Virtual Validation of Cyber Physical Systems

Patrik Feth, Thomas Bauer, Thomas Kuhn
Fraunhofer IESE
Fraunhofer-Platz 1
67663 Kaiserslautern
{patrik.feth, thomas.bauer, thomas.kuhn}@iese.fraunhofer.de

Abstract: The increasing importance of Cyber Physical Systems (CPS) yields new challenges for their systematic and efficient quality assurance. CPS are characterized by open and heterogeneous architectures and environments. For embedded systems, this implies a separation of the currently very tight integration of hardware and software components. Development and testing of these systems require new development environments that enable prototyping and testing of system concepts on different levels of abstraction. In this paper, we describe the extension of our FERAL framework to support the prototyping of automotive CPS by adding an AUTOSAR simulation environment. This supports the virtual development of next generation open architectures that integrate software components from multiple suppliers on one hardware platform.

1 Introduction

Cyber Physical Systems (CPS) are complex compositions of software intensive systems. Software components from different suppliers are executed on the same hardware platform. This requires software platforms that enable the separation of software and hardware components, and enable the definition of standard interfaces. In the automotive domain, the AUTomotive Open System Architecture (AUTOSAR) is this standardized platform that provides standardized access to hardware services, and which has been already widely accepted in industry.

Currently, even AUTOSAR conforming systems are delivered as one entity of Hardware and Software components – usually, a complete Electronic Control Unit (ECU) is provided by suppliers. AUTOSAR systems are statically configured at compile time and suppliers take care that the delivered software configuration fulfills all requirements. OEMs integrate these third party components with other components to create the vehicle. The resulting federated architectures consist of a large number of ECUs that interact via bus systems, common bus types are CAN and FlexRay networks.

In the near future, a first transition to CPS will happen when only software components will be delivered by suppliers. They will be integrated on only few powerful ECUs by OEMs, yielding more flexible and more economic integrated architectures. This has significant impact on quality assurance processes. Deployment decisions and hardware platforms significantly affect the behavior of control algorithms and control loops; therefore, it must be possible to predict the impact of those decisions to an integrated architecture early in development processes.

Right now, the validation of a system functions and performance against specifications is conducted late in the QA process during the hardware-in-the-loop (HiL) testing. This requires software and hardware components to be already integrated. The development of integrated architectures demands early performance evaluations that can be conducted before all components are available and integrated with each other.

Our Framework for Efficient Simulator Coupling on Requirements and Architecture Level (FERAL) [Ku13] is able to provide virtual evaluation platforms by coupling existing specialized simulators into one semantically integrated simulation scenario. This enables simulation of functional behavior in the context of one virtual deployment that adds the effect of task scheduling, the allocation of runnables to tasks, and the impact of communication networks. In this paper, we illustrate the addition of an AUTOSAR platform simulation to enable virtual validation of next generation automotive applications.

The remainder of the paper is structured as follows: Section 2 presents existing approaches in the area of simulation and virtual validation. Section 3 explains the extensions that have been performed to the used simulation framework. Section 4 highlights the contribution of this work and shows our vision of future software development. Section 5 concludes the paper.

2 Related Work

Validation by simulation is a relevant topic for both industry and academia. In research, the complex questions of combining different models of computation for coupled (co-) simulation are of specific interest. Platforms and simulators are integrated to reveal effects and hidden interactions that would not show up if only a single aspect would be analyzed. To only name a few examples of simulator couplings, the authors of [WB13] couple TinyOs with Modelica, [Bj11] describes the coupling of Simulink with the ns-2, and the authors of [RSB11] couple SystemC with the network simulator OMNet++. The resulting simulation platforms cover multiple aspects of embedded systems, but are as specialized as the underlying simulators. All of these publications illustrate that significant effort needs to be spent for the creation of these couplings.

