A Fast Prototyping Framework for Service-Oriented Automotive Applications

Matthias Becker*†, Zhonghai Lu*, De-Jiu Chen†
*Electronics and Embedded Systems, School of EECS
†Mechatronics, Machine Design, School of ITM
KTH Royal Institute of Technology, Sweden
{mabecker, zhonghai, chendj}@kth.se

Service-Oriented Architectures (SOA) provide a flexible platform for advanced automotive software applications. We present a research platform tailored for fast prototyping of platform software and applications.

The target hardware is built around a 1:10 scale RC car (see Fig. 1). Several sensors and actuators are connected over microcontrollers that can be accessed from higher-level compute nodes over bus connections. User applications are executed on 4 Linux-based compute nodes which communicate over a multi-hop Ethernet network.

The platform is designed in a modular way such that all compute nodes can be separated from the network-level, allowing different hardware configurations of the platform. The compute nodes and network-level can further be used apart from the underlying RC car platform.

Network and Communication

The presented hardware platform utilizes a Software-Defined Network (SDN) to interconnect all compute nodes. This allows for a controlled usage of all network resources.

All communication of user-applications is realized over SOME-IP [1], [2], an automotive middleware layer that is based on the SOA principle. We further utilize a network admission mechanism to only admit new communication services if enough network resources are available to meet all requirements [3].

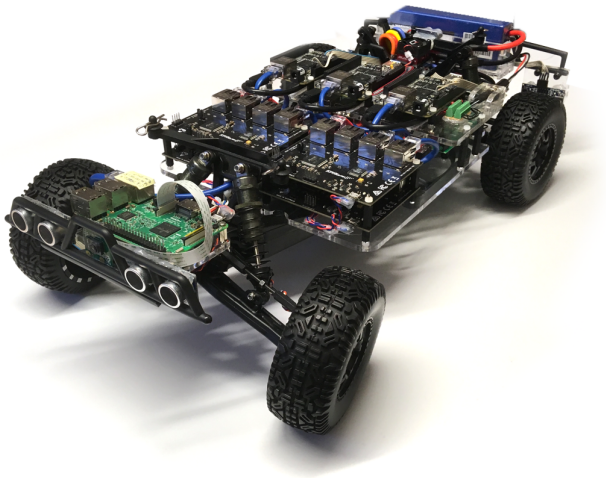

Fig. 1: RC-Car Hardware Platform used for prototyping

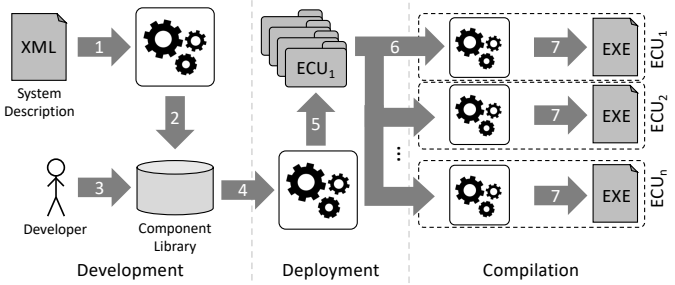

Fig. 2: Tool support during the prototyping process.

Development Framework

The tailored development framework is a key enabler for fast prototyping cycles. The framework automatically generates code skeletons for user tasks, and all required management and configuration code of the underlying SOA framework, based on a user specified application model. It is then automatically transferred and compiled on the respective compute nodes.

The different stages supported by the framework are shown in Fig. 2. (1) A system description is provided to specify all required software components. (2) The presence of all required components in the component library is then checked, if components do not yet exist, the basic code skeletons are generated, (3) and the developer can implement the internal behavior. Once all implementation is completed (4), the required components can be extracted and necessary configuration files are generated. Further, all required drivers to interact with sensors and actuators are configured (5). The remaining steps transfer the respective files to the target compute nodes on the hardware platform (6), where they are compiled (7).

Demonstration

We demonstrate the usability of the platform based on a remote-operation scenario which includes several distributed software components. The process from application modeling up to the execution of the final application is shown.

References

[1] AUTOSAR FO Release 1.3.0, "SOME/IP Protocol Specification," AUTOSAR, Standard, Dec 2017.
[2] ——, "SOME/IP Service Discovery Protocol Specification," AUTOSAR, Standard, Dec 2017.
[3] M. Becker, Z. Lu, and D. Chen, "Towards qos-aware service-oriented communication in e/e automotive architectures," in *44th Annual Conference of the IEEE Industrial Electronics Society (IECON)*, 2018.