Model-based resource analysis and synthesis of service-oriented automotive software architectures

Stefan Kugele[1], Philipp Obergfell[2], Eric Sax[3]

Abstract: This summary refers to the paper *Model-based resource analysis and synthesis of service-oriented automotive software architectures* [KOS21]. This paper has been published in the Journal on Software and Systems Modeling (SoSyM) in September 2021.

Context: One drawback of today's automotive software architectures is their strong integration into the onboard communication network based on predefined dependencies at design time. The idea is to reduce this rigid integration and technological dependencies by using a service-oriented architecture (SOA) that dynamically regulates network communication at run-time. **Aim:** We target to provide a methodology for analysing hardware resources and synthesising automotive service-oriented architectures based on platform-independent service models that are transformed in a subsequent step into a platform-specific architecture realisation process. **Approach:** For the first part, we apply design space exploration and simulation to derive analysed deployment configurations at an early development stage. We refine these configurations to AUTOSAR Adaptive software architecture models required for a subsequent implementation process for the platform-specific part. **Result:** We present optimal deployment configurations for our next generation of E/E architecture. We also provide simulation results that demonstrate the ability of these configurations to meet the run time requirements.

Keywords: Automotive; Service-oriented architecture; Real-time behaviour; Model-based design

During the last decades, thousands of primarily software-controlled functions were included in modern cars, which are executed on many electronic control units (ECU). Driving forces for this development were (i) safety requirements, (ii) customer demands for more comfort and the newest infotainment systems, and (iii) advanced driver assistance systems allowing to reach higher levels of driving automation. These driving forces led to the current electric/electronic (E/E) architectures best characterised as historically grown, mostly federated, partly integrated architectures with often pragmatic, cost-efficient, and ad-hoc solutions. More than 100 custom-built ECUs realise the behaviour of the functions in an interplay of timed signals sent via heterogeneous bus systems with gateway structures.

In current AUTOSAR Classic Platform-based software architectures, communication paths are static, i.e., distributed software applications are strictly bound to ECUs at the development

[1] Technische Hochschule Ingolstadt, Research Institute AImotion Bavaria, 85049 Ingolstadt, Germany, Stefan.Kugele@thi.de
[2] BMW Group Research, New Technologies, Innovations, 85748 Garching bei München, Germany, Philipp.Obergfell@bmw.de
[3] Karlsruhe Institute of Technology, Institute for Information Processing Technologies, 76131 Karlsruhe, Germany, eric.sax@kit.edu

stage. This, however, impairs the possibility of developing both parts independently of each other. This ECU or message-centric focus can hardly cope with future automotive software and systems engineering challenges. During the entire life cycle, the software contained in the vehicle must always be kept up to date to follow customers' demands and to be in line with the state-of-the-art especially wrt. functional safety and security. To foster faster adoption of improved features and innovative services, more *agile* development methods need to be built into established processes. When designing a system's software architecture, the primary goals are to develop safe and secure, performant, flexible, adaptive, and maintainable systems. *Service-oriented architectures* are known for supporting the design of flexible systems since *late binding* at *run-time* facilitates the integration of new functionalities and services. SOA paradigms help to decouple software and hardware development to a large extent, facilitating fast and light-weight software updates in a fast-moving agile development process based on the idea of *continuous software engineering*.

Our guiding research question for this work is: *In a pre-development project, can the unification between a specification of an independently created novel E/E architecture and initial estimates of the resource consumption of the service-oriented software architecture succeed and a system specification be derived from it?*

To answer this research question, we presented a platform-independent meta-model for service-oriented architectures, drew a methodology for deploying corresponding platform-independent models on a centralised E/E architecture, and finally embedded the gained results in a platform-specific software development process based on AUTOSAR AP. For this purpose, experts first determined resource requirements; in particular, they provided time and memory requirements and safety assessments. The same values were determined for the centralised computing platform. In an approach that used simulation to estimate end-to-end latencies for a sub-function of automated driving, namely the provision of the environment model and a constraint-based design space exploration technique, deployment candidates were synthesised and evaluated. Since the architecture in this pre-development project is to be designed for SAE level 4 automated driving, special safety measures had to be taken into account in the architecture on both the hardware and software sides. The SOA-based software architecture can reconfigure itself if the environment model is not available in time so that fail-operational models can be used. We were able to show through simulation that all real-time requirements could be met for the reconfigurations considered. In summary, it can be said that the leading research question could be answered positively.

Data Availability The original paper [KOS21] is freely available as *Open Access* under the given reference. Unfortunately, we could not provide more detailed data and especially models for reasons of confidentiality.

References

[KOS21] Kugele, S.; Obergfell, P.; Sax, E.: Model-based resource analysis and synthesis of service-oriented automotive software architectures. Softw. Syst. Model. 20/6, pp. 1945–1975, 2021, URL: https://doi.org/10.1007/s10270-021-00896-9.