The interest of the industry in simulation becomes obvious when considering the international standard for the safety of road vehicles: ISO26262 [ISO11]. In this standard, simulation is explicitly recommended as a quality assurance technique for verification of system design artifacts. For the automotive industry, a selection of common tools is the following: VEOS by dSpace [Ro13], ISOLAR-EVE by ETAS [Di13], and the VAP as product of a joint venture of Mentor Graphics, BMW and MBTech [MDF13]. Simulation is currently used for testing of code in context of virtual hardware. The aforementioned tools simulate the behavior of an ECU or a specific communication medium. Support for earlier process stages, e.g. the evaluation of architecture and design decisions is usually not available.

Recent standards for the coupling of simulation models substantiate this. The most prominent approach, Functional Mockup Interface [Bl11] stays on the abstraction level of functional models that read data from sensors and control their environment through actuators. Integrating other relevant classes of CPS such as (wireless) networks is possible, but not efficient due to the fixed model of computation and communication that functional mockup units are based on.

The discussed findings indicate that currently used simulators are accurate but highly specialized. Simulator coupling is possible, but also yields specialized and tailored solutions. Validation of Cyber Physical Systems that use, for example integrated architecture, requires the addressing of multiple system aspects with sufficient accuracy. This requires simulation environments that can be quickly tailored to reflect architecture decisions. As contribution of this work, we will demonstrate how our approach performs on this requirement.

3 FERAL based Simulation Approach

To enable virtual validation through simulator coupling in the automotive domain, we have extended our FERAL framework to support AUTOSAR applications. FERAL is based on simulation components that communicate over links with ports as communication endpoints. Simulation components realize simulation models either by integrating external simulators, or by defining explicit simulation behavior. Directors define the model of communication and computation for a set of components. Through nesting of directors, execution semantics can be combined into one scenario.

AUTOSAR is split into two main layers, which are coupled by the Runtime Environment (RTE). AUTOSAR application components implement application behavior, while AUTOSAR basic software components realize basic services. The AUTOSAR basic software is realized based on an OSEK conforming operating system. Our goal is the integration of AUTOSAR concepts to the level of source code compatibility. Our integration of the AUTOSAR platform is therefore based on discrete event execution semantics, which are already supported by FERAL. Discrete event semantics enable the implementation of numerous communication and execution schemes, and therefore are able to express all of the AUTOSAR and OSEK communication and execution concepts.

With the RTE, AUTOSAR provides a defined mechanism for communication between AUTOSAR applications and basic software services like communication, hardware access, and scheduling. For the execution of AUTOSAR applications, we therefore had to provide a tailored RTE that links to the simulator, which will be providing all necessary AUTOSAR and OSEK basic services. This way it is possible to virtually validate a new automotive application in a variety of abstraction levels that realize open-loop and closed-loop scenarios. For example, it is possible to execute a new software function in an idealistic environment without resource constraints for unit testing. By changing components, it is however possible as well to connect this software component to a simulated bus that replays formerly recorded communication patters, e.g. from test drives. Finally, simulation components could be replaced by proxies that link to simulation servers of OEMs, providing standardized test beds for function types, e.g. functions that interact with braking systems.

This requires the integration of AUTOSAR conforming C code into the FERAL simulation framework. AUTOSAR uses generated macros to pass parameter values (sender-receiver interfaces), or to perform remote procedure calls (client-server interfaces). These macros are generated based on a machine description file together with a tailored basic software. FERAL realizes one instance of such a tailored basic software by providing simulation components that realize basic services and integrate dependent simulation models, e.g. Simulink models. Figure 1 illustrates an integrated automotive simulation scenario in FERAL. It applies the two aforementioned AUTOSAR communication paradigms: client-server interfaces and sender-receiver interfaces. To represent them in FERAL, two different simulation components have been developed.

