that the failure compensation is consistently carried out at the same time, synchronization of the control units and real-time communication is required (cf. Section III.A).

IV. Demonstrators & Evaluation

The approach has been evaluated within different prototypes addressing specific aspects. Flexible extensibility in a plug'n'play fashion and interworking with state-of-the-art technology is evaluated by an AUTOSAR model-car demonstrator. An automatic safety re-evaluation demonstrates the extensibility capabilities, e.g., which would be carried out at an official maintenance service provider. A *Driver-in-the-Loop simulation* in the *DynaCar* demonstrator specifically addresses energy-efficiency use cases. A similar system setup as in the full-scale e-vehicle has been chosen, while a low battery state-of-charge of less than 35% is simulated. To extend the range, the SAPC adapts the configuration so that braking system's recuperation is maximised and the non-safety-critical drowsiness detection application is shutdown. Experiments executing a standardized driving profile of the New European Driving Cycle (NEDC) show potential energy savings resulting in a *range extension of up to 25%* within the executed scenarios. Moreover, driver studies have been undertaken in the safe simulation environment to derive the maximal time, the SAPC mechanism has to complete its adaptation. By the example of a failing SBW application, the study results show that drivers do not notice the fault if it is handled by the SAPC fail-operational mechanism *within 150ms*. If the steering is unresponsive for more than 250ms, the situation is not controllable anymore. The SAPC mechanism has initially been specified to recover hot-standby functionality within 50ms, so that from a safety perspective it provides a suitable solution for enabling such fail-operational functionality. A *prototype e-vehicle* is utilized to mainly show the enhanced failure management in a real environment. A schematic overview of the prototypes hardware architecture is depicted in Figure 3.

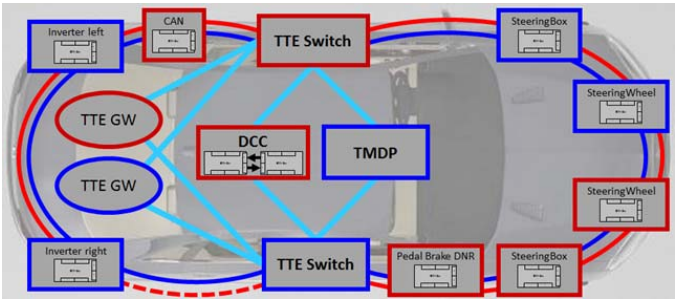

Figure 3. Hardware architecture of e-vehicle demonstrator

For this, two control units (*RACE DCC* and *TMDP*) are used to create a reliable core for the control infrastructure, through a highly diversified hardware design. Both platforms not only use different hardware elements but also apply diverging diagnostic principles (cf. Section III.A). The control units are synchronized and redundantly interconnected via *Time-Triggered Ethernet (TTE)*. All sensors and actuators of the vehicle system are integrated through *Gateways (GWs)* to a unified Ethernet backbone. This demonstrator proofs the feasibility of the SAPC mechanisms by an implemented fail-operational SBW. The proposed approach and its architecture have been compared to a state-of-the-art architecture providing the same safety and functionality (s. Figure 4). It can clearly be derived that significant potential for savings with respect to the different metrics, such as cost, weight, size, and power can be gained. However, such numbers depend on the specific architectural constraints and functionality, so that this has to be evaluated for each vehicle system individually.

V. Related Work & Outlook

SafeAdapt aims to fill the gap between different approaches towards future flexible and dependable automotive electric / electronic systems. Within the FP6 project DySCAS [10][10], for example, a middleware for self-adaptive vehicles was developed. In contrast to SafeAdapt's technology independent approach and alignment with AUTOSAR, a completely new middleware with policy-driven management was utilized.

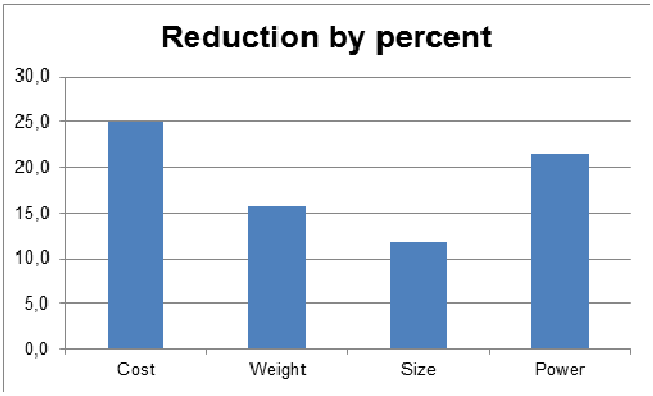

Figure 4. Calculated improvements of utilizing SafeAdapt approach instead of traditional redundancy

For general embedded systems domains, several projects have pursued integrating adaptive control. For instance, ACTORS [11] addresses the design of complex embedded systems and adaptive resource management, MUSIC [12] provides middleware-base self-adaptation for mobile phones, or iLand [13] pursues improvement of systems' flexibility, scalability, and composability through developing a modular component-based middleware. Also, the DREAMS project [14] aims at providing methods, tools, and adaptation strategies for cyber-physical systems across domains. Our approach in turn focuses on safely downscaling fail-operational E/E-architectures to meet the cost pressure of the automotive industry. With respect to the main influencing market drivers the wide-ranging European Green Vehicles Initiative [1] can be mentioned for electric mobility and here exemplary the projects Adaptive [15] and Ko-HAF [16] for automated driving, which are also related partially to E/E-architecture changes.