The *Services* component in Figure 1 simulates client-server communication. It offers a defined set of operations to the automotive application that resembles common basic services, e.g. access to communication busses. Optionally, service calls may be routed to multiple components to simulate more complex configurations that include e.g. complex device drivers. If the application calls one of the operations, synchronous call semantics are applied to simulate a call to a simulated basic service component or to another component offering a service. The calling component is suspended until the basic service has been completed, then, control is handed back to the caller. Calling of services is mapped to event messages that are transmitted between components. This way, FERAL simulates both possible types of service calls: service calls on the same ECU would be realized by a direct call on real hardware, service calls to a different ECU require serialization of the call to a message that can be transmitted through a network. Event messages can be transmitted through simulated networks, therefore, platforms that support client-server calling across ECUs are supported as well.

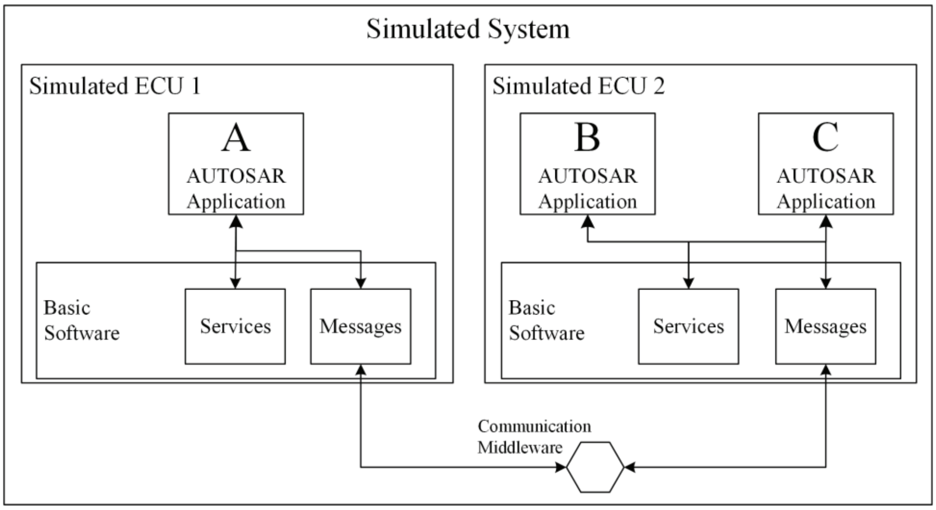

Figure 1: Structure of an automotive simulation scenario with FERAL

The second communication paradigm, the sender-receiver paradigm, is implemented by the *Messages* component shown in Figure 1. It simulates a storage of global values, in real deployments these are usually realized by global variables that store values that are communicated between AUTOSAR applications. Those represent for example values read from sensors or control values to be transmitted to actuators. New values may be received at any time from other applications or through communication networks. Usually, the arrival of new data is not explicitly signaled. Data values are transmitted periodically by explicit tasks or hardware controllers. Simulation of this behavior is realized by the *Messages* component via a Map that stores the most recent value for every received signal.

It is updated with new values from other applications or from communication networks. Simulation components may read and write the values in this map at any time. To enable simulation of multi-core deployments, multiple concurrently executed applications may be attached to the same *Messages* component instance. The chosen implementation decides whether potentially concurrent accesses will be silently serialized, or whether they will be reported as scheduling error.

By creating different RTE configurations, several architectures may be simulated. FERAL represents each ECU by a number of event based simulation components that integrate AUTOSAR applications, as well as simulated basic services and complex device drivers. When simulating deployments, simulation components that represent software are allocated to tasks with defined runtimes and priorities. An ECU type is then defined by a number of independent OSEK conforming schedulers. Each scheduler represents one processor. By adding and removing task schedulers, different single- and multicore ECUs may be simulated. By changing the required cycles for completion of a task, deployments to more and less sophisticated processor architectures are simulated. Using this approach, it is possible to simulate inter core interconnections like crossbars or busses by specialized simulation components. Simulation of ECU networks is possible as well by using already existing models, e.g. for CAN bus communication or wireless networks.

4 Contribution and Vision

As contribution of this work, we have shown the integration of AUTOSAR applications into our simulation framework. Existing solutions in this area are specialized to simulate the behavior of hardware in a very precise way. Although this is beneficial as a replacement or a support for HiL testing, it is not suited for validating applications in an open and integrated architecture. By including AUTOSAR application components as an executable artifact in FERAL, we created the possibility to perform an early virtual validation in a highly flexible environment. This supports the transformation of traditional hardware software systems to Cyber Physical Systems in the automotive domain.

In future it will be possible for suppliers to develop pure software systems, while the OEM performs an efficient distribution of the application components on a small set of ECUs. To give the suppliers a possibility to test their applications in the system context, the OEM can offer the access to a flexible simulation platform via a web service. This platform can then be used jointly by different companies under protection of their individual intellectual properties. Thereby the functional behavior and the performance of the software can be validated.

By supporting such a virtual validation process, suppliers are able to deliver products of high quality without developing own dedicated hardware. This way, the shift of paradigms in the development of automotive systems can be achieved.

5 Conclusion

The characteristics of CPS demand new validation techniques to support their development. One way to tackle this is the use of a virtual validation in system design stages to support the development of new system and software architecture concepts. In this paper, we have illustrated the extension of our FERAL framework to support AUTOSAR conforming software components, a precondition for evaluating future automotive integrated architectures. With help of the simulation, new information can be gathered that could not be retrieved with traditional testing techniques in early stages of development processes. This way, early design decisions can be substantiated by facts, reducing the probability of wrong design decisions, which saves great amounts of effort.
The work presented in this paper should be perceived as a first step to address the challenges of validating CPS. There is a tremendous potential for future work both on the technical and on the conceptual side. For example, to further enrich the functionality of FERAL and to include the collection of coverage metrics, the tool could be enhanced by a component to track the control flow within the components under test during execution. On the conceptual side, more focus could be set on the validation procedure. The combination of simulation models and dedicated test models for generation of representative test scenarios shall bring the benefit of enabling virtual validation with a higher degree of automation, specification coverage, and failure sensitivity.

References

[Bj11] Bjoerkbom, M. et al.: Wireless control system design and co-simulation. Control Engineering Practice, 19(9):1075 – 1086, 2011. Special Section: DCDS'09 The 2nd IFAC Workshop on Dependable Control of Discrete Systems.

[Bl11] Blockwitz, T. et al.: The Functional Mockup Interface for Tool independent Exchange of Simulation Models. In Proceedings of the 8th International Modelica Conference, 2011.

[Di13] Dietrich, M. et al.: Die virtuelle ECU fuer AUTOSAR. In Automobil Elektronik. 02/2013.

[ISO11] International Organization for Standardization: ISO 26262: Road vehicles – Functional safety. 2011.

[Ku13] Kuhn, T. et al.: FERAL - Framework for simulator coupling on requirements and architecture level. In Formal Methods and Models for Codesign (MEMOCODE), 2013 Eleventh IEEE/ACM International Conference on, pages 11–22, Oct 2013.

[MDF13] Martinus, M.; Deicke, M.; Folie, M.: Virtueller Fahrversuch – Hardwareunabhngige Integration von Seriensoftware. In ATZelektronik. 05/2013.

[Ro13] Rolfsmeier, A.: Virtuelle Absicherung von Fahrerassistenzfunktionen. In HANSERautomotive. 07-08/2013.

[RSB11] Roth, C.; Sander, O.; Becker, J.: Flexible and Efficient Co-simulation of Networked Embedded Devices. In Proceedings of the 24th Symposium on Integrated Circuits and Systems Design, SBCCI '11, pages 61–66, New York, NY, USA, 2011. ACM.

[WB13] Wang, B.; Baras, J.S.: HybridSim: A Modeling and Co-simulation Toolchain for Cyber-physical Systems. In Proceedings of the 2013 IEEE/ACM 17th International Symposium on Distributed Simulation and Real Time Applications, DSRT 13, pages 33–40, Washington, DC, USA, 2013. IEEE Computer Society.