Within the automotive industry the paradigm shift towards flexible high-performant in-vehicle computing has already been recognized as substantial challenge. Great advantages are given for small volume manufacturers, e.g. of electrical vehicles. For instance, in the Adaptive City Mobility projects [17] mechanisms of the presented approach will be utilized to enable a flexible inner city mobility solution by exploiting an adaptive architecture. Another example is given with the ongoing activities of the AUTOSAR association towards standardizing an adaptive platform [18]. Unlike, SafeAdapt provides integrated mechanisms for the classic AUTOSAR platform and other enhanced future control units, like RACE or TMDP. The general feasibility of integrating adaptation into AUTOSAR has already been shown by different approaches [19][20]. Thus, it poses as a solution for the required flexibility and dependability of classic driving control functions in addition to power-computing on adaptive platforms.

VI. Conclusion

Tremendous changes in today's mobility require disruptive changes of the underlying E/E-architecture. Within the SafeAdapt project novel concepts are developed to pave the way for the required flexibility and dependability of such architectures. Based on several use cases a generic hardware and software architecture has been defined, which is capable of providing fail-operational behavior. For this, a generic safety mechanism is introduced that enables safe responses to changing conditions, like failures, by exploiting adaptability. Our approach has been implemented and evaluated in different demonstrators, e.g., a driver-in-the-loop simulation and real e-vehicle prototype.

Acknowledgments

The research leading to these results has partially received funding from the European Commission within the Seventh Framework Programme as part of the SafeAdapt project under grant agreement number 608945 and from the German Federal Ministry for Economic Affairs and Energy (BMWi).

References

[1] European Green Vehicles Initiative: http://www.egvi.eu

[2] SAE International, "Taxonomy And Definitions For Terms Related To On-Road Motor Vehicle Automated Driving Systems", SAE J 3016:2014, 2014.

[3] S. Clarke, B. Fitzgerald, P. Nixon, K. Pohl, K. Ryan, D. Sinclair, and S. Thiel, "The role of software engineering in future automotive systems development", In Proceedings of the SAE World Congress 2008.

[4] D. Reinhardt and M. Kucera, "A New Approach for Large Scale Software Integrated Automotive Systems", In Proceedings of the 3rd International Conference on Pervasive Embedded Computing and Communication Systems, pp. 221-226, 2013

[5] Project SafeAdapt – Safe Adaptive Software for Fully Electric Vehicles: http://www.safeadapt.eu

[6] J. Windsor, M.-H. Deredempt, and R. De-Ferluc, "Integrated modular avionics for spacecraft - User requirements, architecture and role definition," in Proc. of the 30th IEEE/AIAA Digital Avionics Systems Conference (DASC), Oct. 2011, pp. 1–16 (8A6).

[7] AUTomotive Open System ARchitecture. http://www.autosar.org

[8] International Organization for Standardization (ISO), "ISO/DIS 26262: Road vehicles - functional safety," 2011.

[9] A. Ruiz-Lopez, G. Juez, P. Schleiss, G. Weiss, "A Safe Generic Adaptation Mechanism for Smart Cars", 26th International Symposium on Software Reliability Engineering, 2015.

[10] R. Anthony, A. Rettberg, D.-J. Chen, I. Jahnich, G. de Boer and C. Ekelin, "Towards a Dynamically Reconfigurable Automotive Control System Architecture". In IESS, volume 231 of IFIP. Springer, 2007.

[11] Project ACTORS – Adaptivity and Control of Resources in Embedded Systems: http://www.actors-project.eu

[12] J. Floch et al., "Playing MUSIC - building context-aware and selfadaptive mobile applications", Software: Practice and Experience 43(3), March 2013, pp. 359-388.

[13] Project iLand – mIddLewAre for deterministic dynamically reconfigurable NetworkeD embedded systems: http://www.artemis-ia.eu/project/index/view?project=10

[14] Project DREAMS – Distributed REal-time Architecture for Mixed Criticality Systems: www.uni-siegen.de/dreams

[15] Project AdaptiVe – Automated Driving APplications and Technologies for Intelligent VEhicles: http://www.adaptive-ip.eu

[16] Project Ko-HAF – Cooperative Highly Automated Driving: https://www.ko-haf.de

[17] Projects:ACM – Adaptive City Mobility: http://adaptive-city-mobility.de

[18] M. Bechter, "AUTOSAR Adaptive Platform", 8th AUTOSAR Open Conference, Tokyo, Japan, October 2015.

[19] Marc Zeller, Christian Prehofer, Daniel Krefft, and Gereon Weiss. Towards runtime adaptation in AUTOSAR. *SIGBED Rev.* 10, 2013.

[20] H. Martorell, J.-C. Fabre, M. Roy, and R. Valentin. Improving adaptiveness of AUTOSAR embedded applications. In *Proceedings of the 29th Annual ACM Symposium on Applied Computing* (SAC '14). ACM, New York, NY, 2014.